CETONI Elements

Manual

ORIGINAL MANUAL – FEBRUARY 2022

CETONI GmbH
Wiesenring 6
07554 Korbussen
Germany

T +49 (0) 36602 338-0
F +49 (0) 36602 338-11
E info@cetoni.de

www.cetoni.de

Software License

The software and the documents provided are protected by copyright. Installation declares your acceptance of the conditions of the licensing contract.

License Agreement

CETONI GmbH grants the purchaser the non-exclusive and non-transferable license right to use the software on a standalone computer or a networked computer system (LAN). It is expressly forbidden to copy or duplicate the software in part or in whole in any way. The same is applicable for mixing and combining the software with other software. The purchaser may create one back-up copy of the software for archival purposes. CETONI GmbH reserves the right to alter, develop and improve the software or develop new software to replace it. cetoni is not obliged to inform the purchaser about any alterations, new developments or improvements to the software or make these available to him. No legally binding assurance of certain features is given. CETONI GmbH is not liable for damages unless the damages are as a result of wilful intent or gross negligence on the part of CETONI GmbH or its employees. Any liability for indirect, collateral, or consequential damages is excluded.

1 Summaries and directories

1.1 Table of contents

1.2 Change history

REVISION	CHANGE
20110722	Creation of software manual QmixElements
20111215	Documentation neMESYS-, QmixV- and Qmix Q- Plugin added Scripting system documented
20120312	Documentation of new core script functions added (Show Message und Interrupt Script)
20120319	Documentation of new core script functions added(Parallel Sequence)
20120522	Documentation of tubing pump plugin, image analysis plugin and video builder plugin added
20120523	Variable system documented device script functions documented
20120724	Improvements and changes of variable system documented Updated core script functions documentation Documentation of I/O channel scaling and configuration added Added documentation for Conditional Loop and Conditional Sequence
20120816	Added chapter about creating device configurations Updated rotAXYS Documentation (new features *Add Move XY Script* and *Add Move Z Script* and Space Navigator usage documented)
20120927	Documentation of rotAXYS / neMAXYS plugin updated (documented manually setting of speeds and documented script based speed setting)
20121012	Added documentation of *User Input* function, variable output via *Show Message* function documented properly
20121114	Added documentation for Data Logger and Camera plug-ins
20121220	Documented how to change function captions
20130118	Added process data graph documentation
20130206	Add documentation for *Create Property Variable* function Add documentation for single axis control functionality (manual and script functions) Updated neMESYS continuous flow documentation with new features Added documentation for PID Control function

20130311	Added documentation of Startup Screen Added documentation for process data graph curve color selection
20130402	Added documentation for Device Configurator functionalities Add and Exchange devices
20130508	Added documentation for neMESYS continuous flow cross flow feature
20130523	Updated Valve-Plugin documentation Updated Controller-Plugin documentation, added documentation for controller channel scaling Added camera video recording feature documentation Documented Well Plate Configuration dialogue
20130715	Updated Tubingpump-Plugin Added documentation for tubingpump script functions
20130718	Added documentation for synchronous neMESYS pump start / stop
20130821	Added documentation for LED array plugin
20130828	Added documentation for neMESYS continuous overlap flow
20131014	Added documentation for neMESYS change flow rate functionality
20131119	Added documentation for neMESYS I/O channels
20140127	Updated documentation of Qmix Controller Plugin Added documentation for user-defined controller channels
20140214	Added documentation for device process data identifiers and direct device property access to section Script System Updated documentation of PID controller script function
20140909	Inserted documentation of the analog channel monitoring functionality of the neMESYS plugin and a section about predefined analog channel confiugurations in the qmixio plugin chapter.
20141104	Documented functionality linking neMESYS pumps and scripting system. Added section about creating device configurations with simulated devices.
20150105	Updated LED array documentation for new LED array V2 devices
20150629	Added documentation for spectroscopy plugin
20150723	Updated rotAXYS plugin documentation
20151021	Updated documentation for new syringe configuration

	Updated a number of neMESYS screenshots Added documentation for virtual I/O channels (Virtual Channels) New script function *Write Device Property* documented
20160218	Changed format to new corporate design Updated Qmix I/O Plugin documentation with to document latest software changes
20160403	Added chapter “Working with projects” for new project feature Updated documentation for new device configurations functionality (import / export)
20160414	Added documentation for valve automatically Updated script function icons
20160516	Updated spectroscopy plugin documentation
20160822	Updated Continuous Flow documentation Added documentation for continuous flow valve configuration Added section about pump valve assignment Section added about configuration of external devices connected to neMESYS I/O interface
20161121	neMESYS Plugin: Added documentation for pressure controlled continuous flow
20161128	Spectroscopy Plugin: Added documentation of online analysis channels CSV Data Logger: Documented script functions Start/Stop Logging Process Data Graph: Documented script functions Start/Stop Logging
20161216	CANopen Tools Plugin documentation added
20170105	Spectroscopy Plugin: Added documentation of Analyze IQ add-on
20170217	Added documentation of project import / export functionality Updated images in chapter “Creating Device Configurations”
20170502	rotAXYS / neMAXYS plugin: Added documentation for loading work table configuration files for rotAXYS360
20170614	neMESYY plugin: Fixed typo in continuous flow section
20170823	neMESYS Plugin: Added documentation for valve assignment via drag & drop, added documentation for pressure sensor assignment via drag & drop, updated pressure monitoring documentation, updated dokumentation for I/O scaling and assignment of external valves, added documentation for Reference Move script function Qmix I/O Plugin: added documentation of I/O channel configuration, updated documentation of I/O channel scaling and added documentation for new unit system

	rotAXYS Plugin: documentation added for Find Home script function
20171019	Updated event log documentation
20171025	Updated installation instructions
20170122	neMESYS Plugin: Documentation for syringe search added, pictures updated Datalogger Plugin: Documentation added for switching the scaling of the X-axis of the graphical logger, pictures updated CANopen Tools Plugin: Documentation of the new functions: DCF Export / Import, CAN Bus Trace
20180205	Qmix I/O Plugin: Added documentation for I/O channel calibration
20180306	Device Configurator Plugin: Updated image material Script System Plugin: Updated image material, added documentation for new item scaling functionality, added documentation for new Read Timer function, added documentation for improved function selection and for function grouping Controller Plugin: Pictures updated, documentation for setting controller parameters updated
20180405	rotAXYS Plugin: Updated pictures, added documentation for new Z-Axis slider
20180621	Introduction to QmixElements: Added UI Access Manager documentation
20180821	Spectrocopy Plugin: • Added documentation for inserting metadata when saving spectra manually • Added documentation for Analyze IQ Dataset Preparation Dialog • Documentation for Realtime Baseline Correction added • Pictures updated • Conversion of AIQ to CSV files documented
20181105	neMESYY Plugin: • Variable support for Continuous Flow Script function documented • New script function Change Continuous Flow documented Script system: • New variable declaration sequence documented • Loop termination for counting loop documented Peristaltic pumps Plugin • Pictures updated • Documentation for creation of user-specific hoses Device Configurator: • Adding devices by double click documented

20181116	neMESYS Plugin: • Documented new continuous flow configuration
20190102	Introduction to QmixElements Software • added documentation for new advanced docking system with Views and Perspectives neMESYS Plugin: • added documentation for pressure monitoring shield icon state
20190304	Introduction to QmixElements Software • Documentation for new Settings dialog for configuration of global settings Scripting System • Changes to Show Message function documented • Documentation for new Stop Request Button and $StopRequested script variable • Script Auto-Start functionality and configuration documented
20190320	Spectroscopy Plugin: • Updated images • Value display in spectrum plot added • Documentation updated for Analyze IQ Model creation • new Spectra Viewer documented CSV data logger • Pictures updated Process data diagram: • Image material updated • Documentation for value display in diagram using mouse over
20190607	Qmix V Plugin: • added section Creating Valves • added Section Deleting Valves • updated images
20190726	DAQ Plugin: • added DAQ documentation
20190919	neMESYS Plugin: • updated documentation for syringe selection dialog
20191016	Introduction to QmixElements Software • added documentation for Event Log export functionality Qmix I/O Plugin • documented new grouped I/O channel view • updated pictures • added search functionality documentation

20200624	Scripting System • Added documentation for script error handling configuration • documented new Break script function
20200819	Creating Device Configurations: • Added documentation for optional Add-ons
20210105	Nemesys Continuous Flow • updated documentation for presure controlled flow configuration • removed documentation for overlap flow
20210318	Balances Plugin: • Added documentation for balances plugin
20210325	Nemesys Plugin: • Added documentation for Nemesys S and Nemesys M force monitoring Scripting system: • Added documentation for creating own script functions • Documentation for Script Functions Calls added
20210406	CANopen Tools Plugin • Added documentation for CANopen script functions
20210603	Renaming of QmixElements to CETONI Elements
20210707	Updated minimum system requirements (switch to 64-bit version)
20210830	Balances Plugin: • Updated Sartorius Balances configuration settings documentation
20220116	Introduction CETONI Elements: • New license system and license key documented • Documentation for First Start improved • Image material updated • Documentation of new personalization settings (color and style of the interface) Nemesys Plugin: • Show and hide Nemesys force sensor documented • Added documentation about using device references in script functions • Image material updated Scripting system: • Documentation on device references added • Documentation for new variable functions (lists, arrays, device references) added • Image material updated Graphlogger

	• Documentation script functions updated - new parameters
20220203	Python Add-on • Added Pyhon Add-on documentation SiLA Add-on • Added SiLA Add-on documentation

2 Introduction

2.1 Foreword

Thank you for deciding to purchase a CETONI product. We would like to support you with this handbook insofar as possible in your interaction with the software. We are directly available for any questions or suggestions that you may have.

2.2 Symbols and Signal Words Used

The following symbols are used in this manual and are designed to aid your navigation through this document:

HINT. Describes practical tips and useful information to facilitate the handling of the software.

IMPORTANT. Describes important information and other especially useful notes, in which no dangerous or damaging situations can arise.

ATTENTION. Indicates a potentially damaging situation. Failure to avoid this situation may result in damage to the product or anything nearby.

CAUTION. Describes a situation that may be dangerous. If this aspect is not avoided, light or minor injuries as well as damage to property could result.

3 Installation

3.1 Installing the Software

IMPORTANT. In order not to impair time-critical control processes of the CETONI Elements software, no other computation-intensive applications should be executed on the controller PC.

IMPORTANT. Install the CETONI Elements software + device drivers before connecting your device via USB to the PC.

ATTENTION. Danger of malfunction / data loss due to standby / sleep mode. Switch off the activation of the standby / sleep mode on your PC or notebook to avoid a malfunction of the hardware driver.

ATTENTION. Risk of malfunction / data loss due to power saving option. Always operate your PC or notebook in mains operation and deactivate selective USB power saving to avoid malfunctioning of the hardware driver.

To install the software, insert the CETONI Elements CD ROM into your CD/DVD drive, or plug the CETONI Elements USB stick into a free USB port. Then start the file *CETONI_Elements_Setup.exe* from the CD or USB stick. The Installation Wizard then leads you through the installation of the CETONI Elements software and the hardware drivers.

IMPORTANT. Under Windows, you need to be logged in with administrator rights in order to be able to install the hardware drivers.

The Installation Wizard now leads you through the installation of the software.

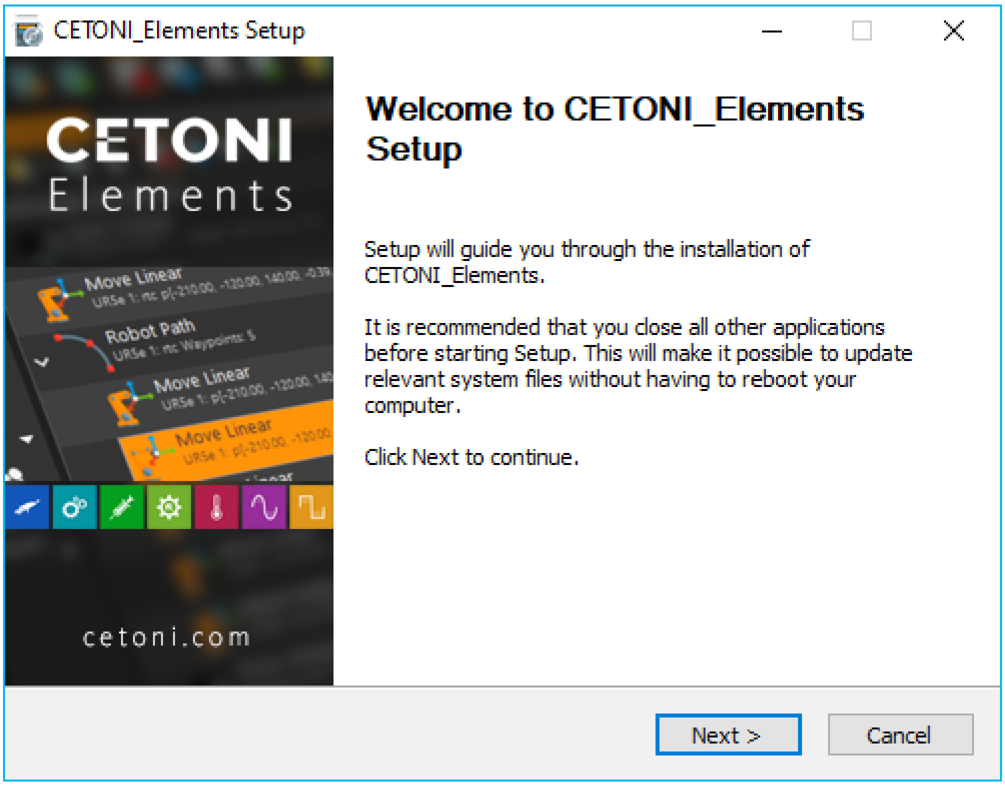


Figure 1: Setup Wizard Welcome Page

The hardware device drivers are installed during the installation. This step is only required if the drivers are not already installed on your computer. If the hardware drivers are already installed, please deactivate the driver components (Figure below).

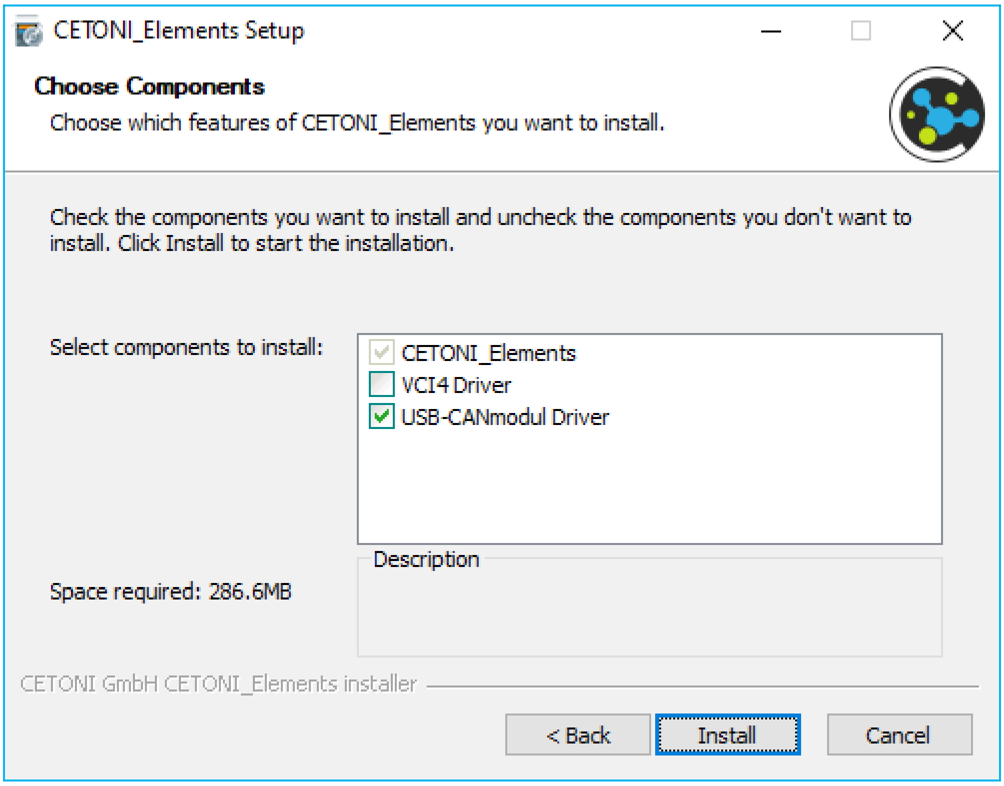


Figure 2: Selection of installation components

3.2 System Requirements

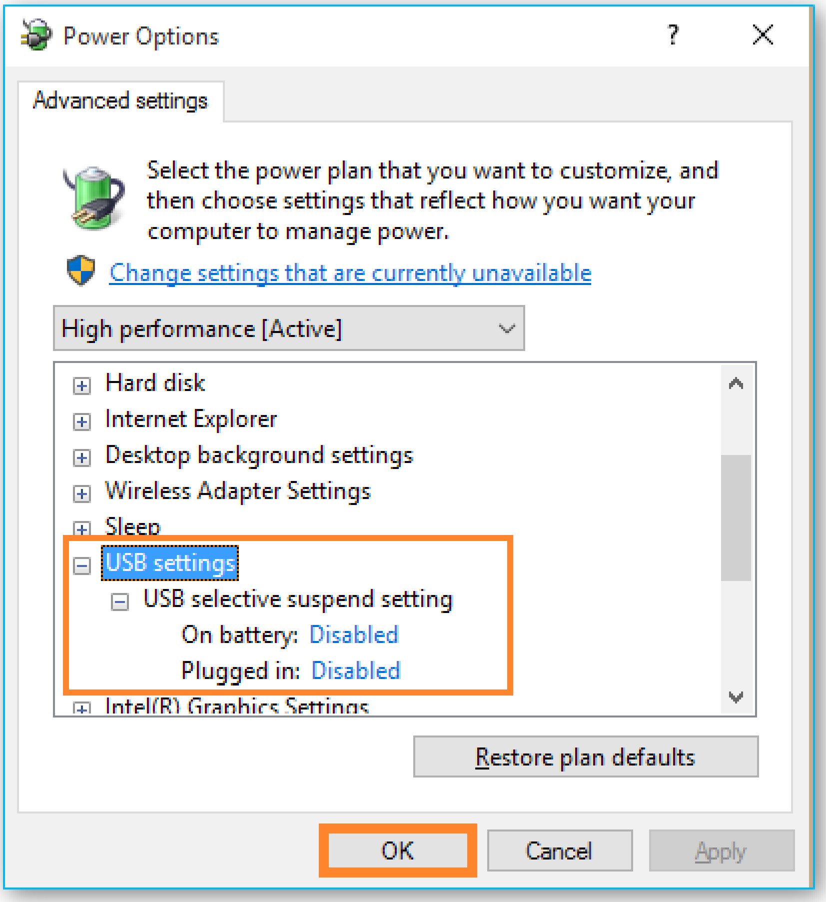
Your computer should meet the following system requirements in order to be able to use the software:

- PC with an Intel Core i3 processor (or higher) - minimum 1.3 GHz (Most processors produced in 2015 and later should work well with this software)
- at least 8 GB RAM (recommended 16 GByte).
- free hard disk space of approx. 400 MByte
- at least 2 free USB (2.0 or 3.0) interfaces
- Operating system: Windows 10 64-bit (Windows 7 and Windows 8 may work, but are not tested)
- Monitor resolution: 1920 X 1080 (HD) or higher
- wheel mouse

Standby or sleep mode must be deactivated under Windows when the software is used, because activation of the standby or sleep mode can lead to malfunctions of the hardware device driver.

ATTENTION. Danger of malfunction / data loss due to standby / sleep mode. Switch off the activation of the standby / sleep mode on your PC or notebook to avoid a malfunction of the hardware driver.

To avoid disconnections from the device, please disable selective power saving for USB ports in the power options:



3.3 Installing USB Device Drivers

The USB device drivers are required for the USB connection to your device. When you connect your device to your PC via USB for the first time or connect the device some time later to another USB slot, Windows starts the Hardware Wizard automatically, this detects a new USB device and installs the requisite drivers if you have previously installed the CETONI Elements software from CD or USB stick.

IMPORTANT. The installation procedure for the device drivers can vary slightly depending on the Windows version used.

ATTENTION. Danger of data loss as a result of uncontrolled switch off procedures! Always close the software before you switch off your device! Only then are all settings saved correctly, and the configuration data will not be lost.

4 Introduction to CETONI Elements Software

4.1 Initial Operation

The first time the software is started, a default configuration is loaded: The software loads only its core without any device-specific plugins. The default configuration is active, when you see the following startup screen (figure below).

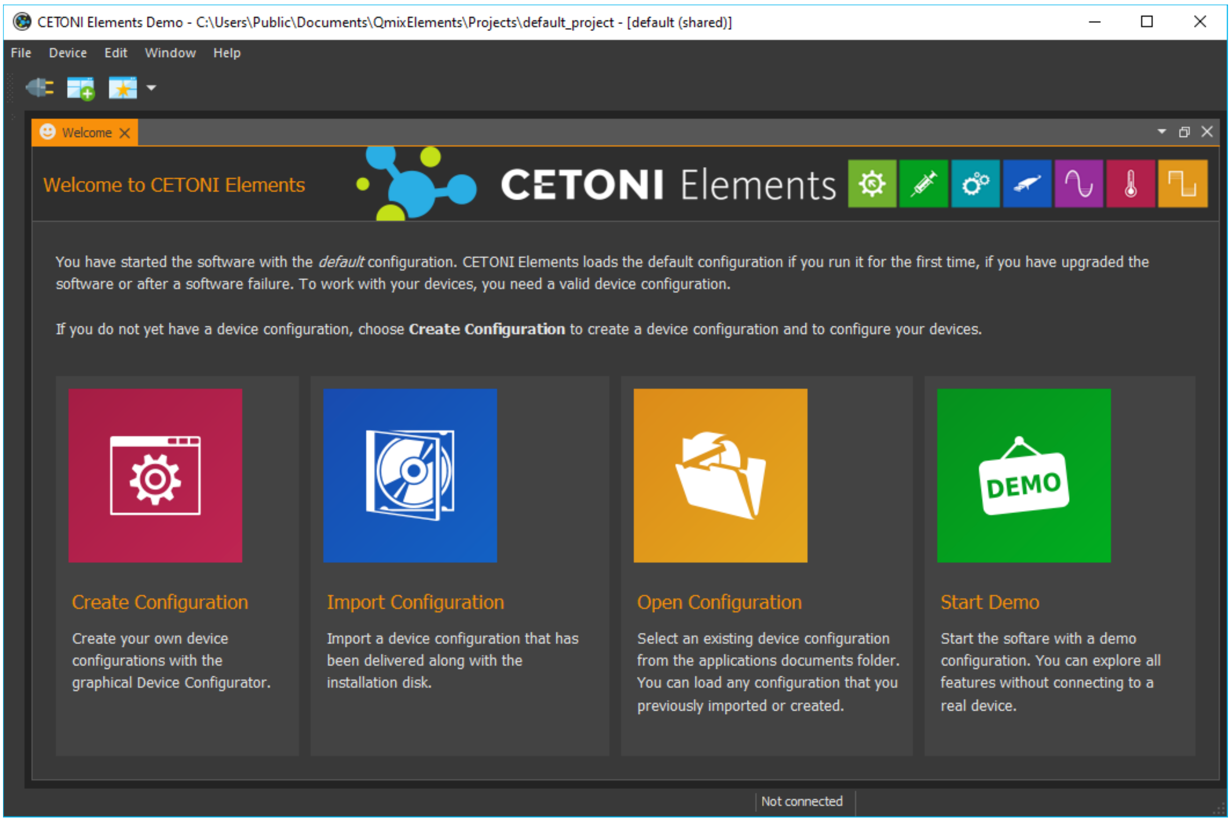


Figure 3: First start of CETONI Elements – Startup Screen

You have to install and load a device configuration, in order to be able to control your devices with the software. A device configuration is a collection of configuration files, which describe the configuration of all devices and modules, which together form a complex system or device. In the startup screen you can see all ways to load a valid device configuration:

- *Create Configuration – This button starts the Qmix Device Configurator that lets you create arbitrary device configurations yourself. More information on creating device configurations can be found in chapter 5 Creating Device Configurations.*

- *Open Configuration* – This allows you to load an existing device configuration that was previously installed on your system.

- *Import Configuration* – Select this menu item to install a device configuration from a disk or USB-stick (eg from CETONI Elements installation CD) – see chapter 4.4 Import Device Configuration.

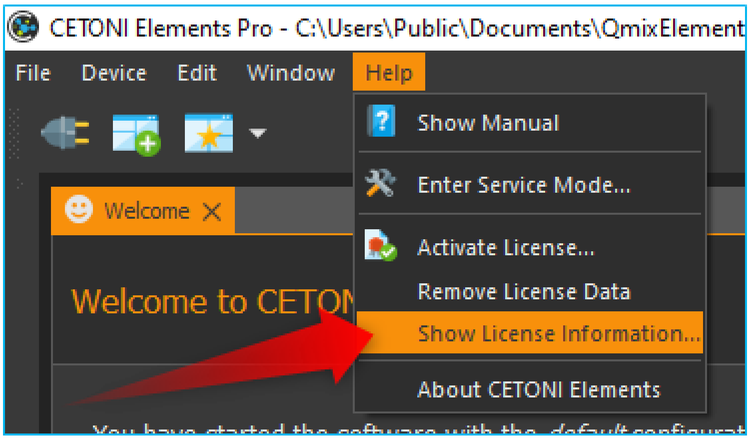
- *Start Demo* – If you want to start the application in demo mode, eg for presentations and tests or if you currently have no real equipment, then click this button.

IMPORTANT. Also after a system crash or after an update of the software, the default configuration is loaded, and the start-up screen appears.

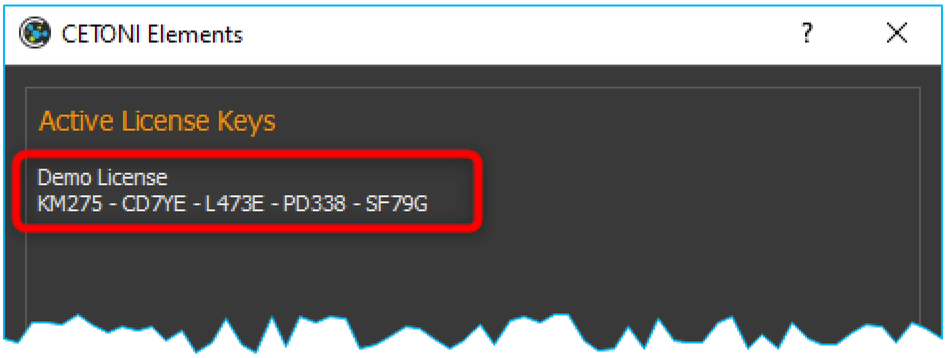
4.2 License Keys

4.2.1 Activate License

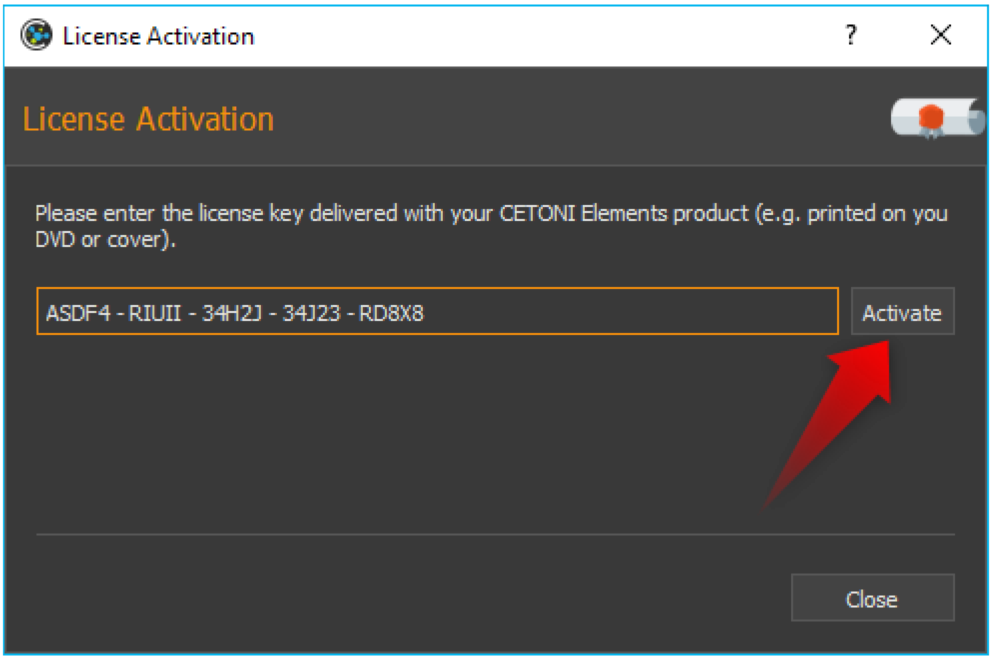
Right after the initial installation of the software, you have an active demo license. To view your installed licenses, select *Help → Show License Information* from the main menu.



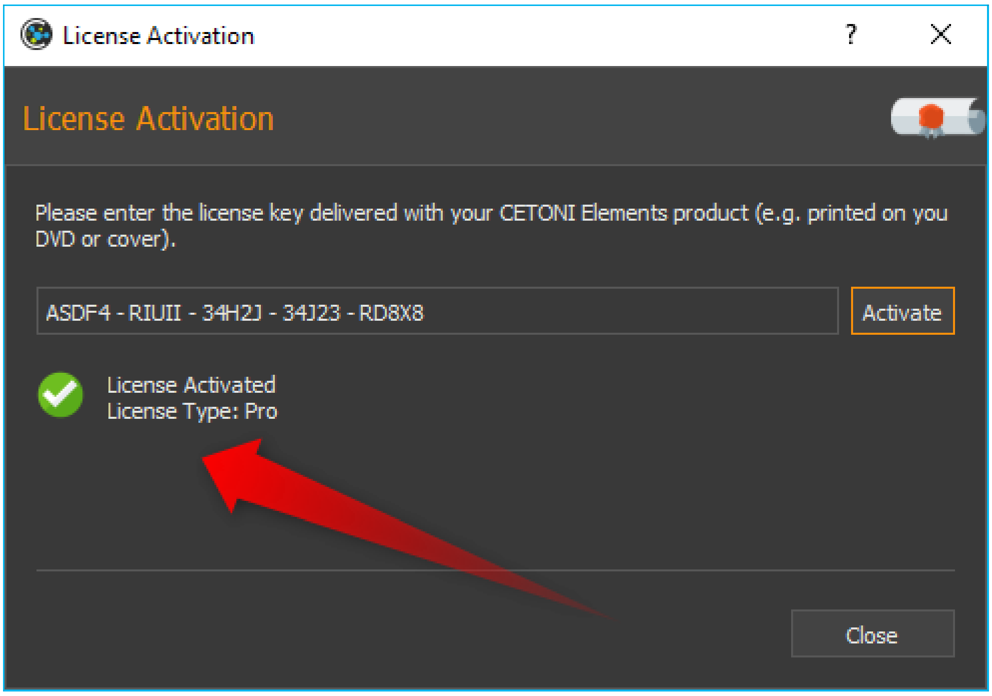
The demo license allows you to create device configurations with simulated devices and to test all functions of the software.



In order to use the software to control your devices, you must register the license code you have received in the software. To do this, select *Help → Activate License* in the main menu.



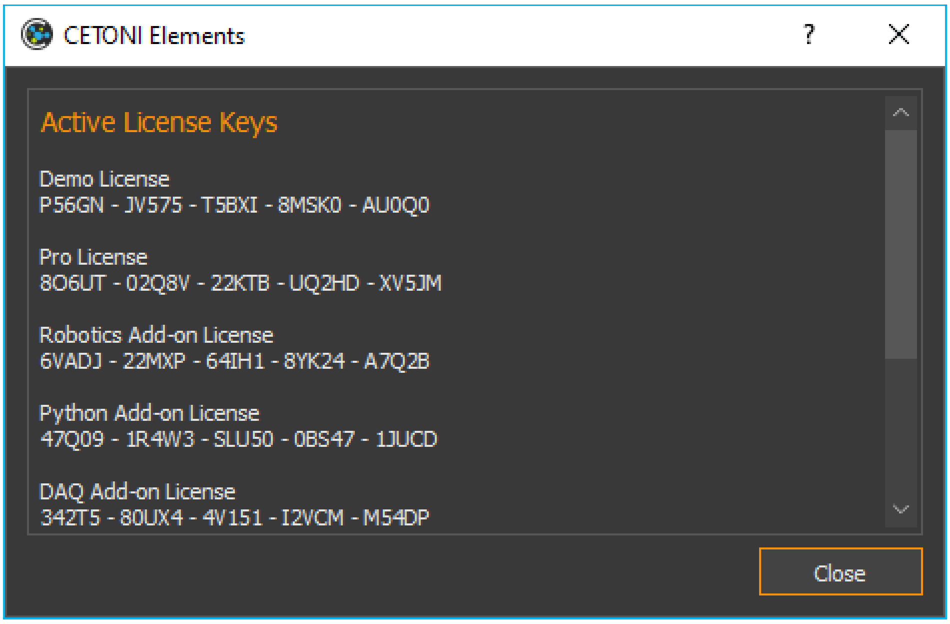
In the license activation dialog, enter your license code and then click the *Activate* button. If you have entered the license key correctly, the activation of the license is confirmed and after restarting the software you can work with the activated license:



If you want to upgrade your license or activate additional licenses for further add-ons, you can proceed exactly as described above.

4.2.2 Overview of Activated Licenses

Using the menu item *Help* → *Show License Information* in the main menu of the application, you can display an overview of the current licenses:



4.2.3 Remove Licenses

If you want to remove all licenses, select *Help* → *Remove License Data* from the main menu. It is not possible to remove individual licenses, you can only remove all licenses together. After removing the licenses, you will work with a demo license again.

4.3 Create Device Configuration

The *Create Configuration* button allows you to create new device configurations. To use individual devices or complete systems consisting of several individual devices in CETONI Elements, you must create a device configuration. For this purpose, use the Device Configurator, which is integrated in CETONI Elements. For detailed instructions on how to create device configurations, see chapter 5 Creating Device Configurations.

4.4 Import Device Configuration

For most devices and modules you can create your own device configurations with the CETONI Elements Device Configurator. If you have devices that are not supported by the Device Configurator or if you use devices that need to be preconfigured by CETONI, you will find the device configuration on the installation media (CD or USB stick) that you received with your device.

Perform the following steps to import an existing device configuration:

HINT. If you create a configuration with the Device Configurator, the configuration will be installed automatically and you don't need to perform the following steps.

(1) Select the *Device → Import Configuration* menu item from the main menu.

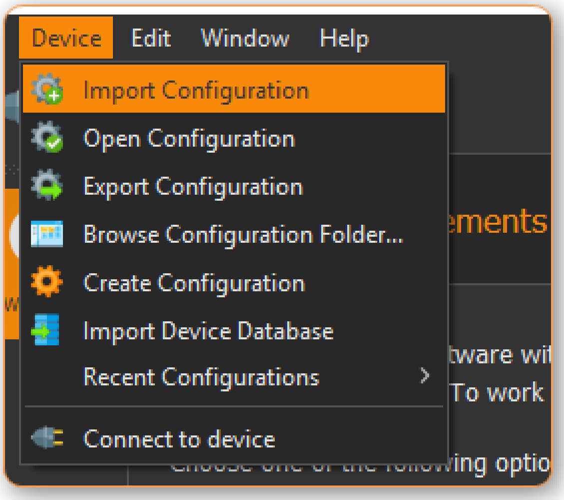
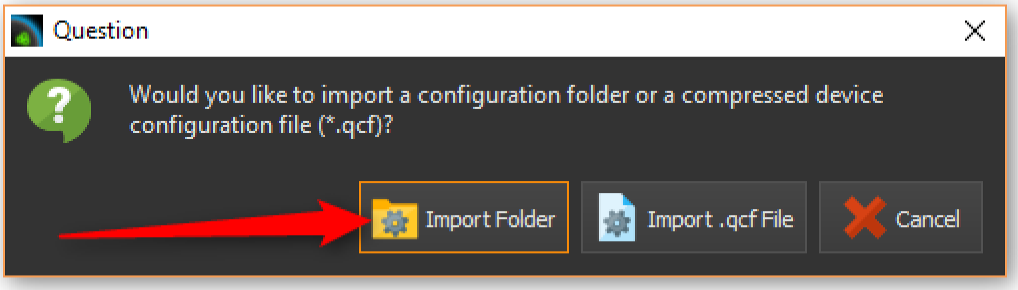


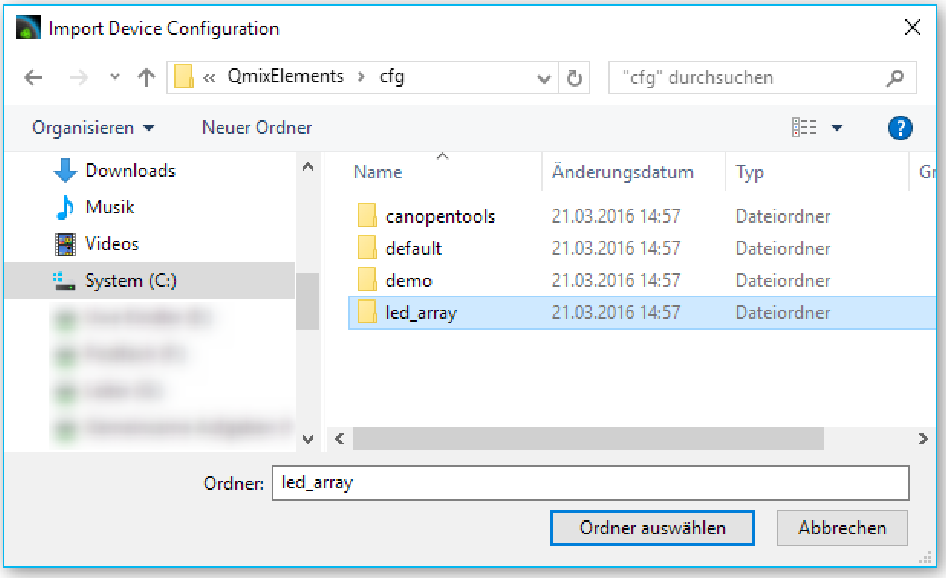
Figure 4: Import device configuration

(2) Now select whether you want to import a folder with a device configuration (*Import Folder*) or a compressed device configuration file (*Import .qcf File*). On the installation media that came with your device, the device configurations are stored in folders. Therefore, select the item *Import Folder*.



(3) In the dialog that now opens, select the *DeviceConfiguration* directory on the installation CD or browse to another directory containing a valid device configuration.

(4) Now enter a unique name for the configuration under which it will be stored on your computer.



(5) Acknowledge the query to restart the application.

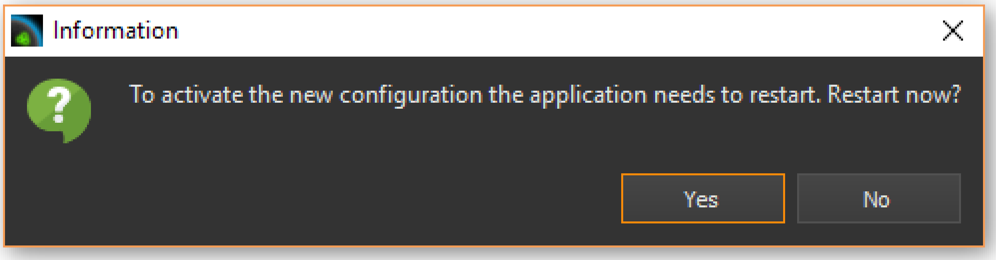


Figure 6: Restart application

After completing these steps, the software with the device-specific plugins will start to load.

IMPORTANT. When importing, device configurations, these are always imported into the current project.

4.5 Export Device Configuration

You can also export device configurations. If you would like to give a device configuration to employees or if you need the send a device configuration to the CETONI support staff, then you should export a device configuration as compressed **.qcf* file. To export a device configuration, select the main menu item *Device* → *Export Configuration*.

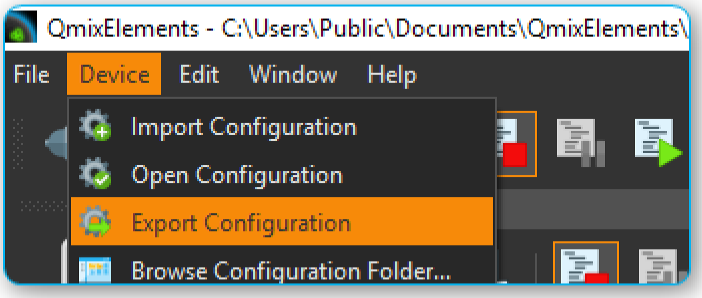


Figure 7: Export device configuration

Device configurations are exported as compressed *.qcf files.

4.6 CETONI Elements – The Basics

4.6.1 Introduction

The CETONI Elements application is a plugin-based, modular software solution for controlling a wide range of laboratory automation devices, in particular CETONI's Qmix modules. The software consists of a core, which provides the basic functions and services. This includes the application window, the event log, and the toolbar.

This core system is then expanded by device- and application-specific software modules, referred to as plugins. They provide the user with specific software solutions appropriate for virtually any device configuration within the standardized user interface.

4.6.2 Show User Manual

The software manual can be accessed via the application's help menu (*Help → Show Manual*).

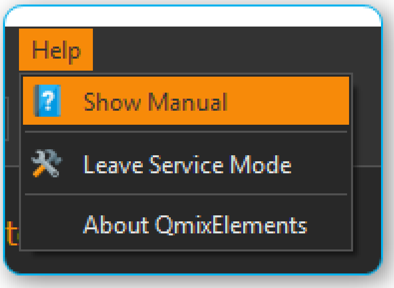


Figure 8: Access software manual

IMPORTANT. The software manual is provided in the pdf file format. A pdf-viewer must be installed on your system to be able to view this document.

4.6.3 Show Software Version

If you need to know the version of the installed CETONI Elements, e.g., for service purposes, select *Help → About CETONI Elements* from the main menu (see figure below).

Figure 9: Displaying the installed software version

4.7 Overview of the Main Application Window

The application window of the software consists of the following central components:

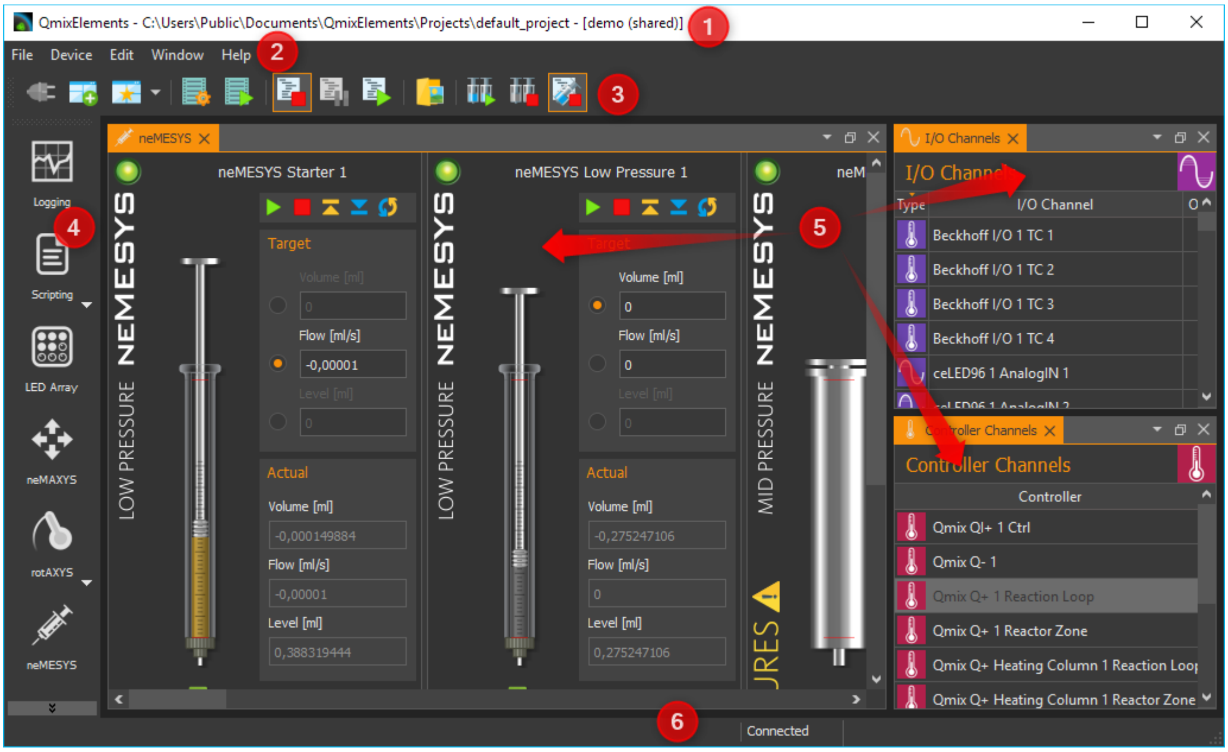


Figure 10: Overview of application Main window

1 Title bar

2 Main menu

3 Toolbars (can be freely positioned and can be hidden)

4 Sidebar for showing *Views* (can be freely positioned and can be hidden)

5 The individual *Views*, which can be moved freely in the user interface

6 Status bar

4.8 Title bar

The title bar displays the application name, the current project path and the name of the currently loaded device configuration. The following example illustrates this more clearly. If the title bar contains the following text:

CETONI Elements – *C:\Users\Public\Documents\CETONI Elements\Projects\default_project* *[demo (shared)]*

then you will get the following information:

- **application name**: *CETONI Elements*
- **current project path:** *C:\Users\Public\Documents\CETONI Elements\Projects\default_project*
- **device configuration:** *demo (shared)*

4.9 Views

4.9.1 Introduction

The *Views* are windows that contain the display and control elements for a specific task or device. For example, there are *Views* for the graphical logger, the display of I/O channels or for the script editor.

All views can be freely moved within the main window via drag & drop. All views can be shown and hidden and the views can also be removed from the main window via drag & drop to move them to another screen. This allows you to adapt the user interface of the CETONI Elements software optimally to your requirements and your application and thus configure an individual graphical user interface.

4.9.2 Show Views

When you start the software, not all views may be displayed. To display a view, select *Window → Show View* from the main menu. You will then see a menu with all views that are available in the application. Simply click on the corresponding menu item to display the view (see figure below).

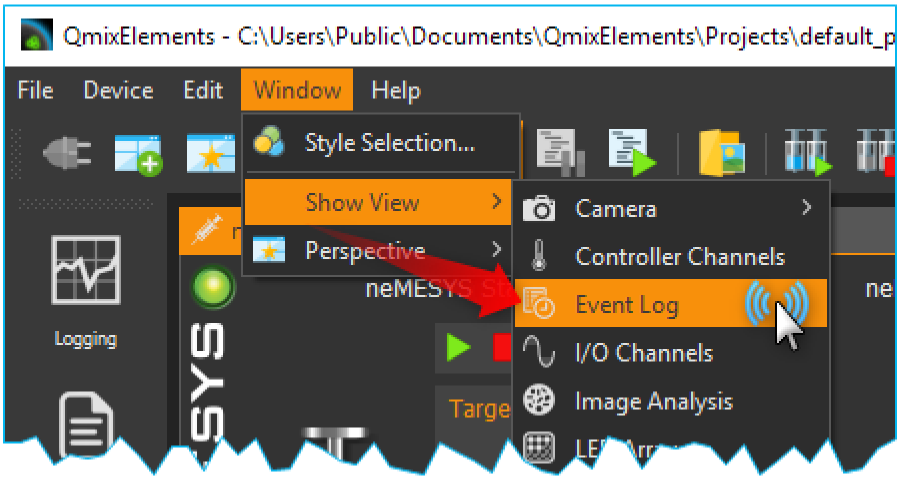


Figure 11: Displaying Views via the Menu

For certain devices, there may be a separate view for each device (e.g. for cameras). In this case, the views in the View menu are grouped together and can be displayed individually.

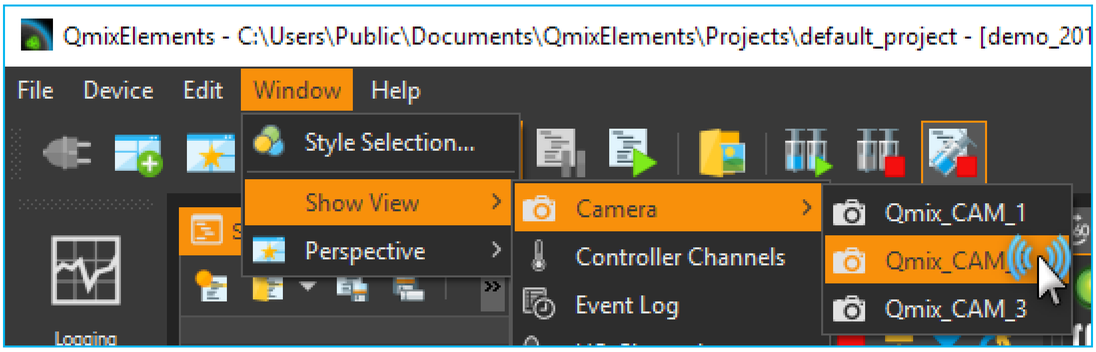


Figure 12: Displaying Views for Devices

Alternatively, you can show some views via the *Sidebar* without having to open the main menu.

4.9.3 Moving Views

To move a view, you must first detach it from its current position. There are several ways to do this. The first option is Drag & Drop. To do this, click with the left mouse button on the tab of the view ❶ and simply drag the view from its position ❷ with the mouse button held down. As long as you do not release the mouse button, you can move the view freely (see figure below).

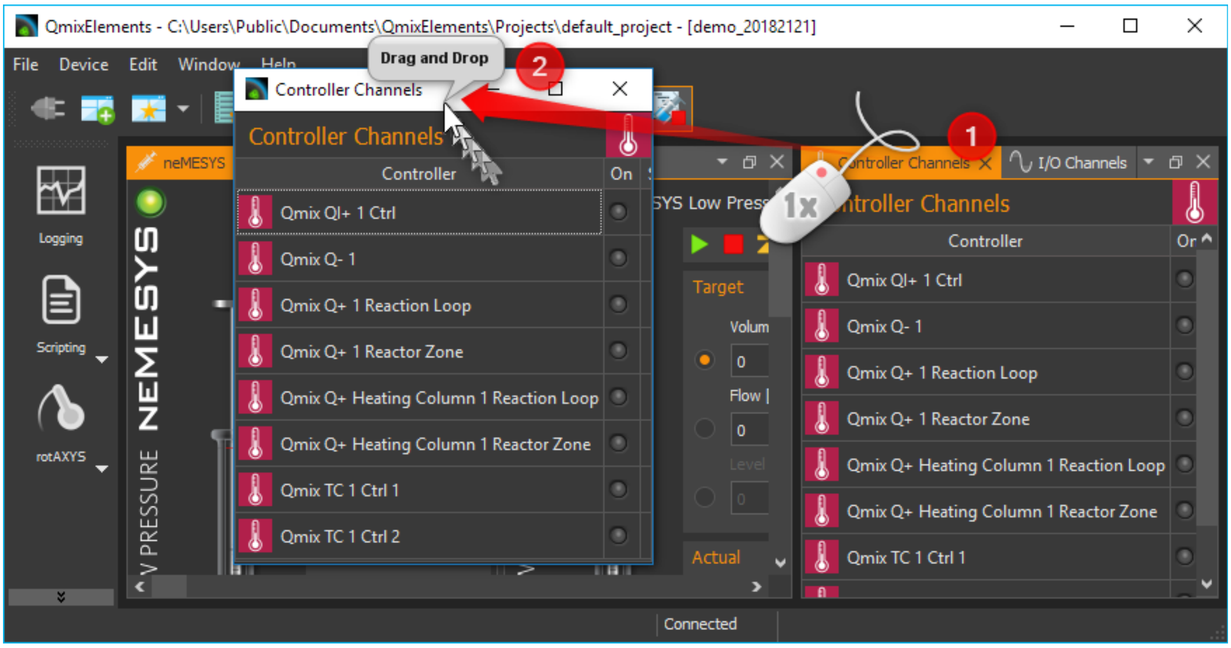


Figure 13: Move View via Drag & Drop

Alternatively, you can detach the view from its position by double-clicking on its tab. The third option is the context menu of the view. To do this, right-click the tab and choose *Detach* from the context menu.

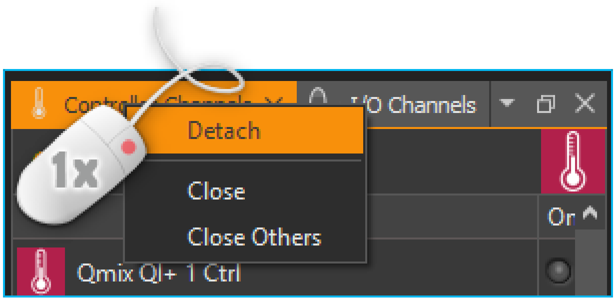


Figure 14: Detach view from its position via context menu

If you have detached the view from its position, the view is displayed in a separate window. Like any other operating system window, you can now enlarge, reduce, maximize, or move this window to another screen.

4.9.4 Docking Views

You can dock views, or groups of views which have their own window, into the main window or any other view window via drag & drop. To do this, simply drag the window of the view over the target window into which the view/view group should be docked. The destination window will then display icons - the *drop targets*. These target markers indicate the areas in which the window can be inserted. If your target window contains only one view, you will see the following 5 drop targets.

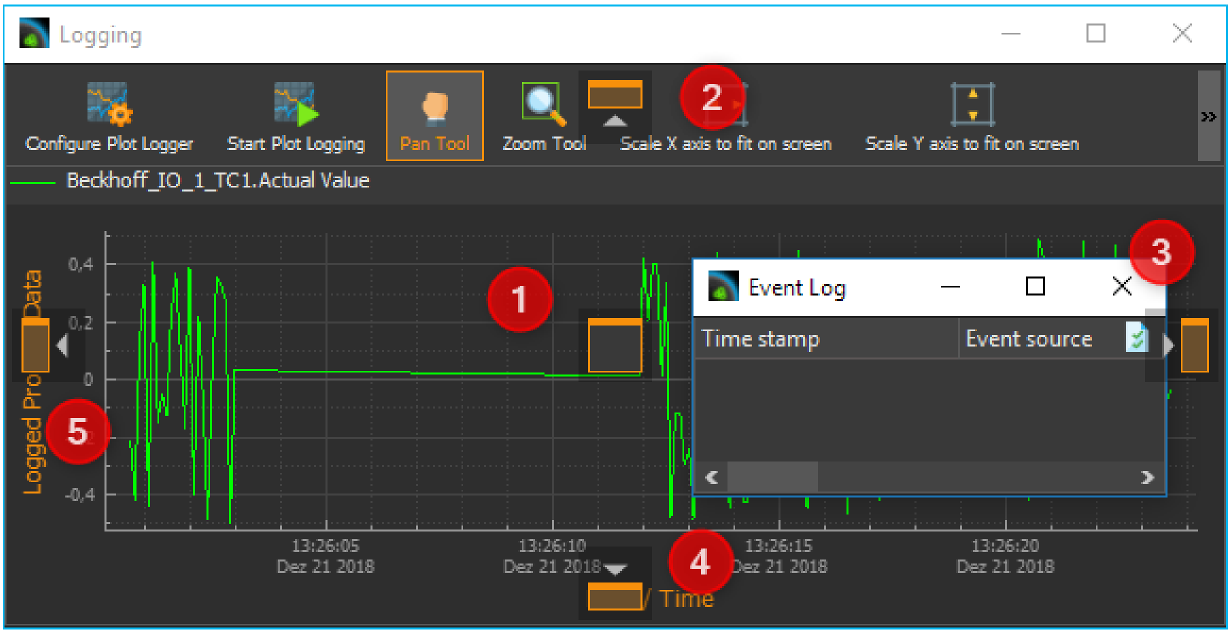


Figure 15: Drop target markers

If you move the mouse cursor over a *drop target* while moving a window, a colored frame (*drop preview*) indicates the position (*dock area*) where the view would be inserted.

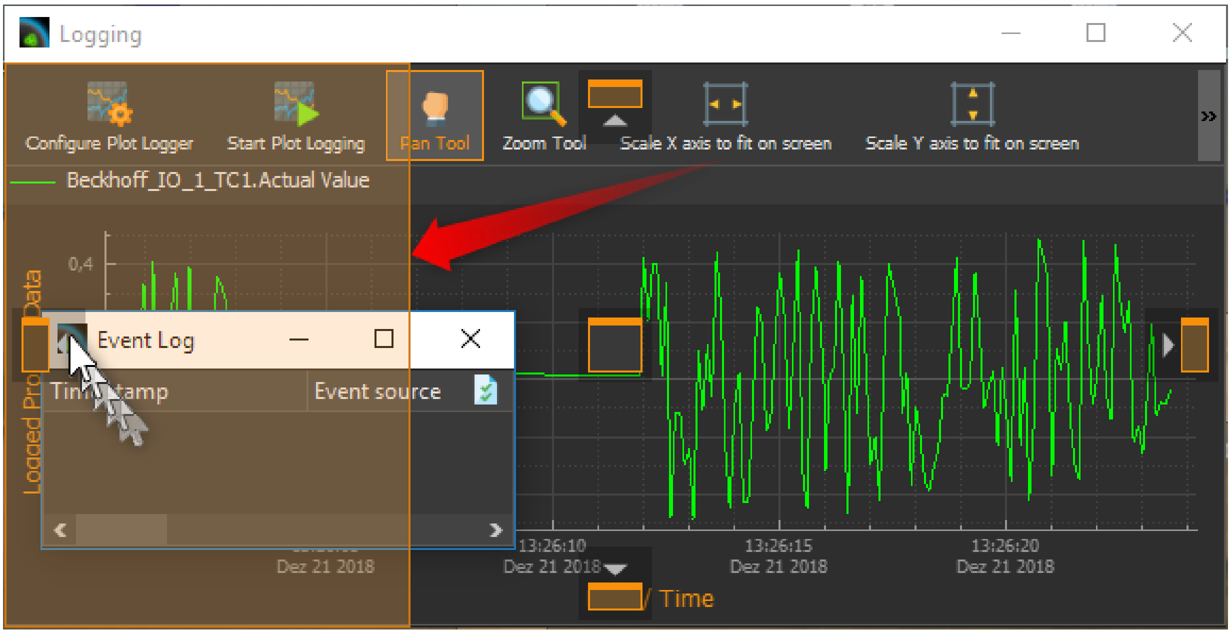


Figure 16: Drop Preview Display

If you now release the mouse button over the target marker, the window is inserted at the position indicated by the drop preview.

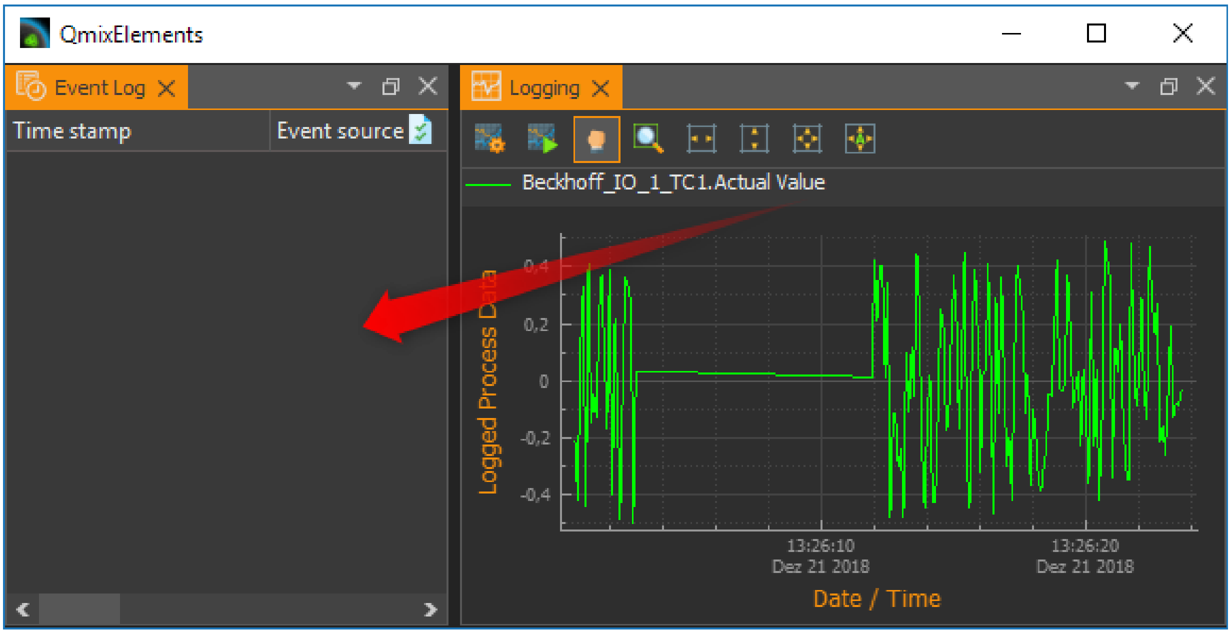


Figure 17: View inserted on the left side

If you select the symbol in the middle of the window as the drop target, the new view is inserted as an additional tab on the position of the existing view.

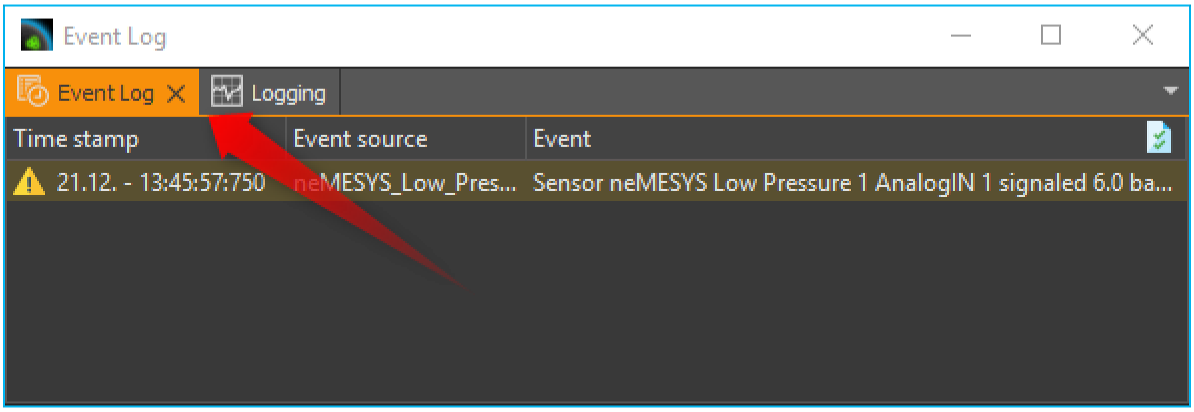


Figure 18: View Group

If you drag your view over a window that already contains several views, additional drop-target markers are displayed. This means that you then have additional options for docking the view window. You can see this in the figure below. There are the outer drop targets ❶, which you already know from the previous section. In addition, there are five other drop targets that are displayed in the form of a cross ❷.

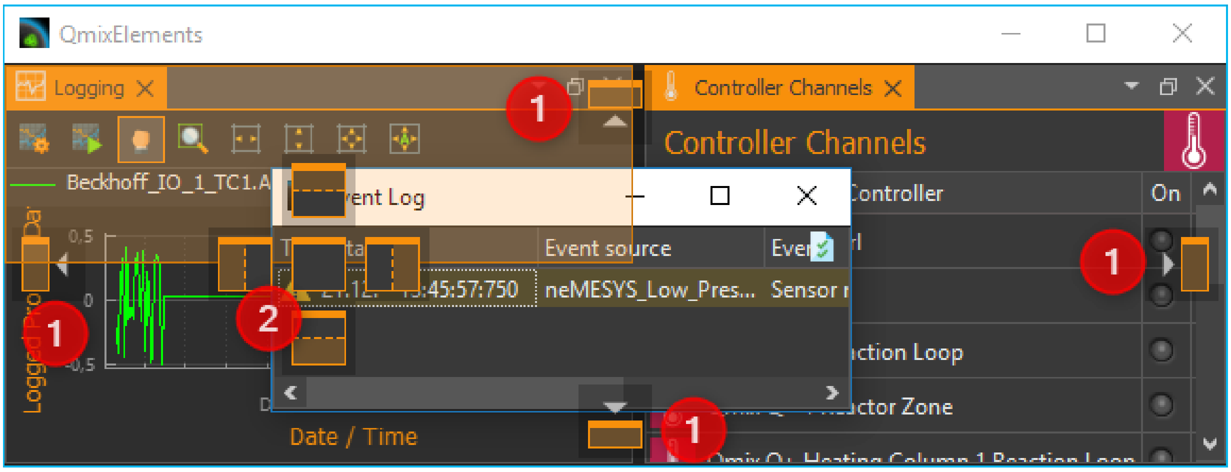


Figure 19: Drop target markers

Move the window with the mouse button pressed over the individual markers to see the drop preview and to get a feel for the possibilities for docking a window.

If you want to dock a view as an additional tab, you can move the mouse cursor either over the middle drop-target icon 1 (see figure below) or over the title bar of an already docked view 2.

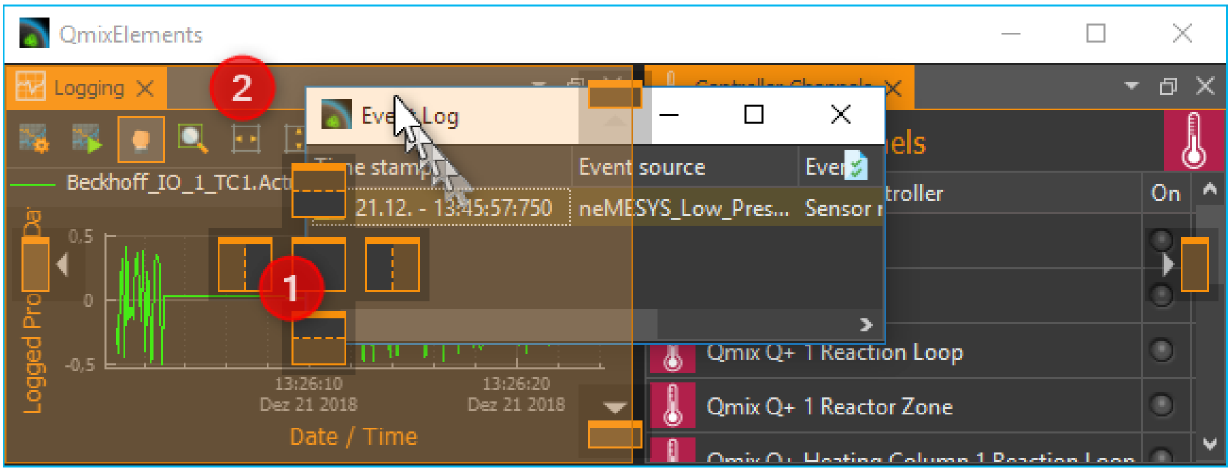


Figure 20: Insert View as Additional Tab

4.9.5 Moving View Groups

Not only individual views can be moved but also complete view groups in which several views are tabbed. To move a complete view group, do not click on the tab of the view, but in the title bar of the group (see figure below). You can now drag and drop the entire group with all tabs from the current position and move it to a new position.

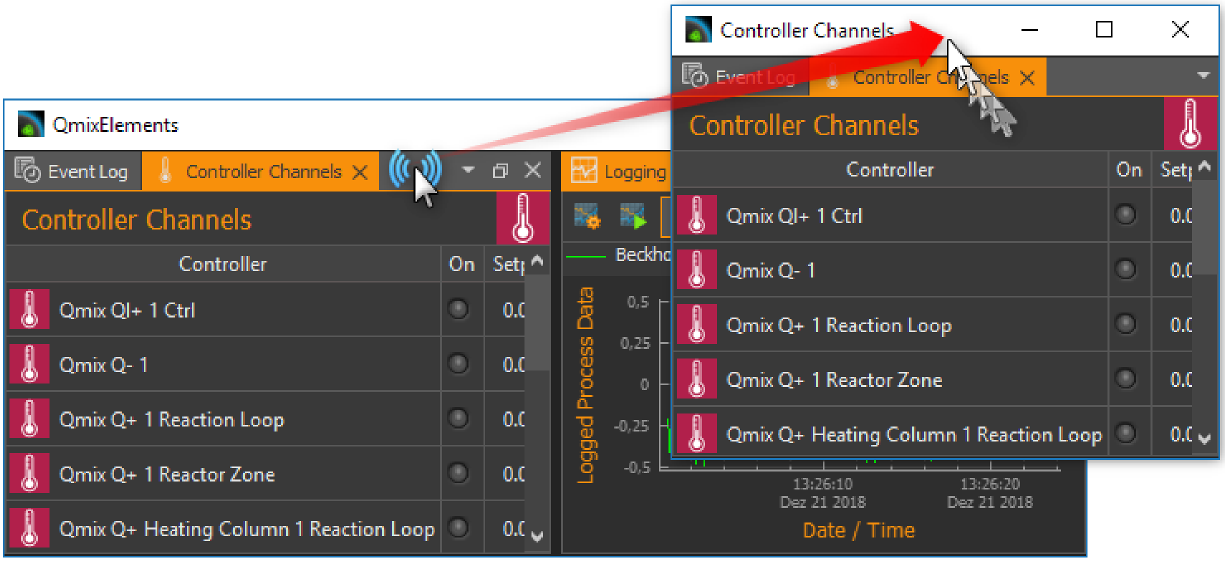


Figure 21: Moving View Groups

Alternatively, you can detach a view group by double-clicking in the title bar 1 of the group, via the menu item *Detach Area* 2 in the context menu or by clicking the *Detach* button in the title bar 3.

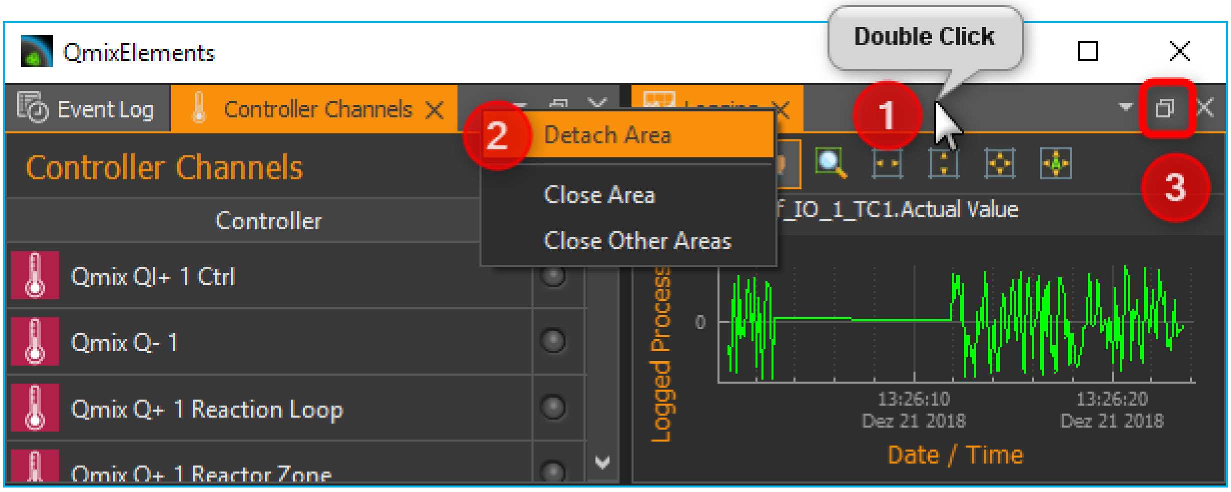


Figure 22: Undock View Group

You can dock a view group or a window with several views in exactly the same way as a window with only one view.

4.9.6 Closing Views and View Groups

You can close a single view either via the Close button 1 in the active tab or via the menu item *Close* 2 in the context menu of a tab.

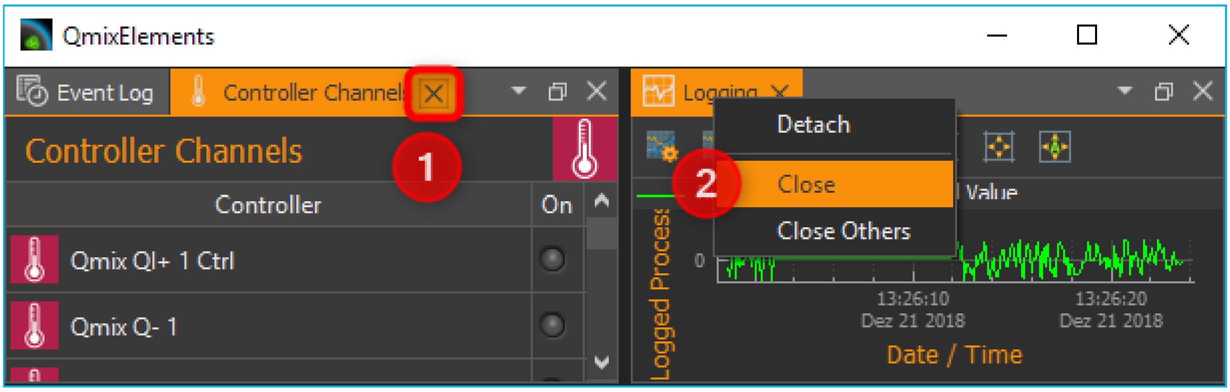


Figure 23: Closing View

A view group can be closed with the Close button 1 in the title bar of the group or with the *Close Area* menu item 2 in the context menu of the view group.

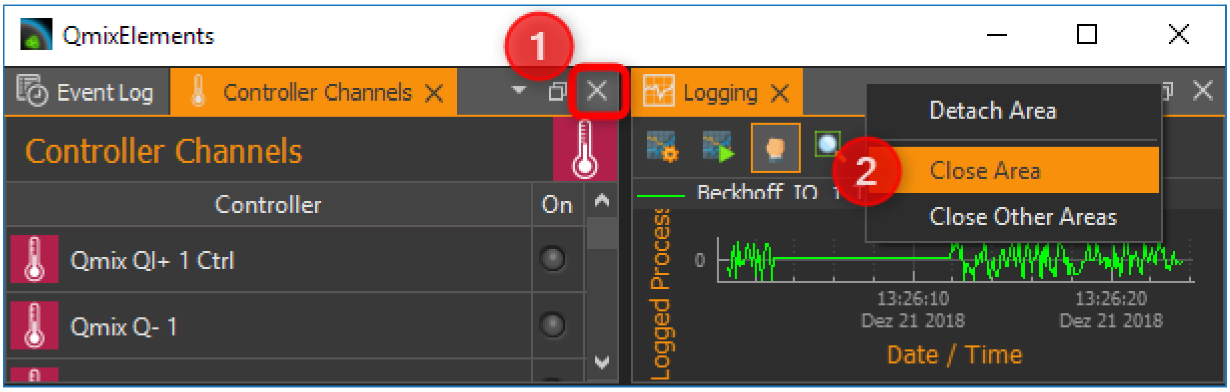


Figure 24: Closing View Group

4.9.7 Useful stuff when working with Views

If you have view groups with a large number of views, it may be difficult to navigate between the individual views in the group due to the number of tabs and the labels that are then truncated. In this case, you can use the corresponding button in the title bar of the view group to call a menu with all views of the group and select the corresponding view.

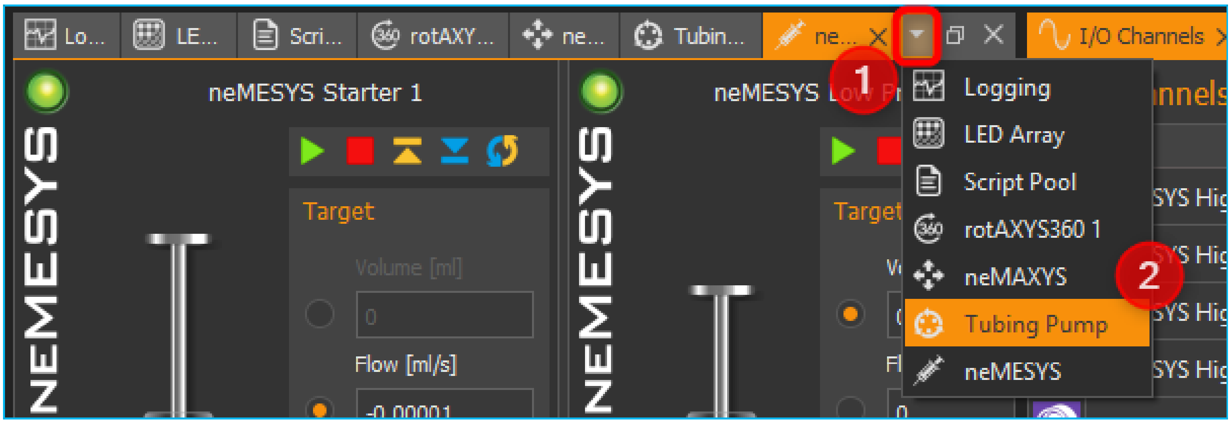


Figure 25: Opening the View Menu of a View Group

If you want to "clean up" a view group quickly, you can use the context menu to close all views except the currently active view. To do this, select *Close Others* from the context menu of the active tab.

Figure 26: Close all inactive views of a group

If you want to quickly "clean up" a window, for example the main window of the application, you can use the context menu of a view group to close all other groups. To do this, select *Close Other Areas* from the context menu of the view group.

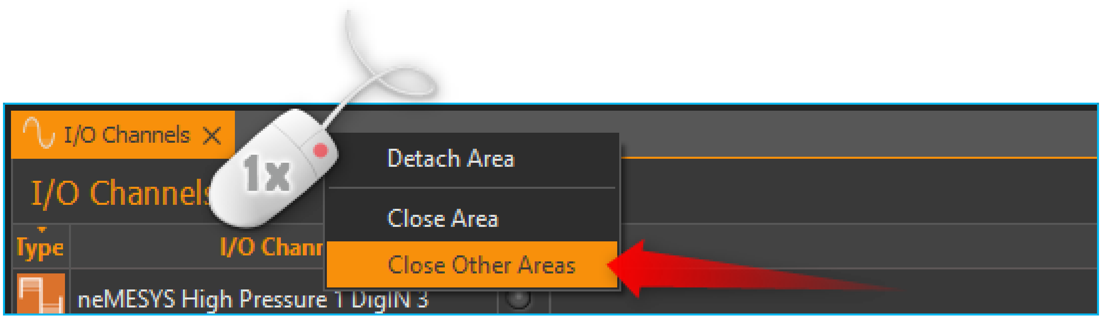


Figure 27: Close all other view groups

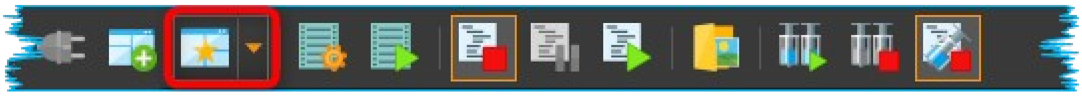
4.10 Perspectives

4.10.1 Introduction to working with perspectives

Different tasks, different devices or even different users require a different arrangement of views. By moving the individual views, you can optimally adapt the interface to your requirements, to the way you work, or to your PC hardware.

Once you have found the perfect arrangement of views for a particular task, you can save this arrangement with a specific name. This stored, names arrangement is called a *perspective* in the software. If you have created several perspectives, you can rearrange the entire interface of the application and adapt it to a new task or requirement with a single mouse click.

You can activate a perspective using the corresponding button in the toolbar.



When you click on the button, the default perspective is loaded. If you click on the arrow next to the button 1 (see figure below), you will open the menu for selecting a previously created perspective 2.

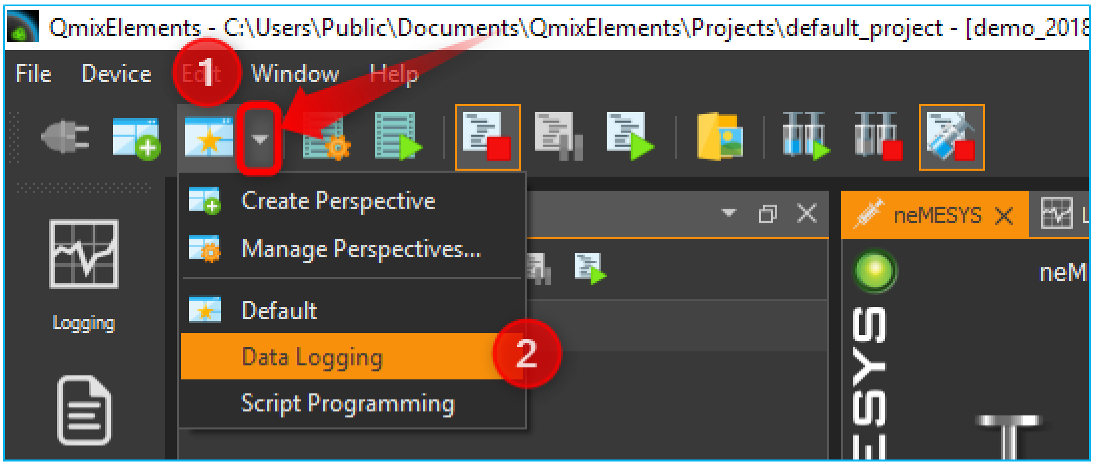


Figure 28: Opening existing perspective

When you click on a perspective, the views in the user interface are arranged in the same way as when the perspective was created. Alternatively, you can access the perspectives from the main menu. To do this, select the menu item *Window* → *Perspective*.

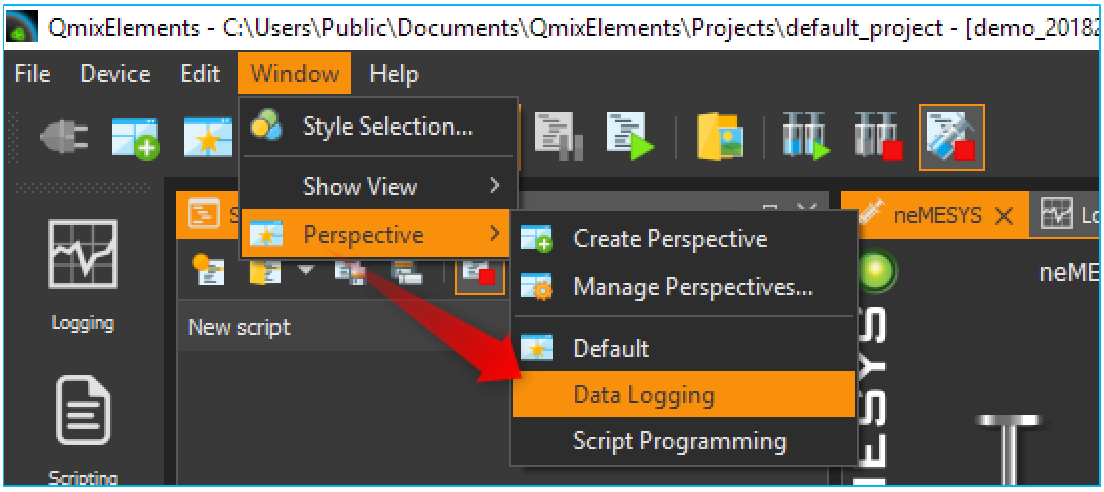
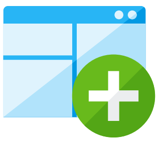


Figure 29: Working with Perspectives from the Main Menu

HINT. If you have lost the overview when arranging the views or if you want to start again with the arrangement of the views, simply use the Perspective button in the toolbar to call the Standard Perspective (*Default*).

4.10.2 Creating Perspectives



To create a perspective, click the *Create Perspective* button in the toolbar. Then enter a unique name for this perspective in the input dialog and click *OK*.

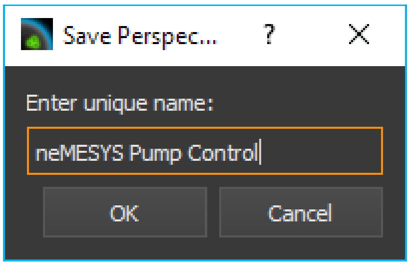


Figure 30: Create Perspective

The new perspective is now available in the Perspective menu.

4.10.3 Deleting Perspectives

To delete perspectives, choose *Manage Perspectives* from the Perspectives menu.

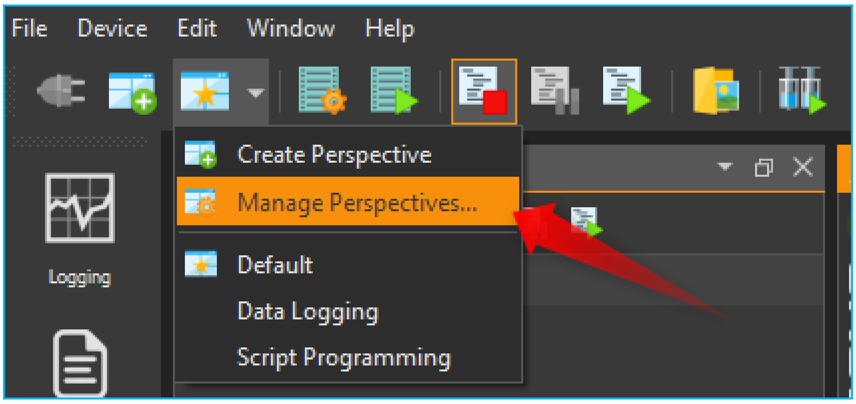


Figure 31: Manage Perspectives

A dialog with a list of all perspectives is now displayed. In the list 1 you can select single or multiple perspectives just like in the file explorer. You can use the *Shift* key to select several related entries. With the help of the *Ctrl* key you can click several entries one after the other and select them.

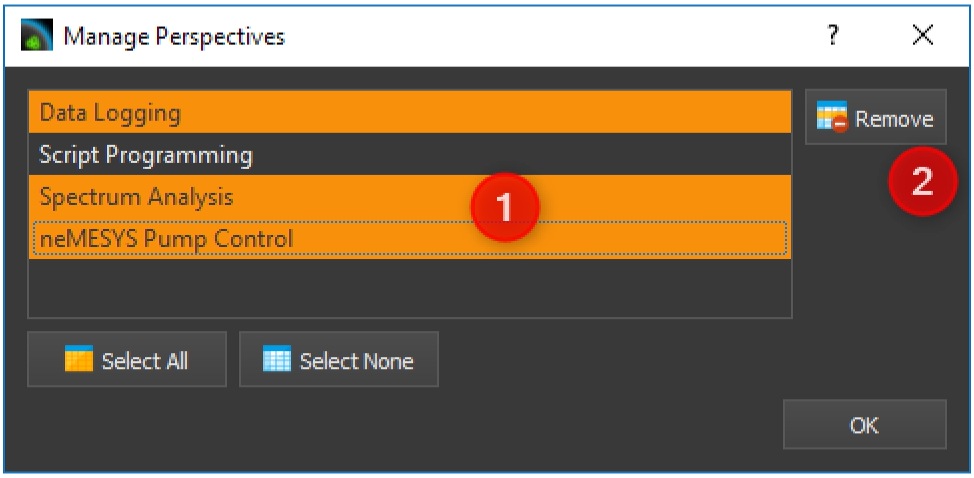
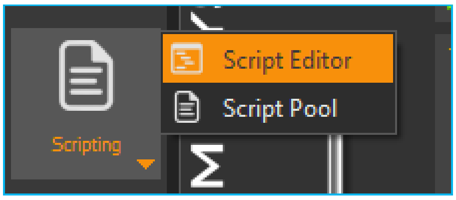


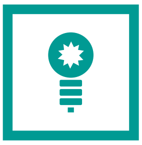
Figure 32: Deleting Perspectives

If you then click the Remove button 2, all selected perspectives will be deleted.

4.11 Sidebar

The sidebar allows you to quickly display different views. Some buttons show a specific view directly. Other buttons allow several views to be displayed and display a menu for selecting the view (see figure below).





HINT. The sidebar, and also all toolbars, can be freely positioned and arranged in the peripheral areas of the main window. If you right-click on a toolbar, you will see a menu for showing and hiding the toolbars.

4.12 Event Log

Important events are displayed in the event log. This includes notes, warnings and error messages. The view with the event log is automatically displayed when an event is written to the log.

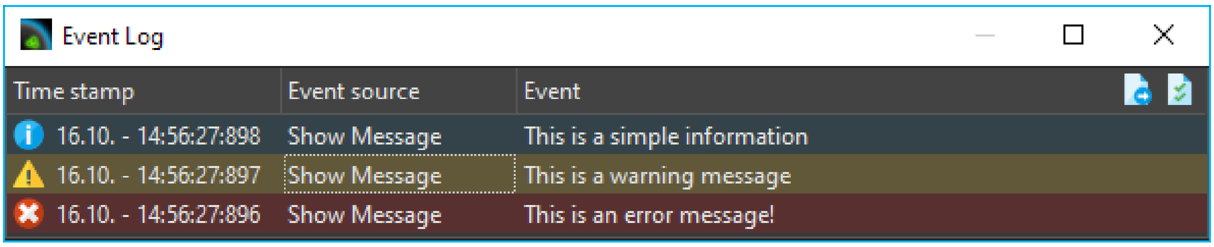
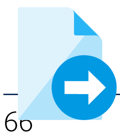
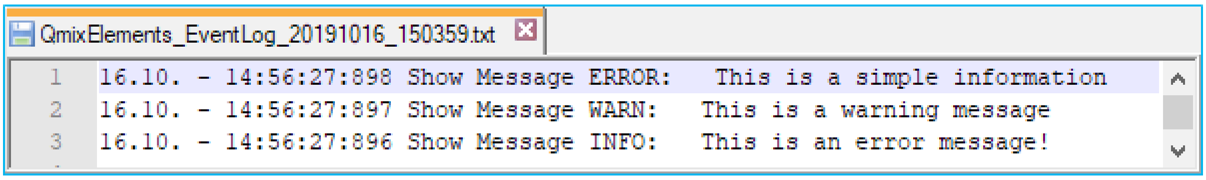


Figure 33: Event Log View

You can clear the event log by pressing the *Clear Event Log* button in the upper right corner of the screen (see figure).



By clicking the *Export To File* button, you can export the contents of the event log to a text file. After you have assigned a file name, the file is saved and you can open it in any text editor.



If you want to display the event log view manually, select *Window* → *Show View* → *Event Log* from the main menu of the application.

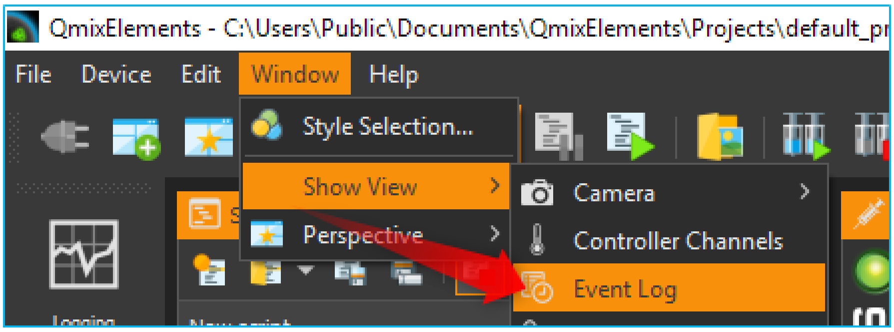
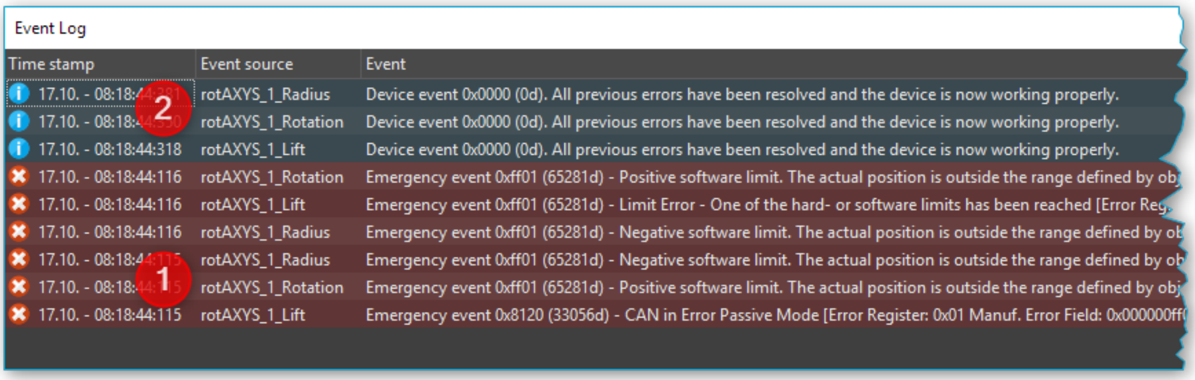


Figure 34: Showing Event Log View

The event log lists the individual events in chronological order. The latest event is always in the first row. This can be seen from the time stamps in the first column.



When connecting to the device hardware, it may happen that device errors are received and displayed in the event log. You can ignore and delete these errors if the event log contains a more recent message from the same event source (the same device) in which the error-free state of the device is reported (example in the figure below).

In this example (see figure above), all 3 axes of the rotAXYS sample handler report an error 1. However, the most recent top three messages 2 show, that there is no more error and that the device works without errors. So you can safely ignore the previous error event.

4.13 Status Bar

The status bar displays a variety of information, such as the connection status to the device.

4.14 Connecting to the Device

In the toolbar, you see the *Connect to device* button signified by the “plug” icon (see figure below). Press this button to connect CETONI Elements with your device(s).

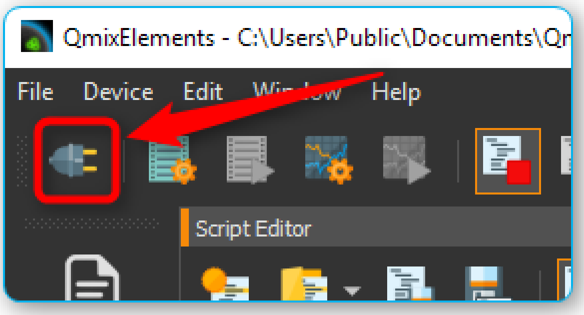


Figure 35: Connecting your devices

IMPORTANT. The device must be connected to the PC via USB and the power supply to the device must be switched on to connect to the device.

4.15 Working with projects

4.15.1 Introduction

If you work with the CETONI Elements software, then you always work in a certain CETONI Elements project. A project stores all project-related data like current device configuration, application-specific device names, syringe and tubing configurations, scaling factors, flow rates, SI units and much more. That is, when you load a project, all user-specific settings are loaded. So you can easily switch between different projects or share projects with other employees.

The title bar always shows your current project path. If you select the menu item *File → Browse Project Folder* in the main menu, the current project will be opened in File Explorer.

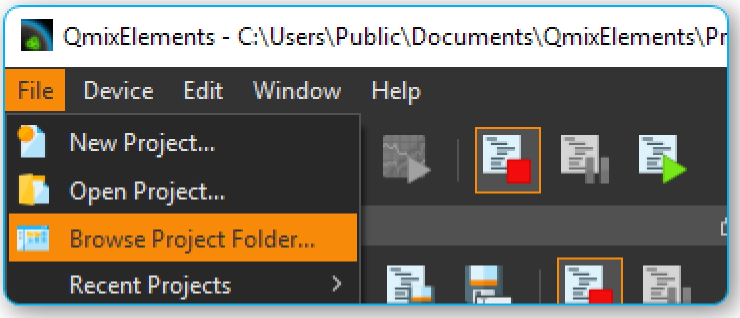


Figure 36: Browse project folder

In the File Explorer you can see the project-specific data of your current project. Each project has a fixed structure of files and sub directories. In the *projectsettings.ini* file all project-specific settings are saved. In addition, the following sub directories exist in the project directory:

- *Configurations* – this directory stores all device configurations that you import or create with the device configurator
- *Data* – this is a directory in which the plugins can store data that can not be saved in the project settings file
- *Log* – used to store log files such as CSV files generated by CSV logger.
- *Pictures* – stores all image files that are recorded in the project (for example, by script

functions or from the camera plugin).

- *Scripts* – all script files that are created in a project are to be stored in this directory
- *Videos* – video files that are generated by the camera plugin will be stored in this directory

4.15.2 Creating a new project

To create a new project, select the menu item *File* → *New Project* from the main menu.

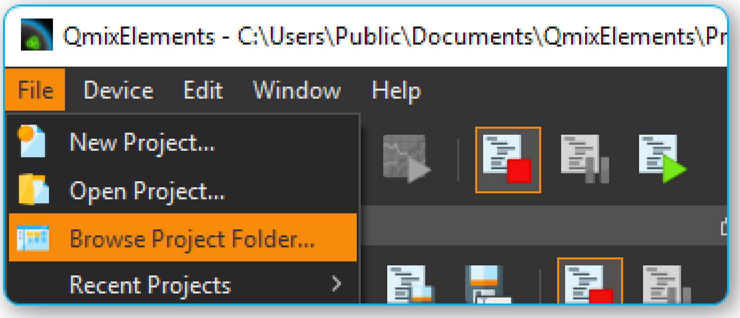


Figure 37: Creating new project

In the dialog window that is then displayed, you can configure the new project. First enter a unique project name 1. Then select whether you want to copy the settings of the current project into the new project (*Copy current project settings*) or whether you want to start with an empty project (*Create empty project*) 2.

Then select the location where the project folder is to be created. If the checkbox *Use default location* 3 is checked, the project will be created in the *Projects* directory of the CETONI Elements data folder. If you want to select a different storage location, for example, to save the project on an external device, remove the check mark 3 and select the location by clicking the *Browse* button 4.

Complete the creation of the new project by clicking OK 5.

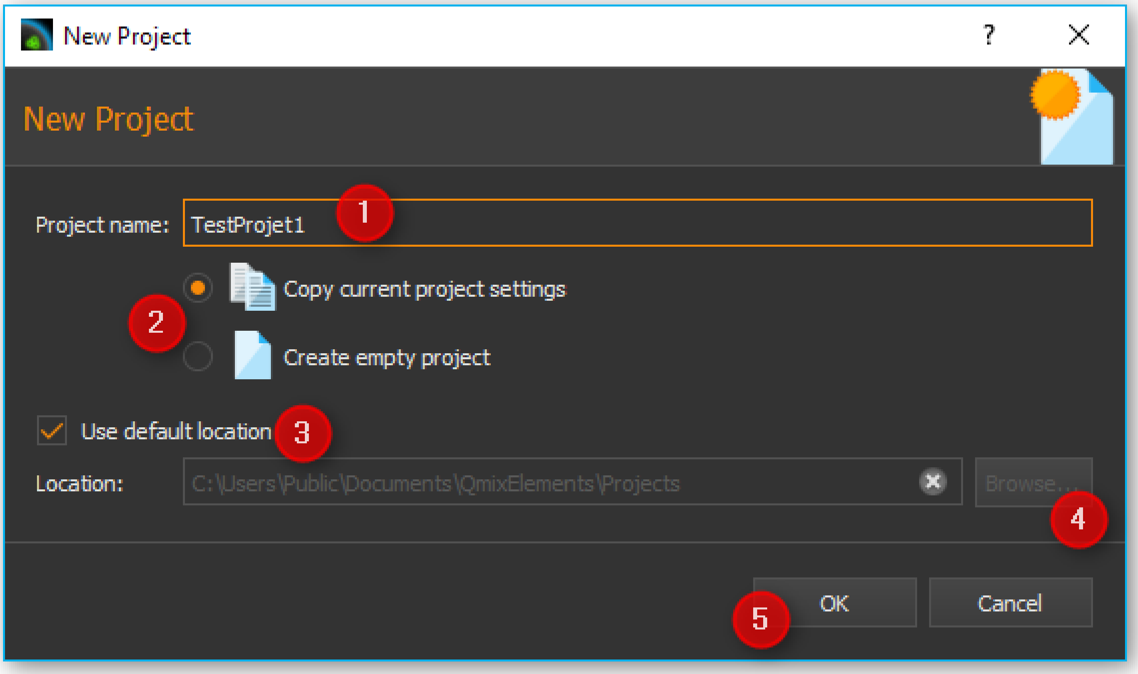


Figure 38: Configuration dialog for creation of a new project

IMPORTANT. Save project data, such as scripts, images, log files, etc., in the project directory or subdirectories whenever possible. This is the only way to ensure that all important data is exported or saved during project export.

4.15.3 Open an existing project

To open an existing project, select the menu item *File → Open Project* from the main menu.

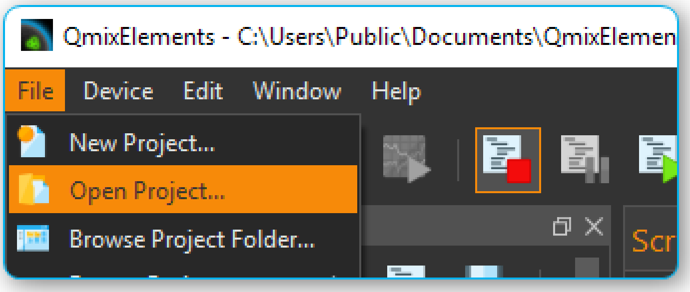


Figure 39: Open an existing project

In the dialog window that is now displayed, you can use the selection box 1 to select a project from the standard project directory. You can also select a project by typing the project name into the selection box. The selection box supports you while typing, through the display of matching projects. If your project is not stored in the default directory, click the *Browse* button ❷. You can then select a project directory with the help of a directory selection dialog window (for example, on an external device).

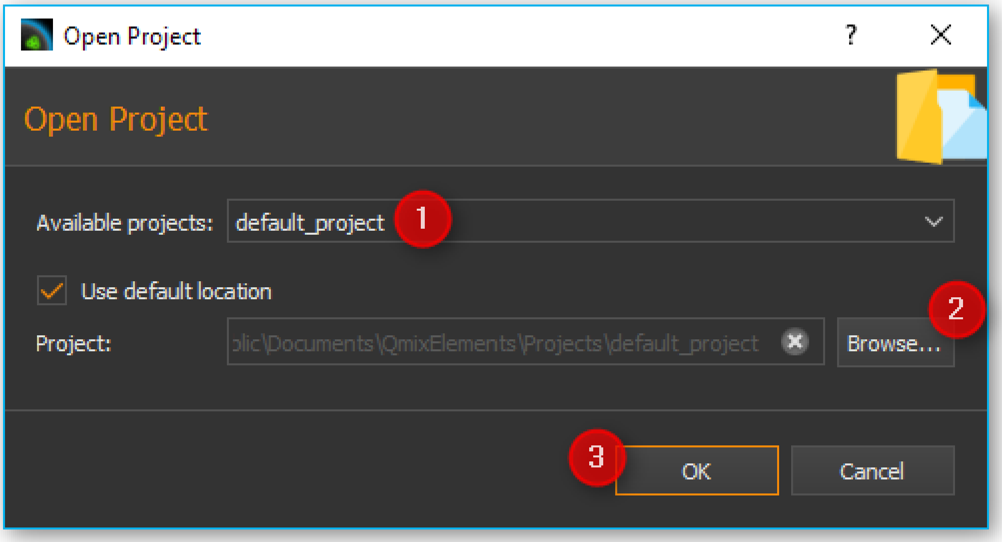


Figure 40: Dialog for opening an existing project

Then open the new project by clicking OK ❸.

4.15.4 Export current project

If you want to archive your project or pass it on to another user, you can do this quickly and easily by exporting the project. During project export, all the data of a project that is in the project directory and which you select for export are stored in a compressed project file (*.qpr).

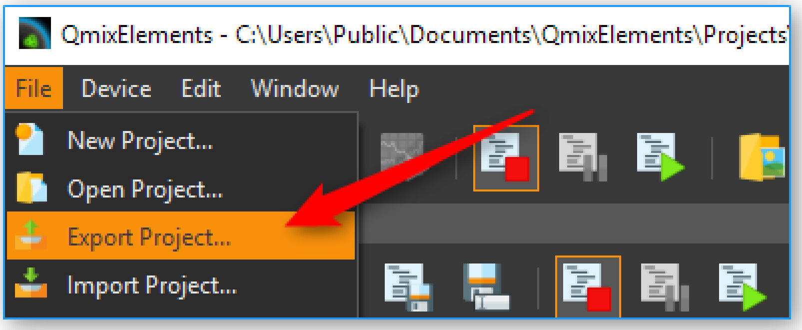


Figure 41: Open the project export window

To export a project, you must first open the project in the software. From the application's main menu, select File → Export Project.

In the export window that appears, you can now select which directories of the project 1 are to be exported.

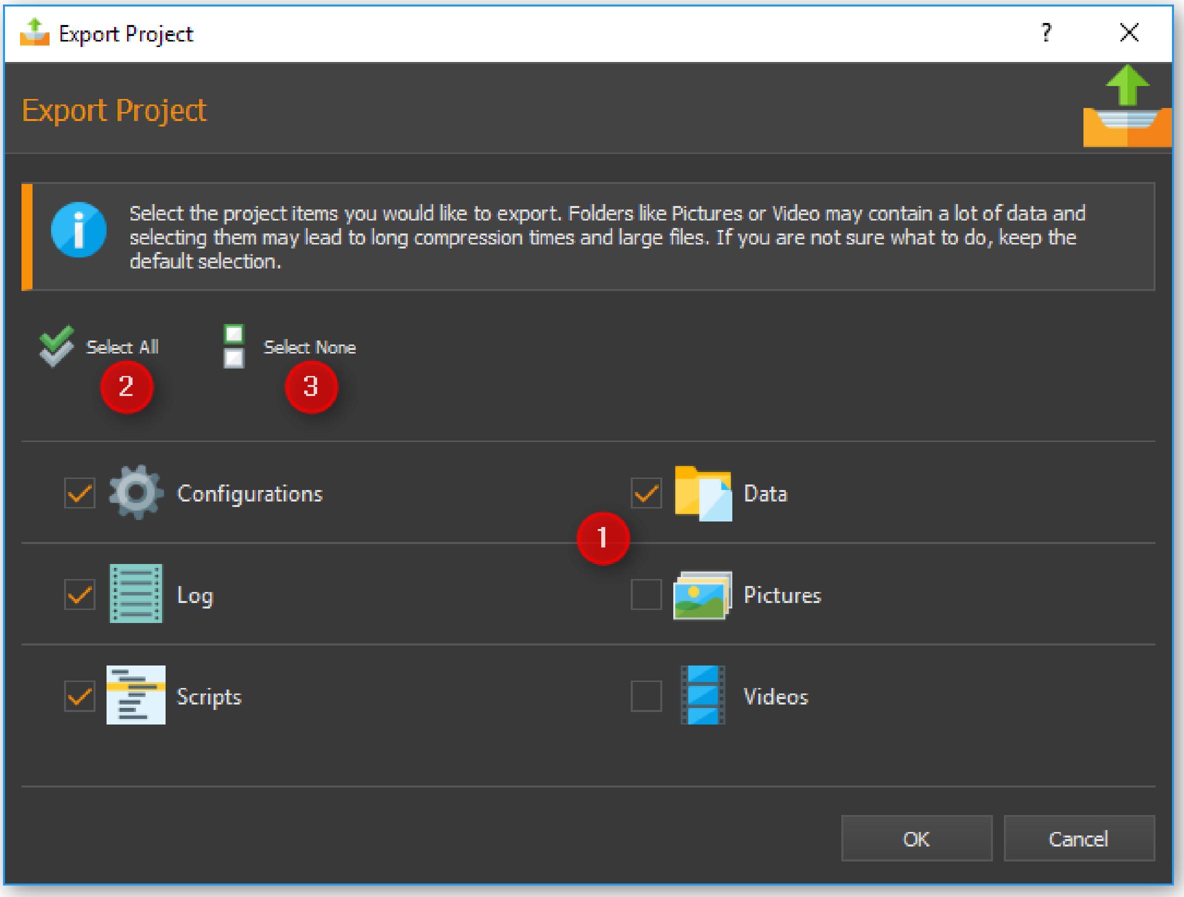


Figure 42: Selection of project directories for export

Click the *Select All* 2 or *Select None* 3 buttons to select all elements or no element.

IMPORTANT. Exporting directories with large amounts of data, e.g. Pictures and Videos directories, leads to large project files and longer times for the export of a project. Export such directories only if it is necessary for the transfer of data.

During the export, you will be informed about the current state of the export by the *Event Log* 1, the *Progress Window* 2, and the Status Line 3.

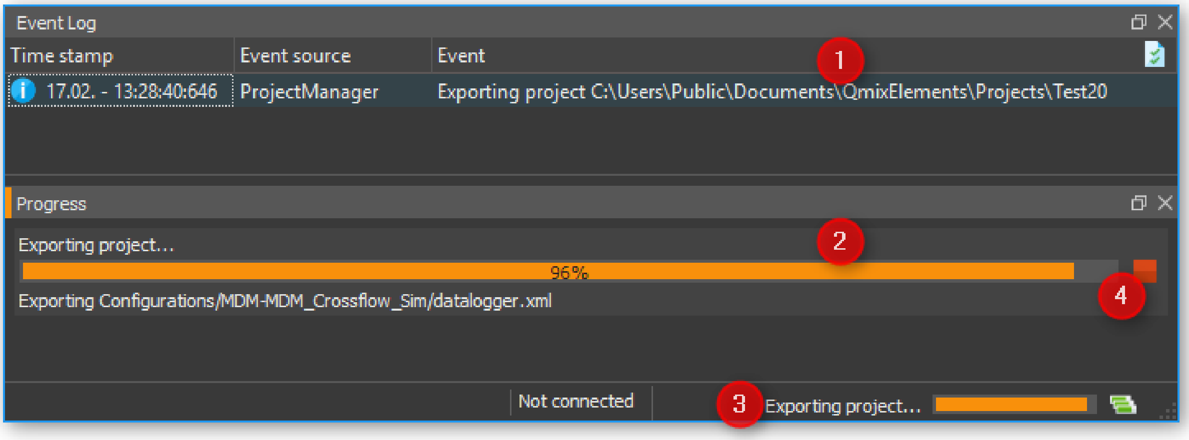


Figure 43: Status display project export

You can cancel the export at any time by clicking the *Cancel* 4 button.

4.15.5 Importing projects

The import function allows you to import project files (*.qpr) into the software. To do this, select File → Import Project from the main menu.

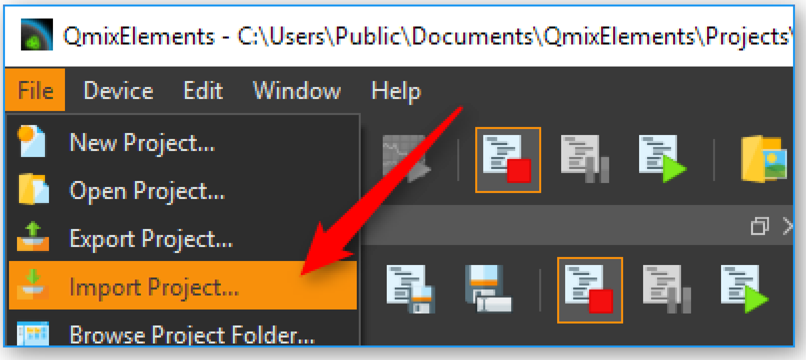


Figure 44: Selecting project import

In the file selection window that appears, select the project file (* .qpr) that you want to import. You will then see a window where you enter the name under which the imported project should be stored in your CETONI Elements project directory.

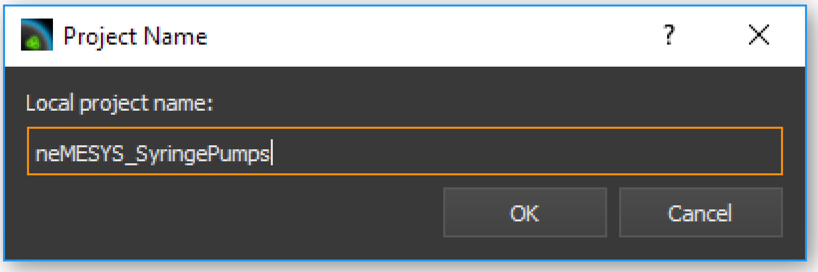


Figure 45: Enter project name

IMPORTANT. Assign a unique project name. If you use a name for which a project already exists in the project directory, an import is not possible.

During the import, you will be informed about the current state of the import by the *Event Log* 1, the *Progress Window* 2 and the *Status Line* 3.

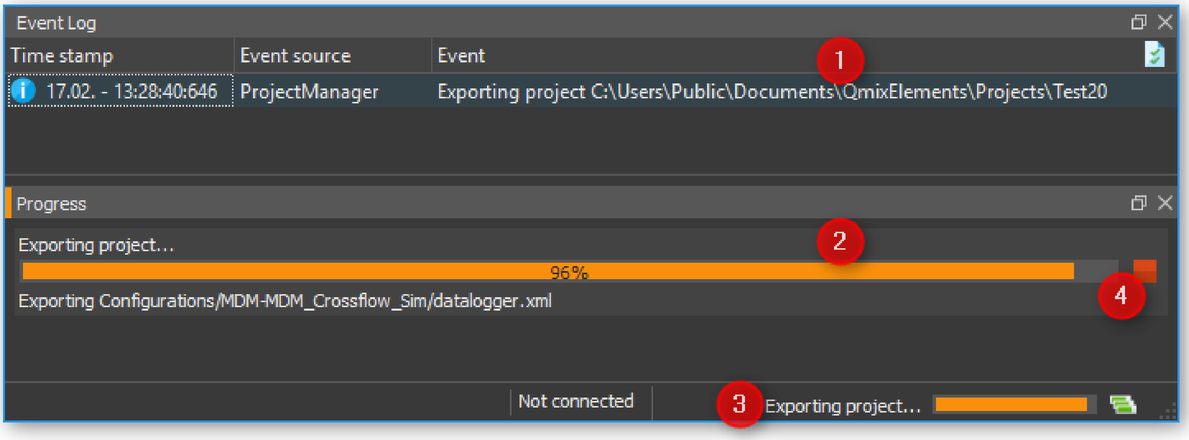
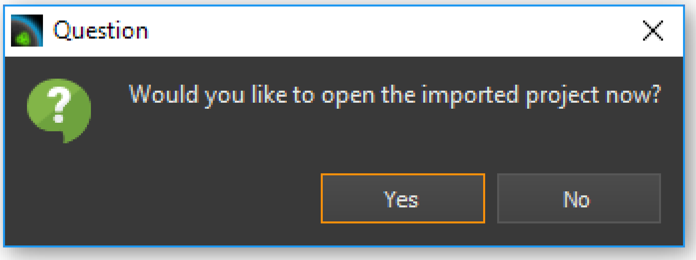


Figure 46: Status display project import

You can cancel the import at any time by clicking the *Cancel* 4 button. After successfully importing the project, the software will offer you to load the imported project.



Click *Yes* to restart the software with the imported project.

4.16 Global Settings

4.16.1 Open settings dialog

The configuration dialog for the global application settings can be opened via the menu item *Edit* → *Settings* in the main menu of the application.

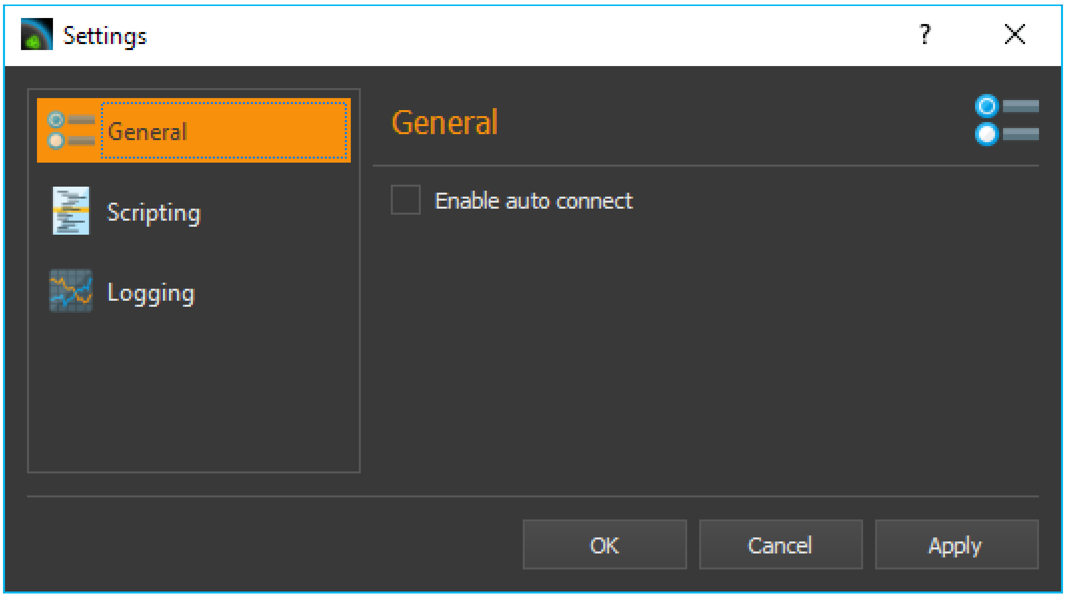


Figure 47: Settings dialog

On the left side you will find the different settings categories. Select a category for which you want to configure the global settings. On the right side you will see the settings that you can configure for the selected category.

Click *Apply* to accept the changes or *OK* to accept the changes and close the dialog.

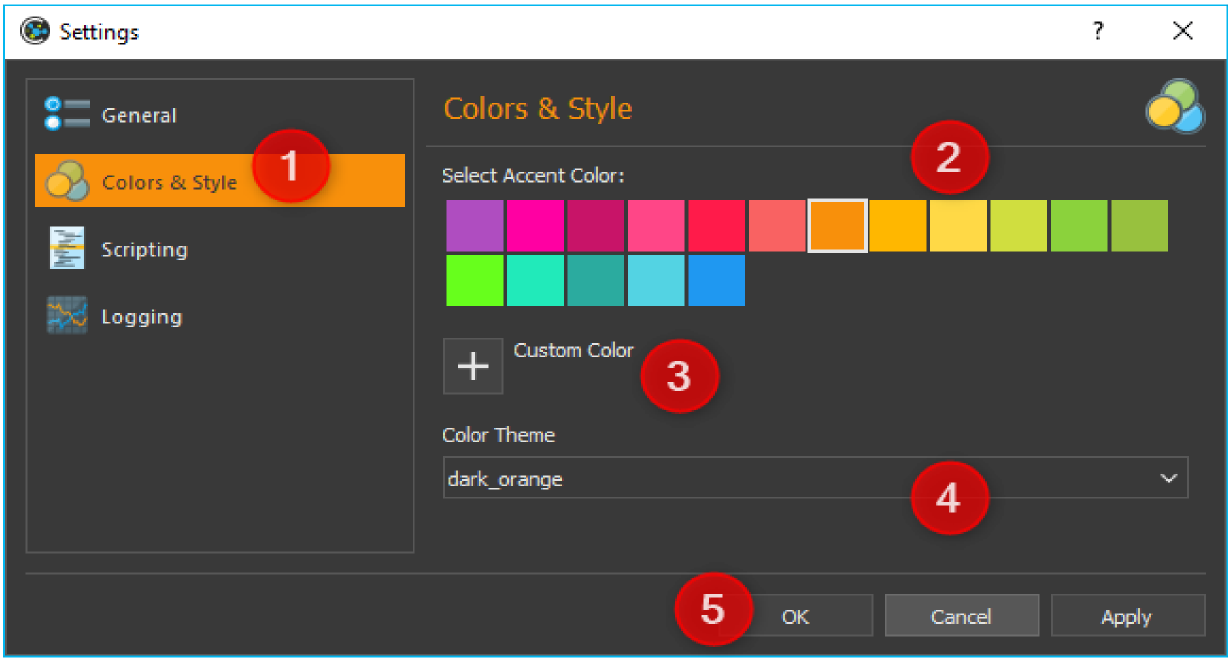
4.16.2 General Settings

4.16.2.1 AUTO CONNECT

If this option is enabled, CETONI Elements automatically connects to the attached devices after startup. If you register CETONI Elements.exe in the Windows *Autostart*, you can start the CETONI Elements software fully automatically after switching on the computer and establish a connection to the device without any user intervention being necessary.

4.17 Personalize Color and Style

Via the menu item *Edit –> Settings* in the main menu of the application you open the configuration dialog of the application. There you will find the *Color & Style* 1 section with the settings for customizing the design of the graphical interface.

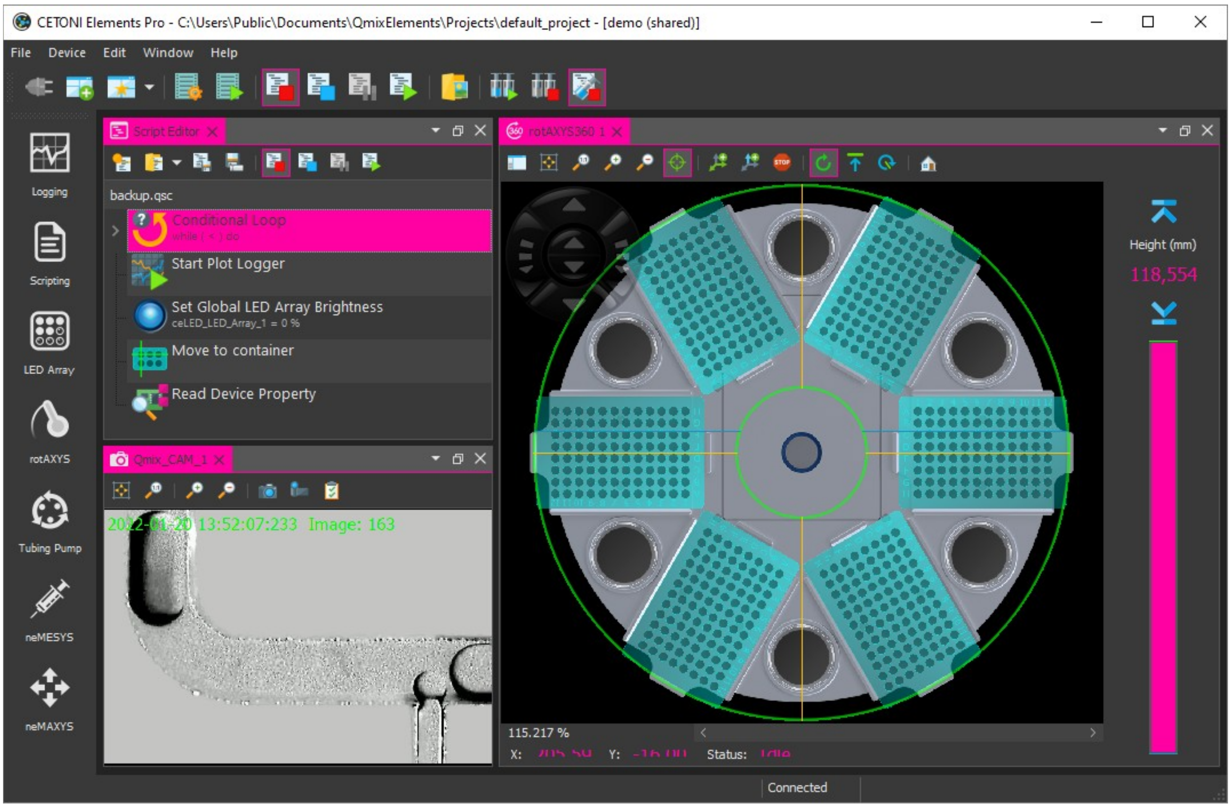


You can customize accent color by selecting a color from the palette of predefined colors 2 or by defining your own color using the *Custom Color* button 3.

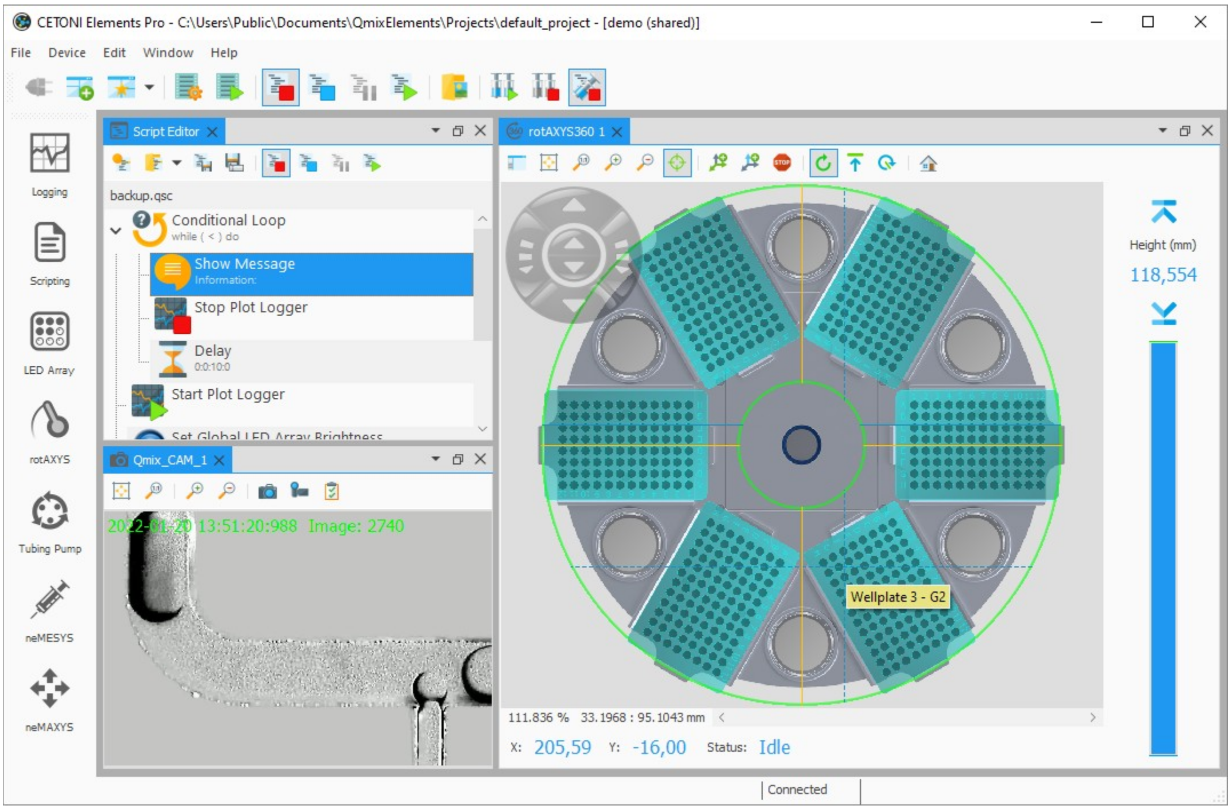
IMPORTANT. Adjusting the graphical user interface after a change blocks the application for a short time. You should therefore not perform these changes if you are working with devices or if a critical control application / script is active.

Using the *Color Theme* selection box, you can switch the complete color design of the application between different themes. For example, you can switch between a Light Theme and a Dark Theme to adapt the application to your wishes or to the corporate identity of your company. The following two screenshots show examples of a customized interface:

Example 1: Dark design with alternative accent color



Example 2: Light design with blue accent color



4.18 Restricting Access to the User Interface (UI)

4.18.1 Introduction

The software allows you to restrict access to control elements in the software using the UI-Access-Manager. You can configure which elements of the application are hidden for users. This may be desired, for example, if you have written a script for other users or employees. If you want to ensure that the devices are only operated via the script and not manually via the user interface controls, you can hide the relevant controls.

4.18.2 Configuring Access to User Interface Elements

To configure access to the software interface, select *Edit*→ *Configure UI Access* from the main menu.

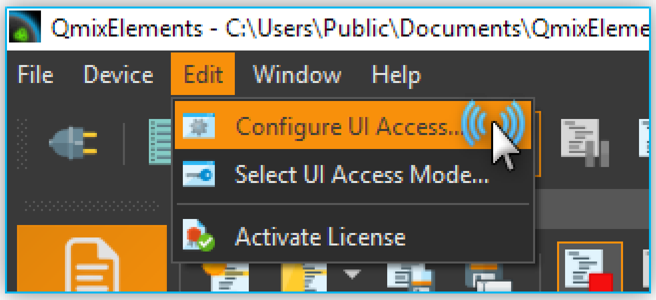


Figure 48: Open UI Access Configuration

The configuration dialog for UI access is displayed. In the first column 1 you can see the elements of the user interface that you can hide and show. Within the individual groups (e.g. workbenches) the elements are sorted alphabetically.

In the second column 2 you can see the configuration of the *Default* mode. This mode is always active when the software is started. I.e. this mode is the mode that every user sees without special authorizations. For each element, you can choose whether it is visible or hidden in the interface. This way you can easily hide controls from normal users.

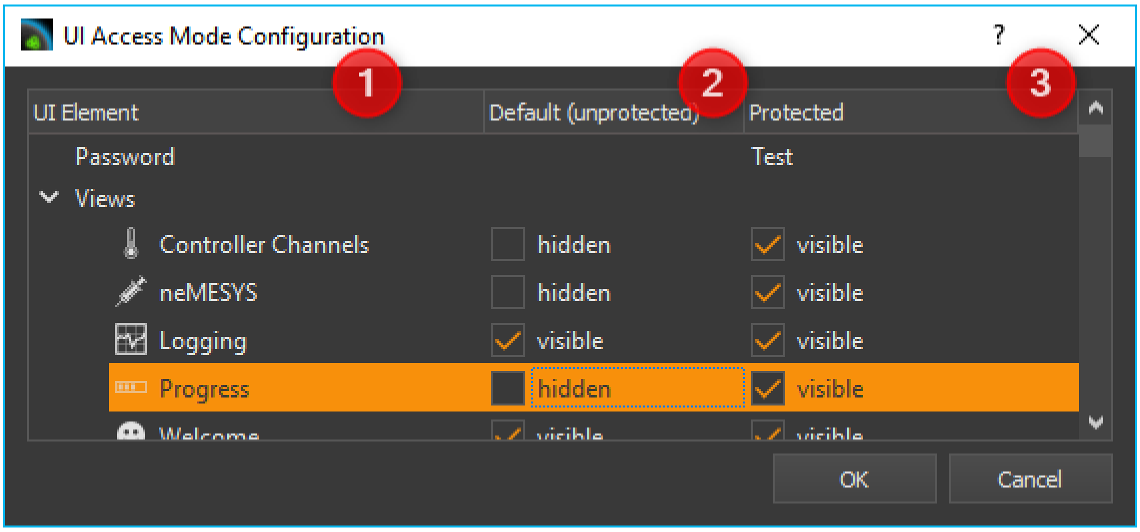


Figure 49: Configuration dialog for UI access configuration

The third column **3** contains the configuration of the interface for access-protected mode. This mode is password protected. This means that you can access this mode by entering the password and thus show previously hidden operating elements.

HINT. For the protected mode, you should set all control elements to visible in order to have access to them if necessary.

You can set the password for the protected area by double-clicking in the corresponding cell of the third column and entering the password.

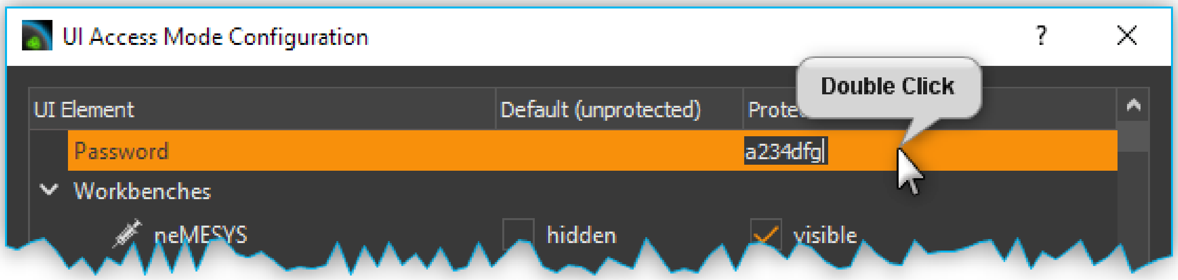


Figure 50: Enter password for UI access mode selection

When you have finished the configuration, click the *OK* button to close the dialog and save the configuration.

IMPORTANT. The configuration is saved in the current project. This means that the configuration of the interface is project-specific and not global.

4.18.3 Switching UI Access Mode

After starting the application, the standard mode is always active. This means that the interface is displayed with the configuration that you have configured for standard mode. To enter protected mode, select *Edit* → *Select UI Access Mode* from the main menu.

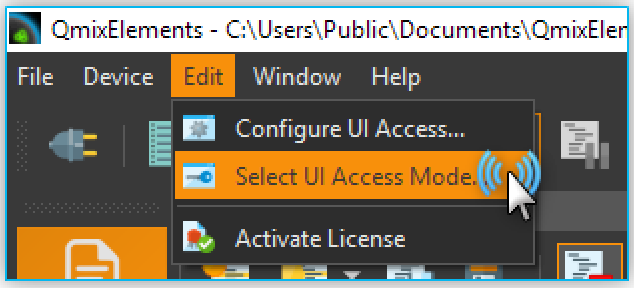


Figure 51: Switching UI Access Mode

After selection, the interface is immediately adapted according to the selected configuration. If you want to leave the protected mode, open the selection dialog again and then select the default configuration.

5 Creating Device Configurations

5.1 Introduction

In order to combine single devices according to your requirements and use them in CETONI Elements, you have to create a device configuration. Creating device configurations can be done using the *Device Configurator* which is integrated in the CETONI Elements software. The Device Configurator features the *Device Configuration Wizard* that guides you through the integration of your devices into your device configuration.

5.2 Starting the Device Configurator

You can start the Device Configurator by selecting the menu item *Device* → *Create* Configuration from the main menu.

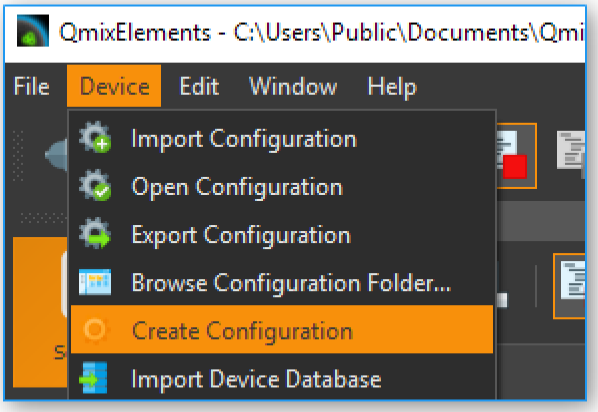


Figure 52: Opening the Device Configurator

5.3 Import a device database

The device configurator and the configuration wizard use a device database (*devices.db*) to read device-specifc parameters and settings for configuration of the devices.

During the installation of the CETONI Elements software this database is already installed. If you use devices that are not yet listed in the device database (e.g. if your devices are newer than the latest CETONI Elements software version), you may need to import an updated version of the device database. This database can be found either on your installation media or your get it directly from CETONI.

When you start the device configurator, you will be asked if you want to import a new device database. If your installation media contains the file *devices.db*, you should import it now.

HINT. If your installation media does not contain a device database, your devices are already supported by the CETONI Elements software and you can skip the device database import.

(1) Confirm the query by left-clicking the *Yes* button.

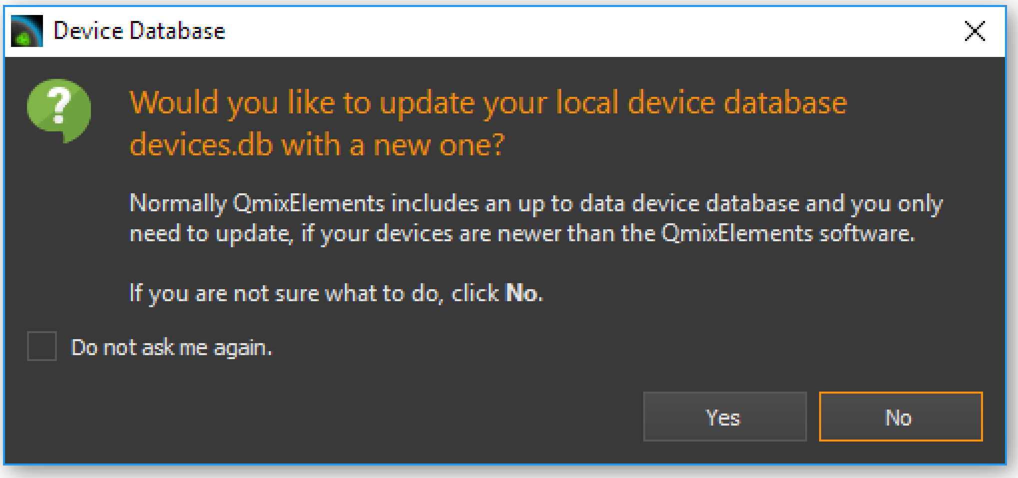


Figure 53: Query for device database import

(2) A file selection dialog opens. Select the database file *devices.db* located in the root directory of the CETONI Elements CD (see figure below).

(3) The device configurator now has an actual device database available.

IMPORTANT. The device database contains important configuration information about your devices. You always have to import an actual device database if you purchase new devices from the cetoni GmbH.

As long as you don't purchase new devices you needn't import a device database on start of the device configurator. If you don't want CETONI Elements to further ask for database import just check *Do not ask me again* and confirm by left-clicking *No* (see figure below).

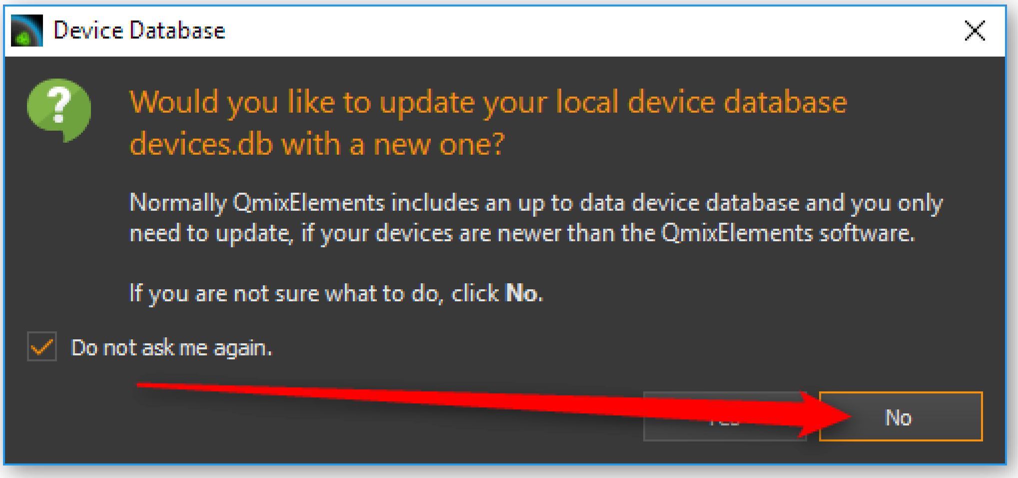


Figure 55: Deselect device database import

If you want to import a device database in the future you can reach the import dialog by selecting

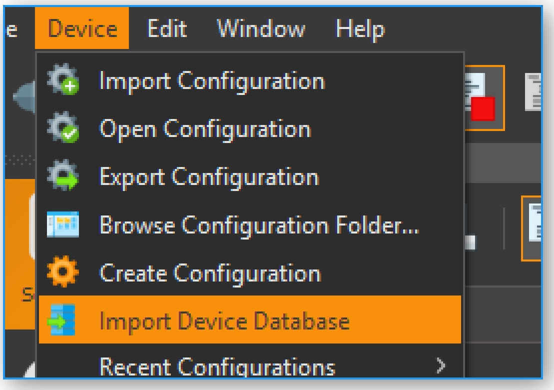


Figure 56: Import device database menu item

Device → Import Device Database from the main menu (see figure below).

HINT. You don't have to import a device database on each start of the device configurator. As long as you don't purchase any new devices, you don't need to import a new device database.

5.4 Activating the Device Configurator

To use the Device Configurator, you need a valid license. If you do not have a license, you can test the creation of device configurations, but you cannot configure your devices using the Configuration Wizard 1. A missing license is indicated by a warning message in the status bar of the Device Configurator 2 (see figure below). The activation of CETONI Elements is described in the chapter Activate License.

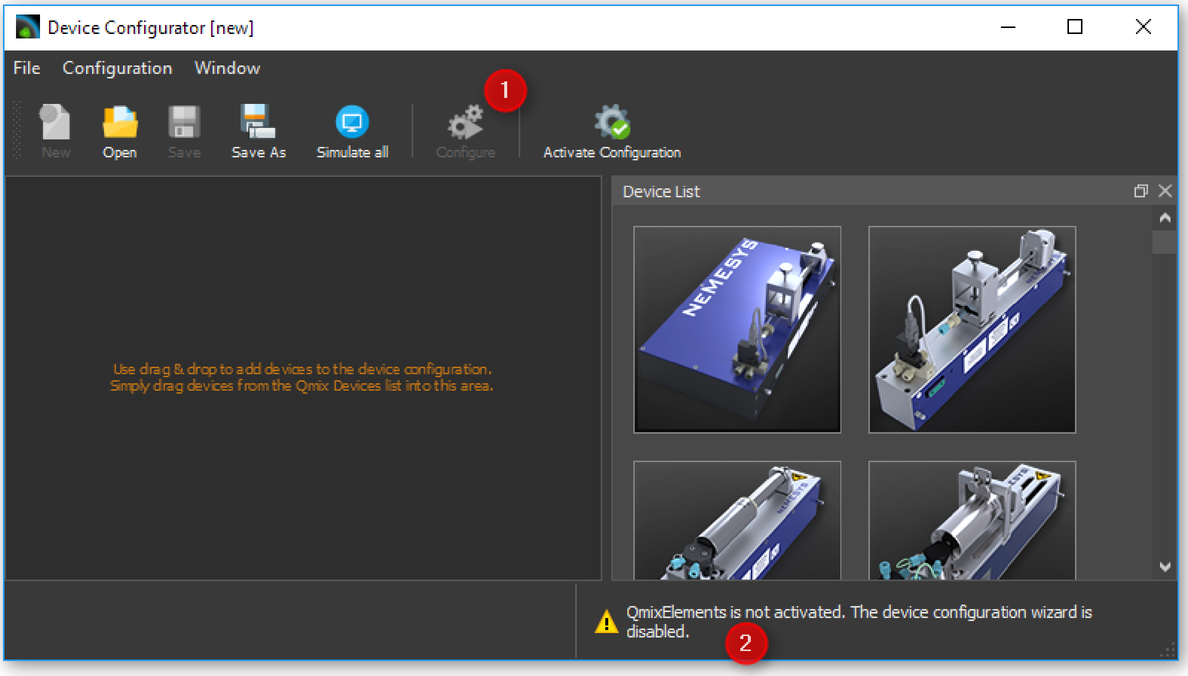


Figure 57: CETONI Elements with missing Device Configurator license

5.5 Overview

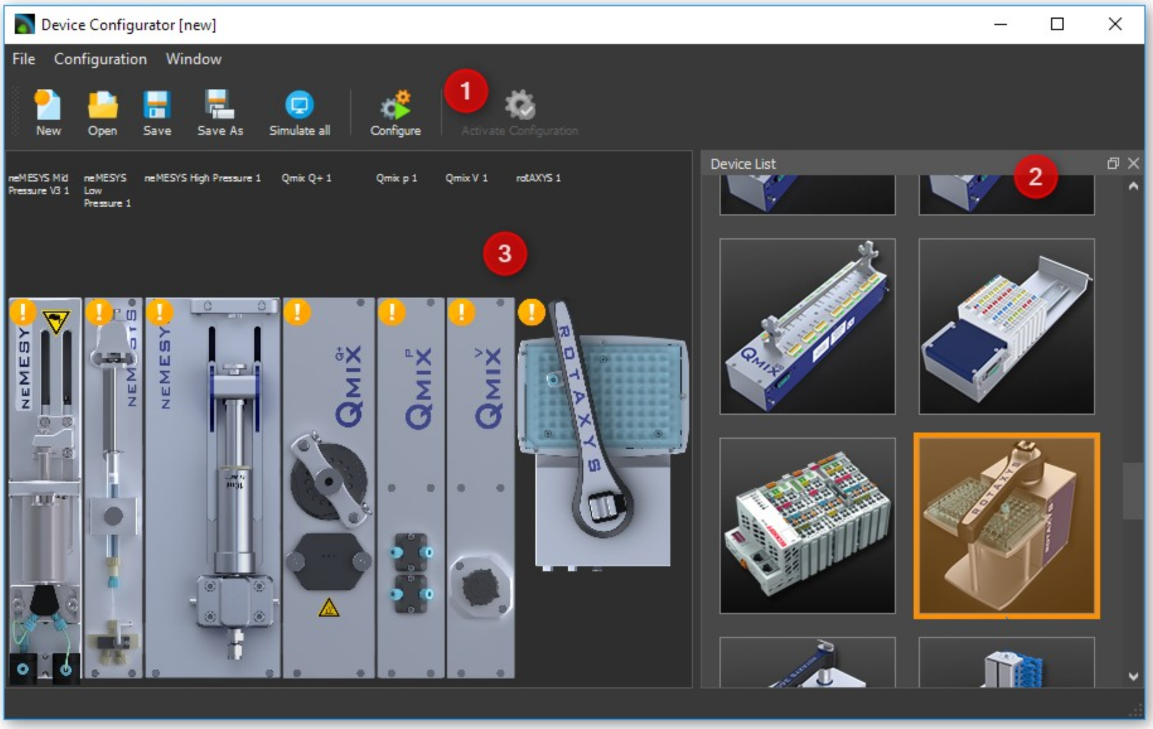


Figure 58: Device Configurator overview

The Device Configurator is divided into the following three parts:

1 Toolbar – Functions for creating, loading and saving device configurations, as well as starting the Device Configuration Wizard.

2 Device List – Selection of available devices.

3 Device configuration created by user.

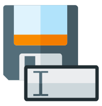
5.6 Toolbar

The toolbar contains the following functions:

Creates a new, empty device configuration.

Loads an existing device configuration.

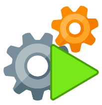
Saves the created device configuration.



Saves the created device configuration using a new name.

Sets all devices of the created device configuration to simulated.

Starts the add-on plugins selection dialog. These are optional software extensions that CETONI has developed customer-specific, possibly on your behalf.



Starts the Device Configuration Wizard.

Restarts the software with the device configuration from the device configurator

5.7 Device Configuration Wizard

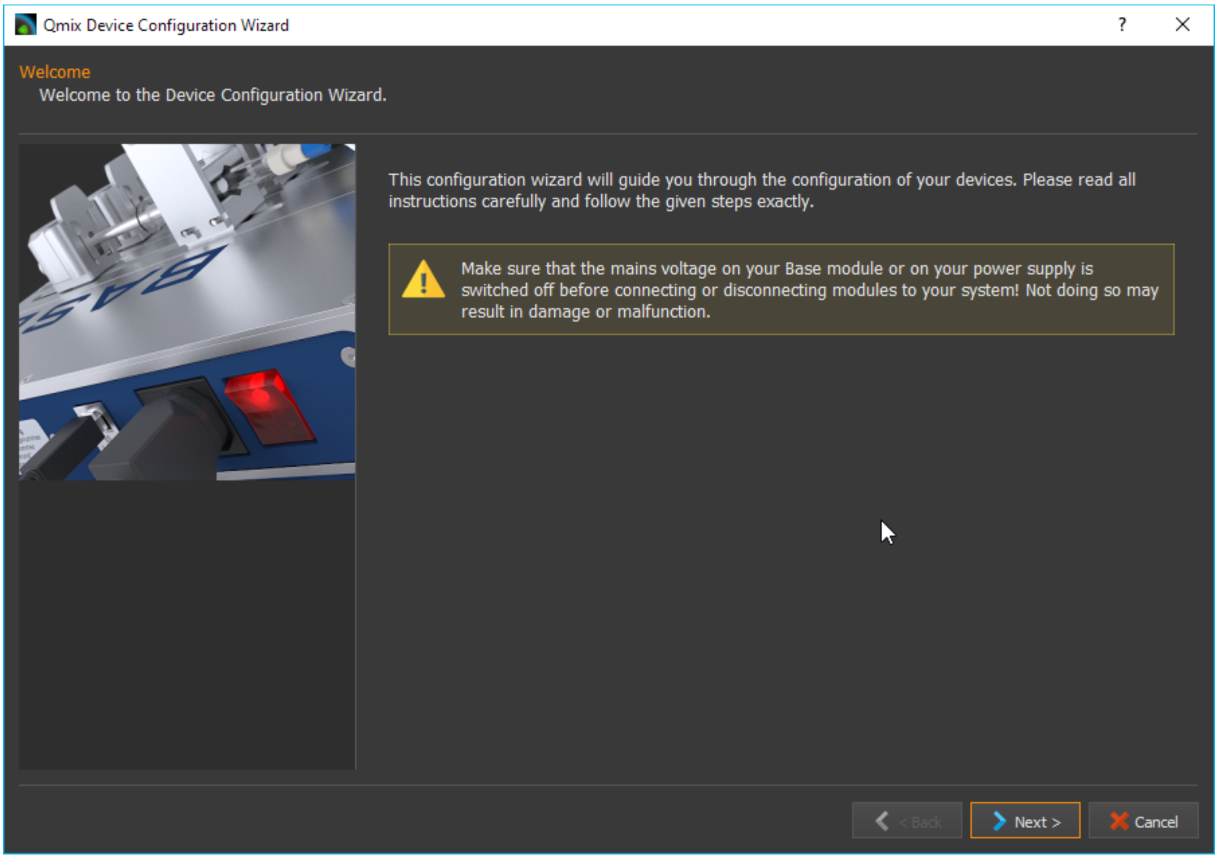


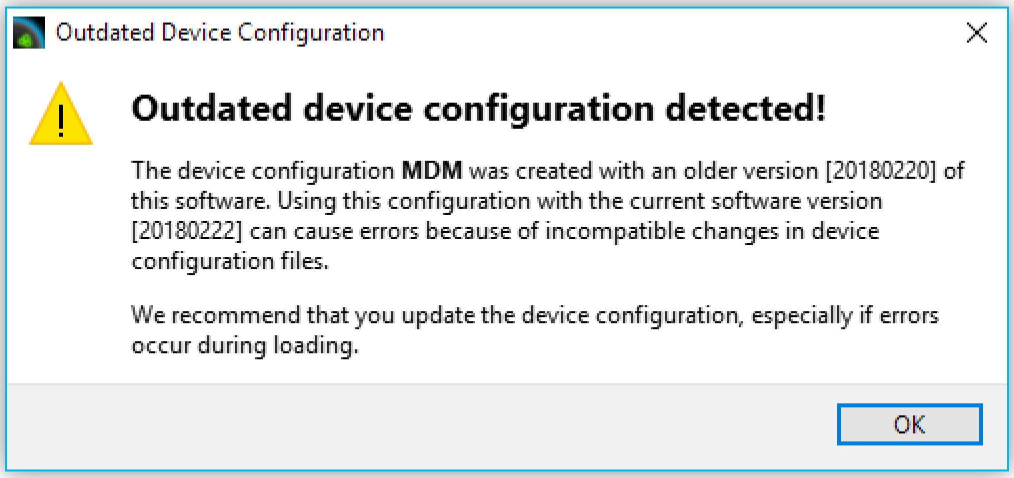
Figure 59: The Device Configuration Wizard

The *Device Configuration Wizard* guides you through the integration of your devices into your device configuration. You must execute the Device Configuration Wizard after:

- creating a new device configuration or changing an existing one.
- swapping devices or integrating a new device into an existing device configuration.
- an update of the CETONI Elements software

IMPORTANT. Device configurations and devices belong together. Always execute the Device Configuration Wizard if you want to change a device configuration or integrate a new device into an existing device configuration.

Your device configuration should always be created with the software version with which you want to use the configuration, otherwise errors may occur. If the device configuration was created with another software version, you will receive a warning message when the application is started.



If you see this warning, you should open the device configuration in the *Device Configurator*, save it and then configure all devices using the *Configuration Wizard*.

5.8 Creating a sample device configuration

The following example describes the creation of a device configuration containing a Nemesys S dosing module and a Qmix Q+ module.

(1) Start the Device Configurator as described in section 5.2.

(2) Left-click on the Nemesys S dosing module listed in the device list 1 and hold down the mouse button.

(3) Drag the Nemesys dosing module into the device configuration 2 holding the mouse button down. As soon as you release the mouse button the Nemesys dosing module is added to the device configuration. Alternatively, you can add a device by double-clicking on it in the device list.

HINT. You can add devices via drag & drop or by a double click.

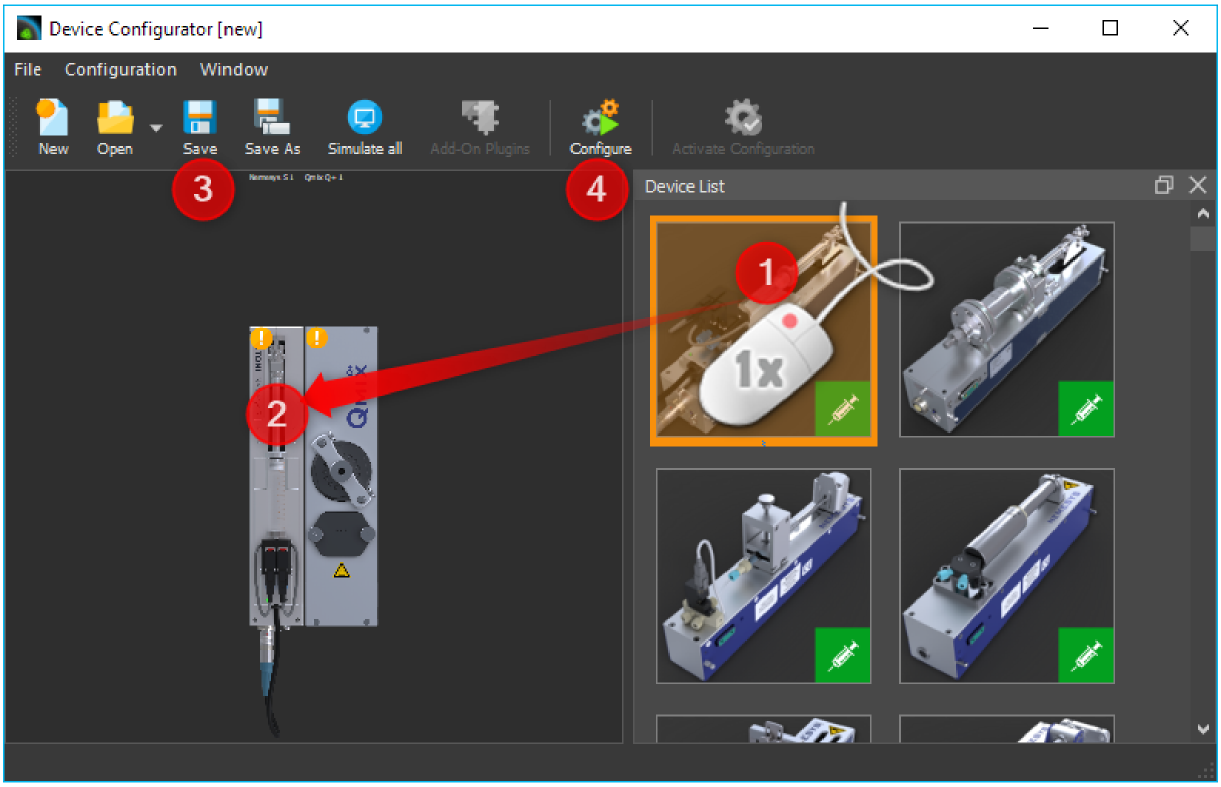


Figure 60: Creating a device configuration

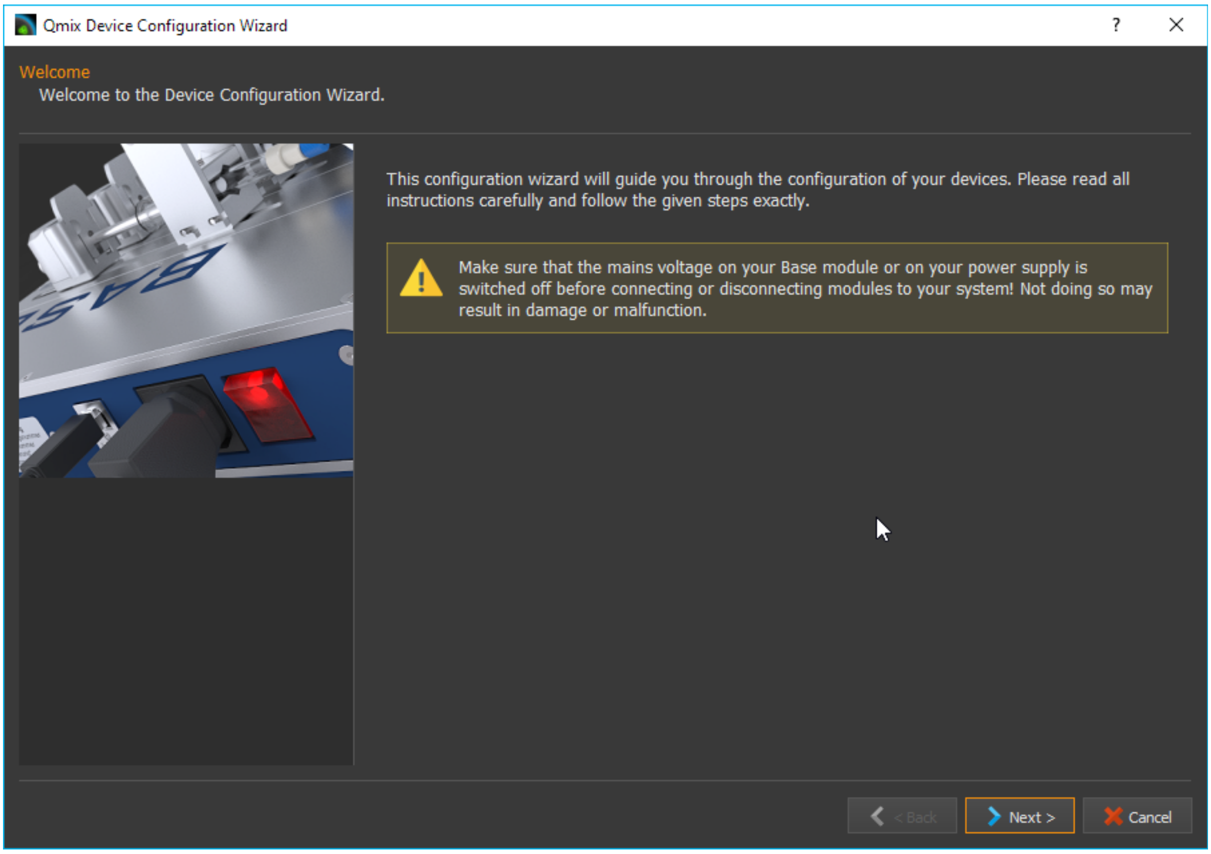
HINT. You can remove devices from the device configuration by selecting the device by clicking it and then pressing the Delete key.

(4) Proceed in the same way with the Qmix Q+ module.

(5) Save your configuration under a meaningful name by clicking on the *Save* button. 3.

(6) Start the Device Configuration Wizard by left-clicking *Configure* 3.

ATTENTION. Make sure that the mains voltage on your Base module or on your power supply is switched off before connecting or disconnecting modules to your system. Not doing so may result in damage or malfunction.



(7) The Device Configuration Wizard advices you to follow these instructions:

- Ensure that the Nemesys S module which is to be configured is connected to your Base module. Turn the Nemesys S power switch ON. The blue Power LED should now light up
- Switch off all other modules that have a power switch
- Remove all other modules from the setup, that do not have a power switch – in this case, that would be the Qmix Q+ module (Attention: Switch of the mains voltage at the Base module before removing or adding modules)
- Ensure that the terminator is plugged into the last module of your setup – the Nemesys S in this case
- Switch on the mains power switch on your Base module or on your power supply unit.
- Check that the Nemesys S module is really the only device that is powered on now. The blue power LED of all other modules must be off

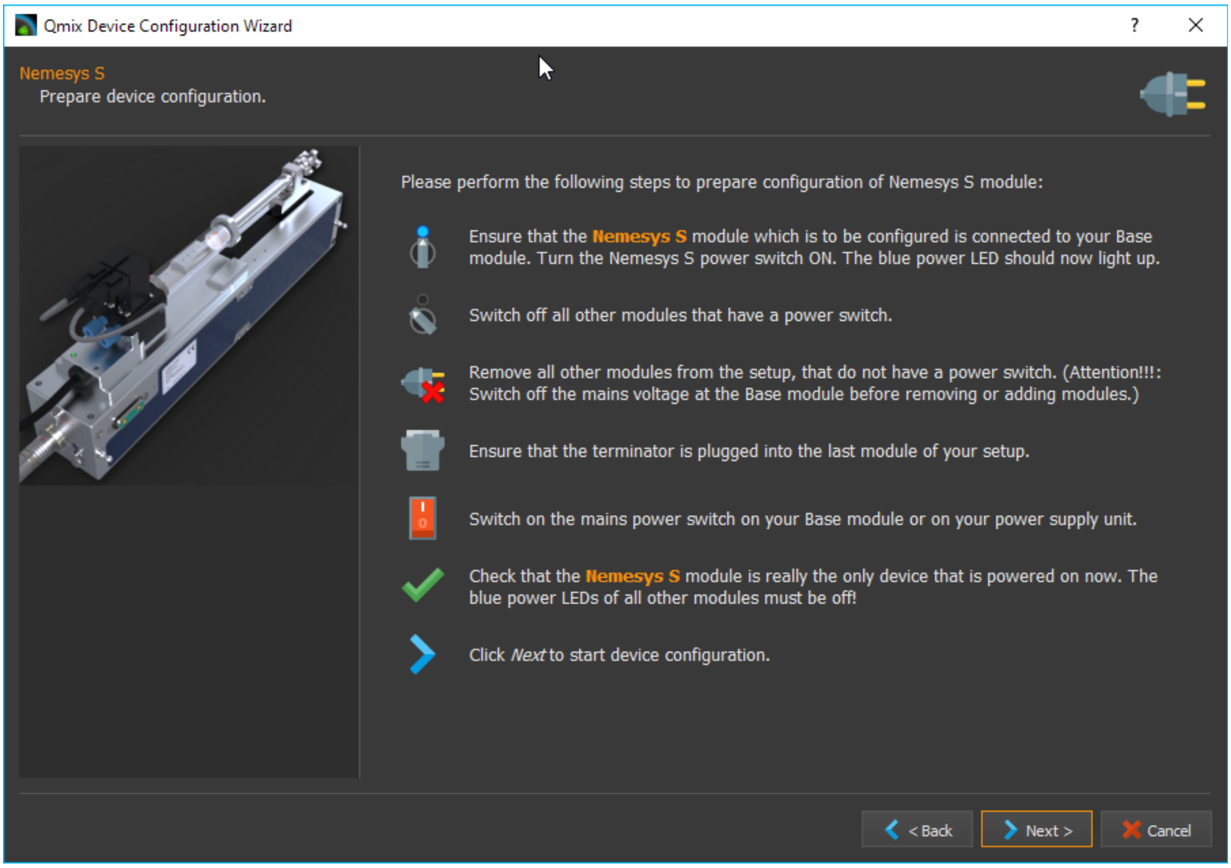


Figure 61: Prepare the device configuration process

IMPORTANT. During the configuration of a device, no other switched-on devices may be connected to the Base module, as otherwise errors may occur during the configuration or subsequent malfunctions.

- Click *Next* to start device configuration

(8) The Nemesys S pump is being configured. This may take a few seconds.

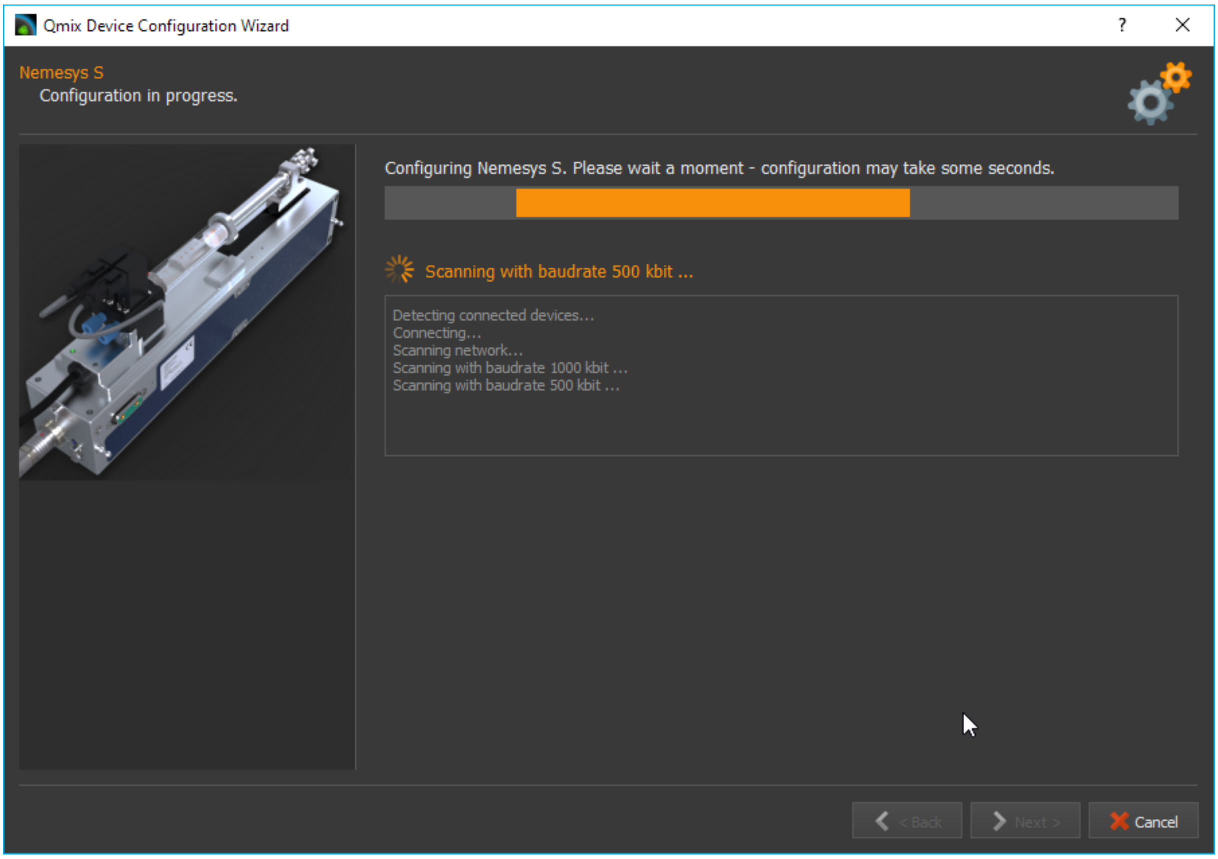


Figure 62: Configuring device

(9) The Device Configuration Wizard displays the Nemesys S dosing module configured successfully. Left-click *Next* to proceed.

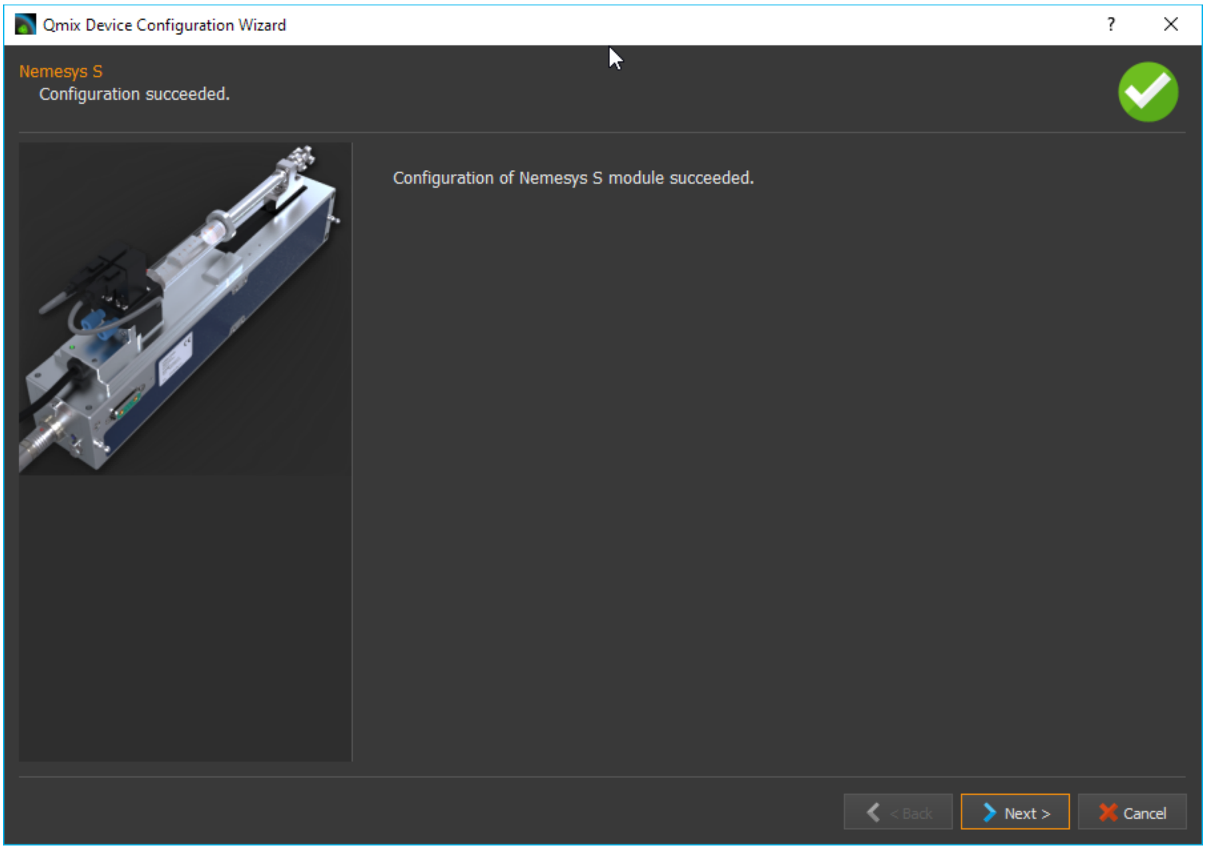


Figure 63: Device configuration successful

(10) The Device Configuration Wizard advices you to follow these instructions (see image below):

- Ensure that the Qmix Q+ module is connected to your base module
- Switch off all other devices that have a power switch – in this case switch off the Nemesys S module. Alternatively, you can remove the Nemesys S pump from the base module and connect only the Qmix Q+ module.
- Remove all other modules from the setup, that do not have a power switch
- Ensure that the terminator is plugged into the last module of your setup
- Switch on the mains power switch on your Base module or on your power supply unit.
- Check that the Qmix Q+ module is really the only device that is powered on now. The blue power LED of all other modules must be off

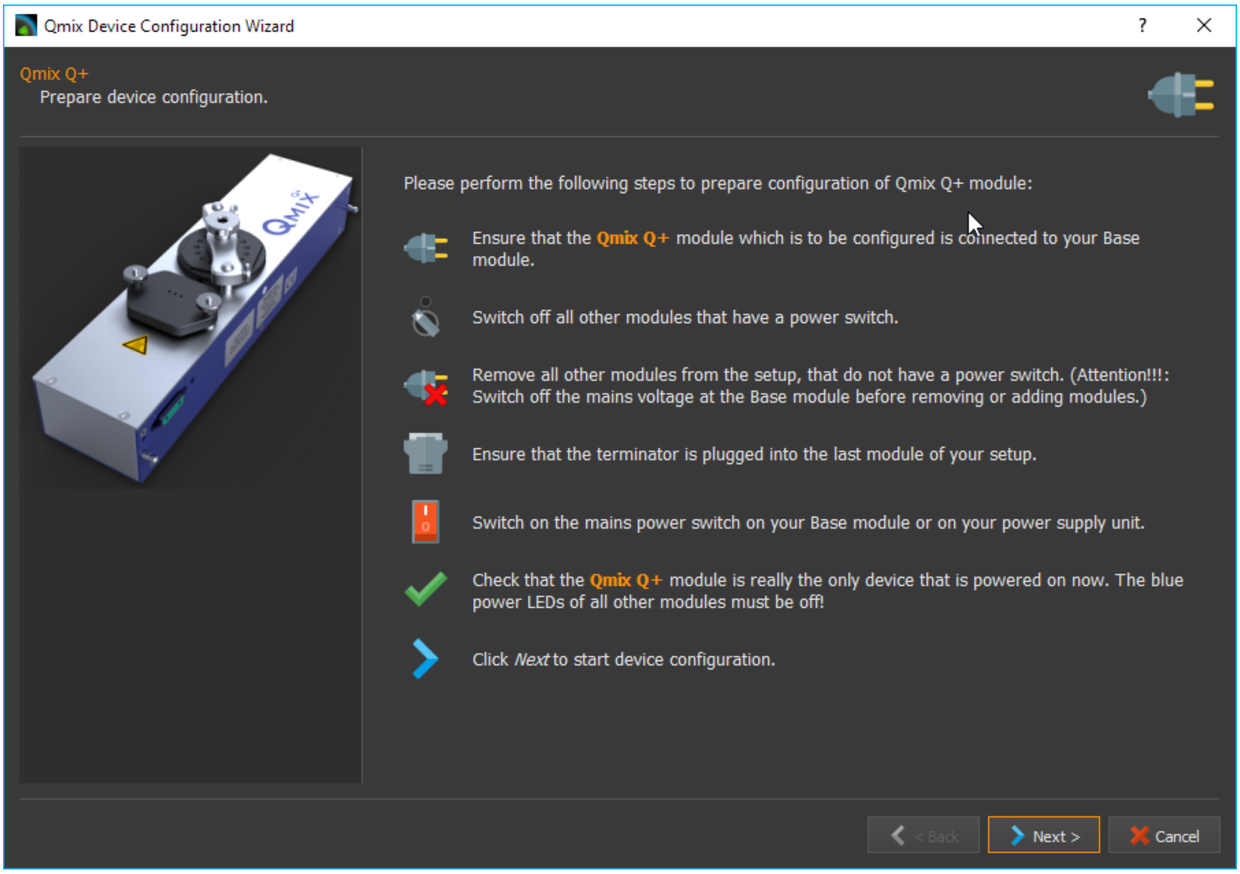


Figure 64: Unplug configured device

(11) Click *Next* to configure the QmixQ+ module.

(12) On the next page check *Import and activate new device configuration into CETONI Elements now*. Doing so CETONI Elements activates the new device configuration and restarts after finishing the Device Configuration Wizard.

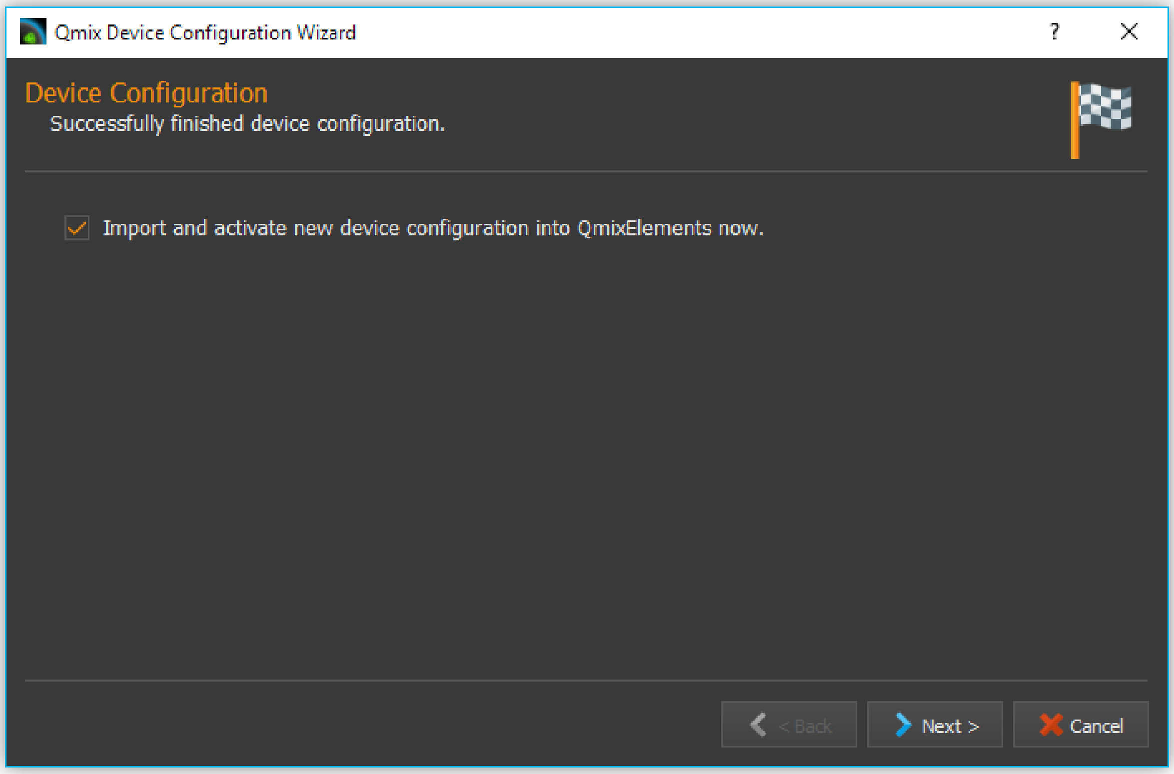


Figure 65: Activating device configuration

(13) Prepare the restart of CETONI Elements with the created device configuration:

- Switch off the Base module.
- Plug the Nemesys dosing module and the Qmix Q+ module to the base module.
- Plug the terminator.
- Switch on the base module again.
- Turn on the power switch of the Nemesys S pump
- Finish the Device Configuration Wizard by left-clicking *Finish*.

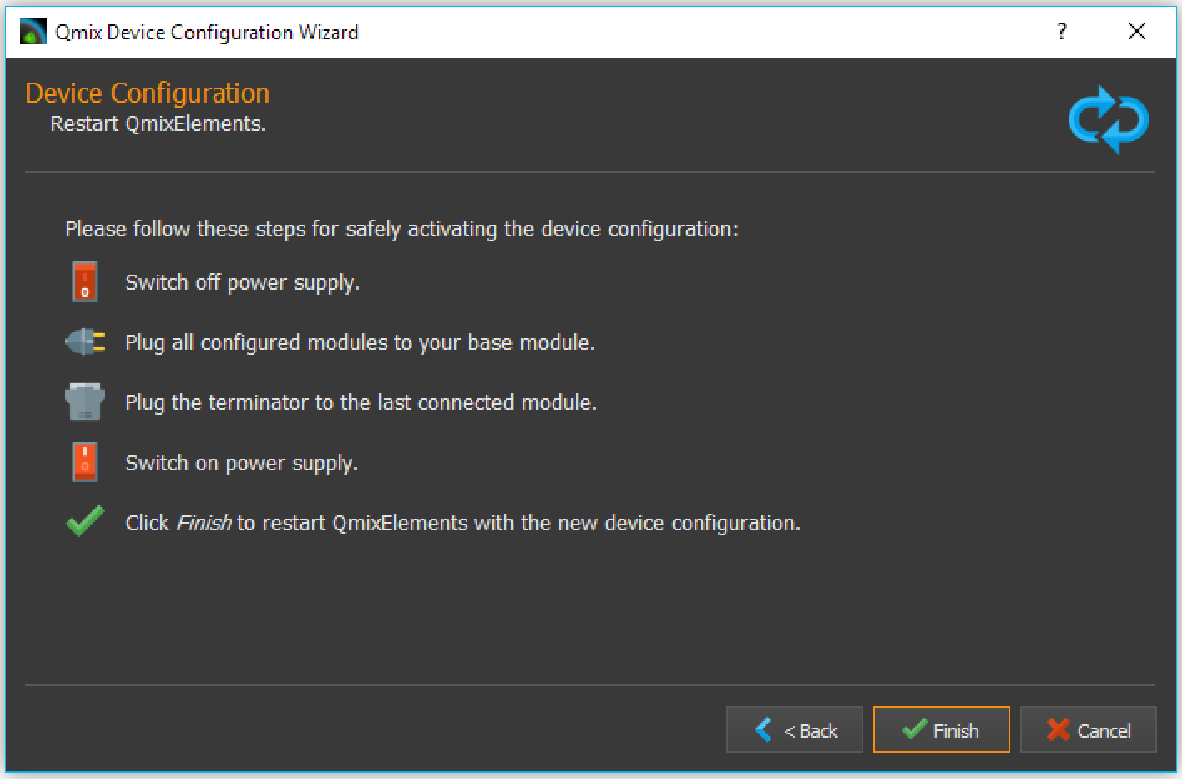


Figure 66: Finishing the Device Configuration Wizard

(14) Confirm the software restart with the new device configuration.

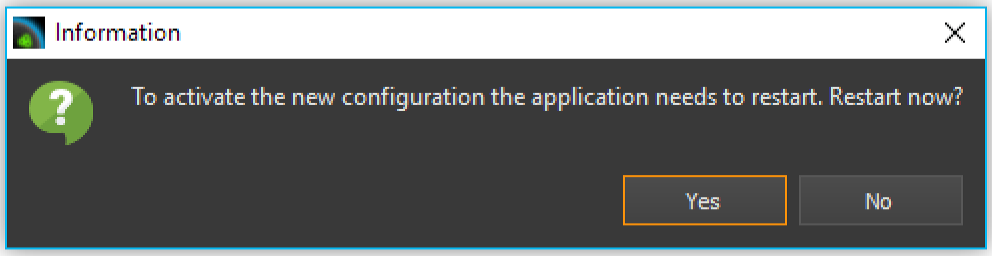


Figure 67: Confirming restart of CETONI Elements

(15) CETONI Elements now offers the functions of the neMESYS dosing module and the Qmix Q+ module.

5.9 Extending an existing device configuration

The following example shows how you can add another device to the configuration created in the previous section.

(1) Start the Device Configurator as explained in section 5.2.

(2) After start of the Device Configurator the device configuration currently loaded by the CETONI Elements software is shown. The green checks displayed on top of the devices indicate that these devices have already been configured.

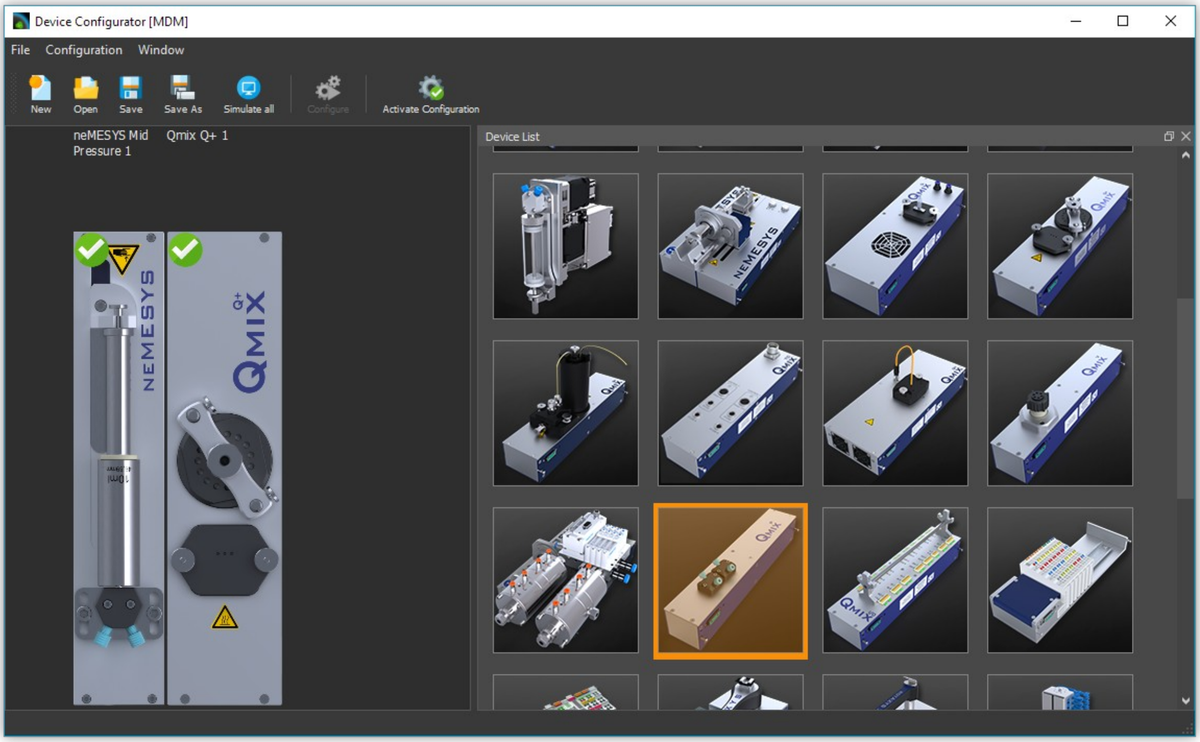


Figure 68: Displaying the currently loaded device configuration

(3) If you want to change another device configuration as the one currently loaded by CETONI Elements, you have to load the desired device configuration using the *Load* button.

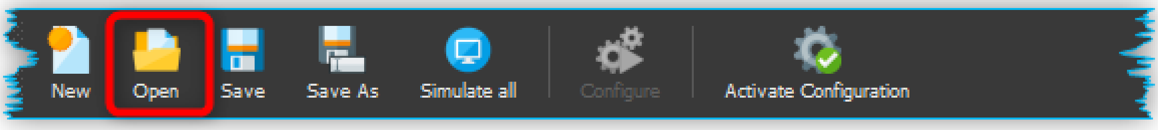


Figure 69: Loading a device configuration

(4) Add a further device via Drag & Drop as described in the previous section. The exclamation mark above the device points out that it has not been configured yet. The configuration can't be used in this state.

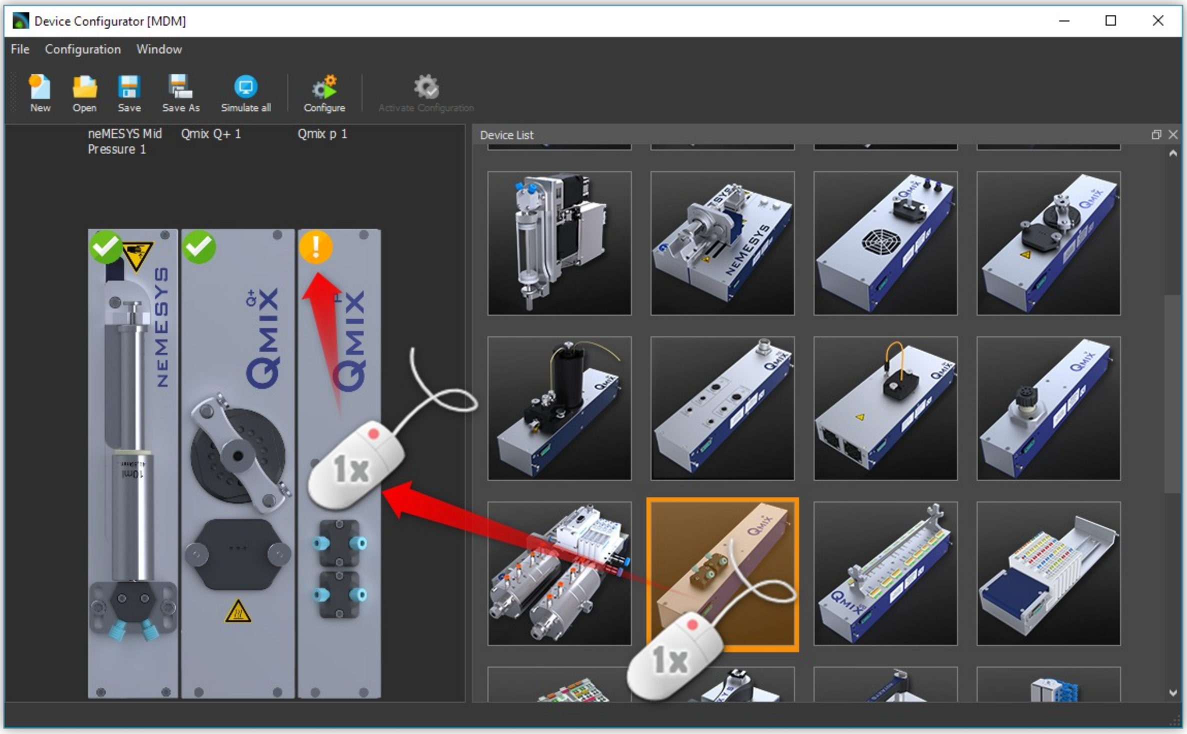


Figure 70: Adding a device to an exisiting device configuration

IMPORTANT. If a device configuration contains not configured devices, it is invalid and can't be used yet.

(5) Start the Device Configuration Wizard by clicking the *Configure* button.

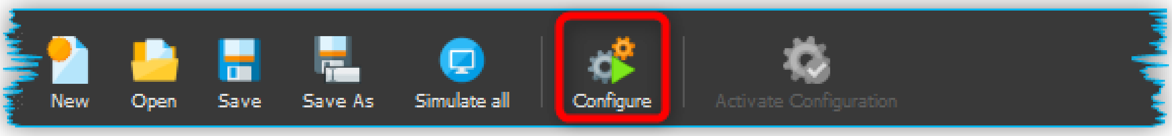


Figure 71: Starting the Device Configuration Wizard

(6) If the software has already been connected to the plugged devices, the device configuration process can't be continued. In this case the Device Configuration Wizard proposes to automatically restart the software and the Device Configurator. Click the *Yes* button if you agree.

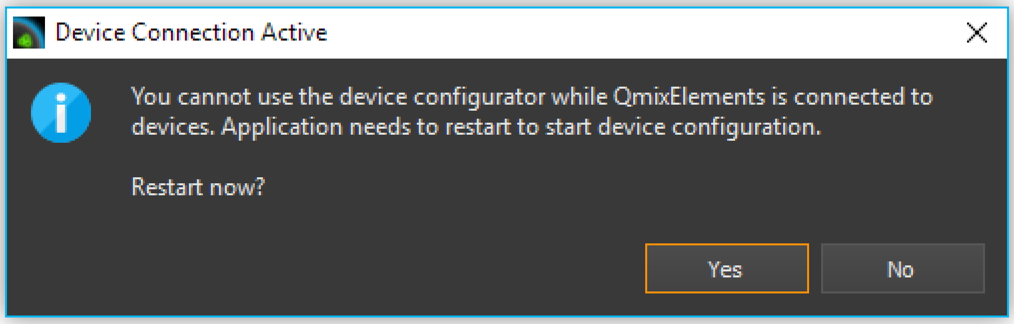


Figure 72: Restart the Device Configurator

(7) The device configuration process known from the previous section starts. But this time the wizard proposes on the second page to exclude devices from the device configuration process that have already been configured. Just check *Skip configured devices*.

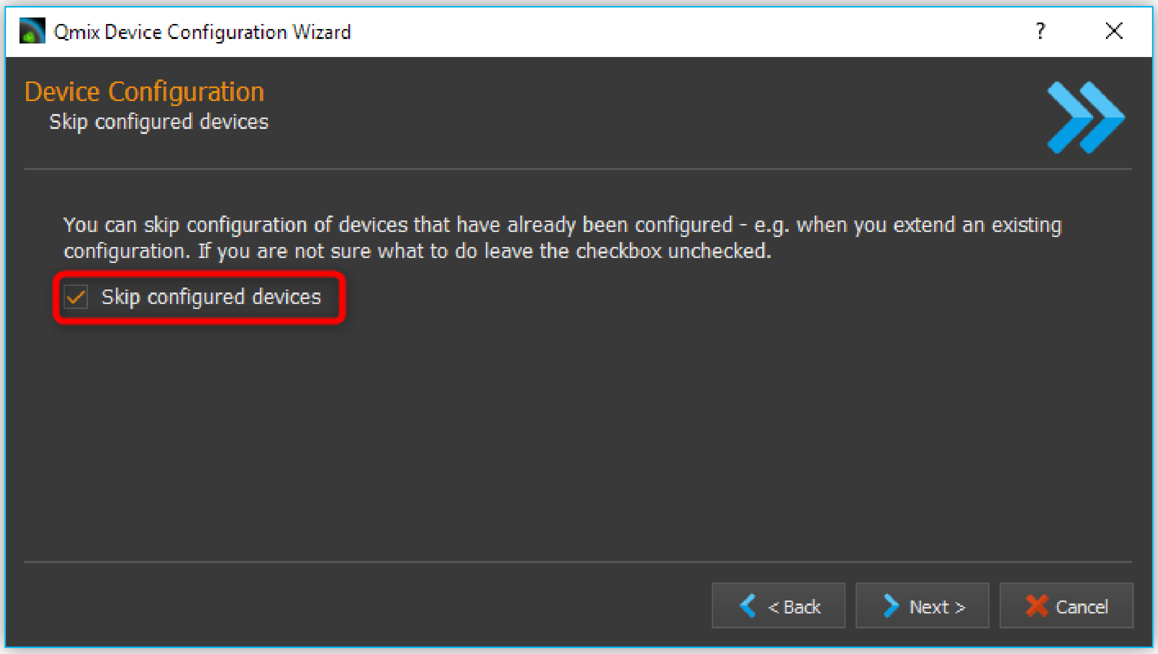


Figure 73: Skip configured devices

(8) The Device Configuration Wizard directly continues with the configuration of the newly added device (Qmix V in this example).

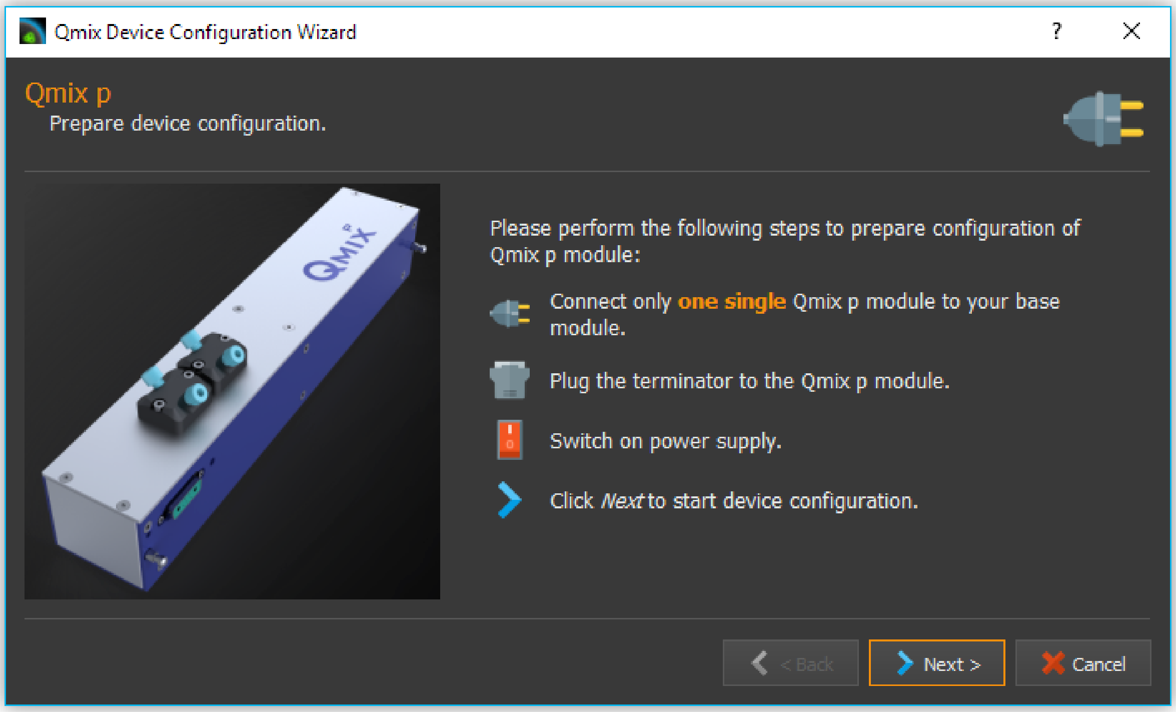


Figure 74: Configuration of the newly added device

(9) Just finish the Device Configuration Wizard the same way you would have done if you had created a device configuration from scratch. (see section 5.8)

(10) After restart the functionality of the new device is available to you.

5.10 Configuration of a single device

You can configure a single device from your configuration in the Device Configurator at any time - e. g. if you have exchanged a defective device for a new one. Simply right-click on the device to be configured and select the menu item *Configure*.

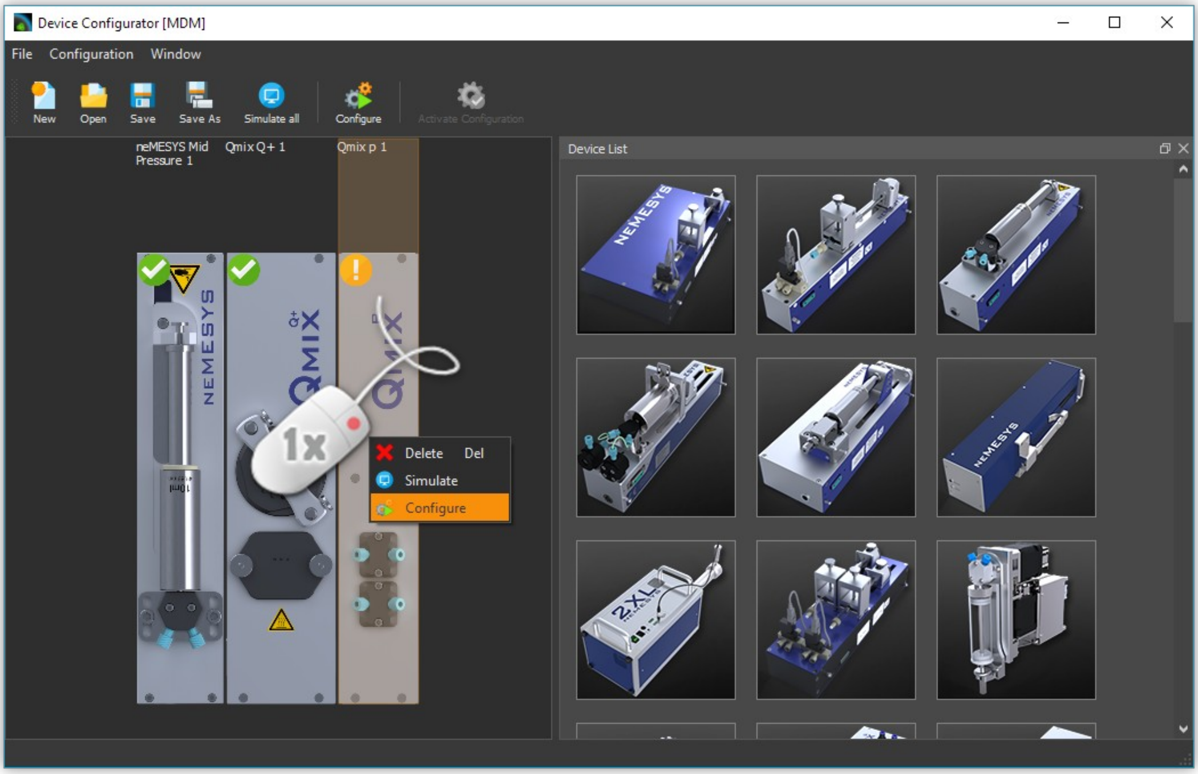


Figure 75: Configuring a single device

The configuration wizard now guides you through the configuration of the individual device.

5.11 Simulated devices

You can set single devices or a whole device configuration to simulated. This is useful if you want, as an example, create CETONI Elements scripts using devices not available at the moment. The CETONI Elements demo configuration for example completely consists of simulated devices. Proceed as follows in order to simulate a single device:

(1) In the device configuration right-click the device you want to simulate.

(2) Left-click the button *Simulate*

Figure 76: Simulating a single device

(3) The device is marked as simulated. Save the device configuration. The next time you load this device configuration in CETONI Elements the simulated device will be available to you.

Figure 77: Simulated device

You can simulate a whole device configuration by left-clicking *Simulate All* in the main toolbar.

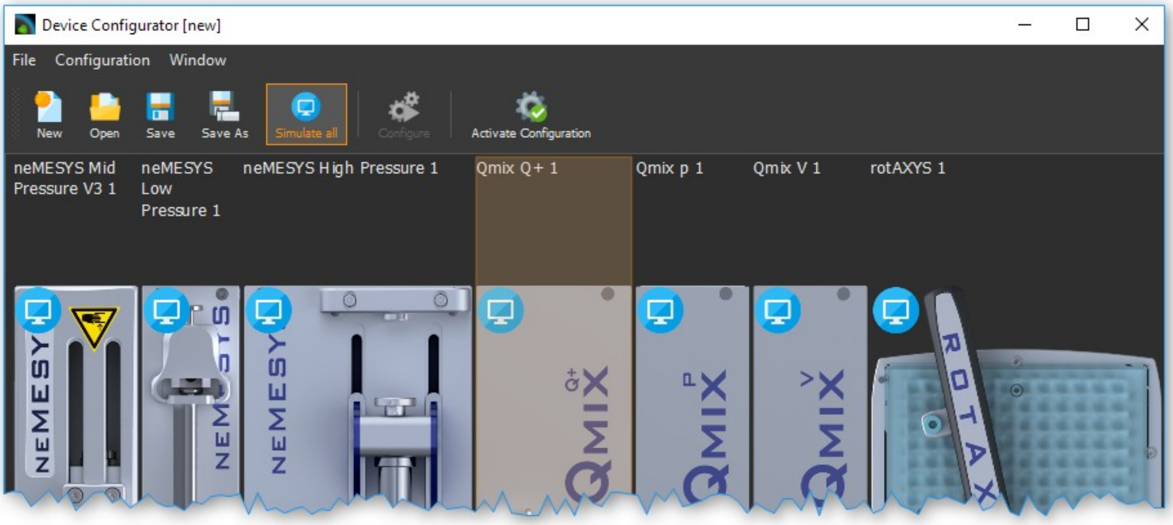


Figure 78: Simulating a whole device configuration

5.12 Optional Add-On Plugins

There are optional extensions of the software, which CETONI has developed, possibly on your behalf, according to customer specifications. However, certain plugins may not be required for a certain device configuration, or may not be suitable for this purpose. When creating and editing a device configuration, you can freely configure which optional plugins should be loaded with your device configuration.

HINT. The CETONI Elements software can be extended with customer-specific plugins. Please contact us if you need a specific adaptation of the software functionality.

If you only have a standard version of CETONI Elements installed, no optional add-on plugins will be available. Accordingly, the function for configuring optional add-on plugins is disabled (see figure below).

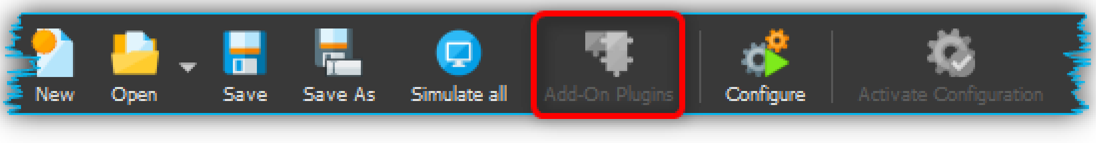


Abbildung 1: Function for selecting optional add-on plugins for standard CETONI Elements installation

If you have installed a CETONI Elements add-on that contains optional plugins, this function is available (see figure below).

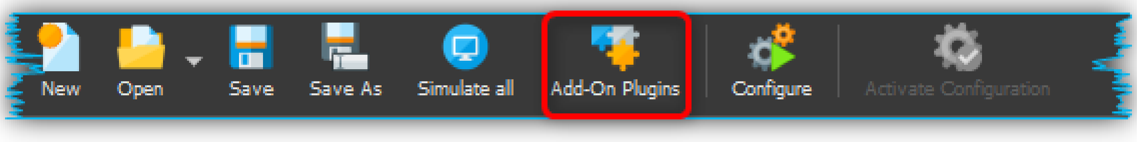


Abbildung 2: Function for selecting optional add-on plugins, if these are availble

IMPORTANT. Not every CETONI Elements add-on contains optional plugins. Many add-ons (e.g. Spectroscopy Add-On) contain only mandatory plugins that must be loaded to provide device functionality. Mandatory plugins cannot be selected using this function. These are always loaded automatically if the respective device (e.g. Qmix λ) has been configured.

If you click with the left mouse button on the *Add-On Plugins* function, the Add-On Plugins Selection Dialog appears. In the lower half of the dialog, there is a table that displays all available optional add-on plugins (see figure below).

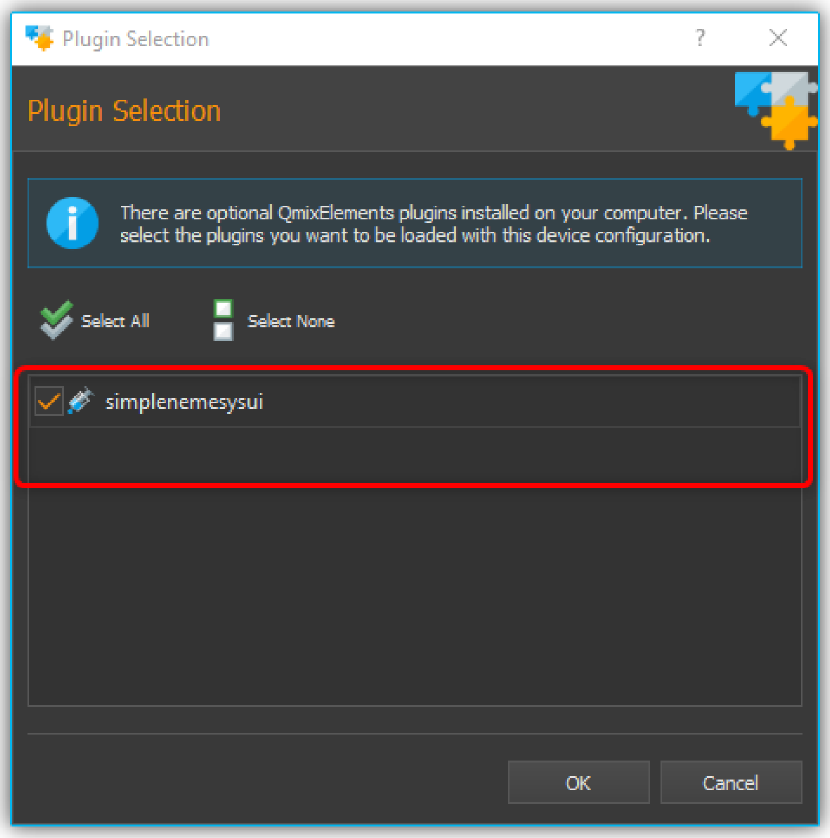


Abbildung 3: Add-on plugins dialog with available plugins

You can display information about the add-On plugin by moving the mouse over the respective table entry (see figure below).

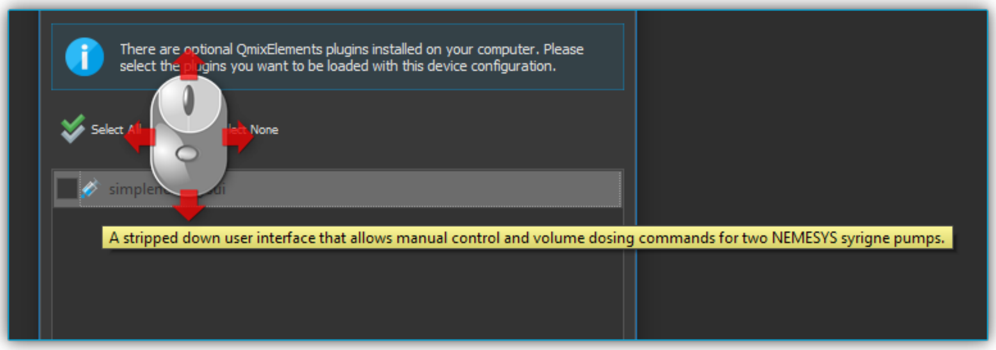


Abbildung 4: Showing detailed information about an add-on plugin

To add an optional add-on plugin to your device configuration, check the box next to the plugin 1 and then click Ok 2. Finally, you must save the configuration 3 to permanently apply the changes (see figure below).

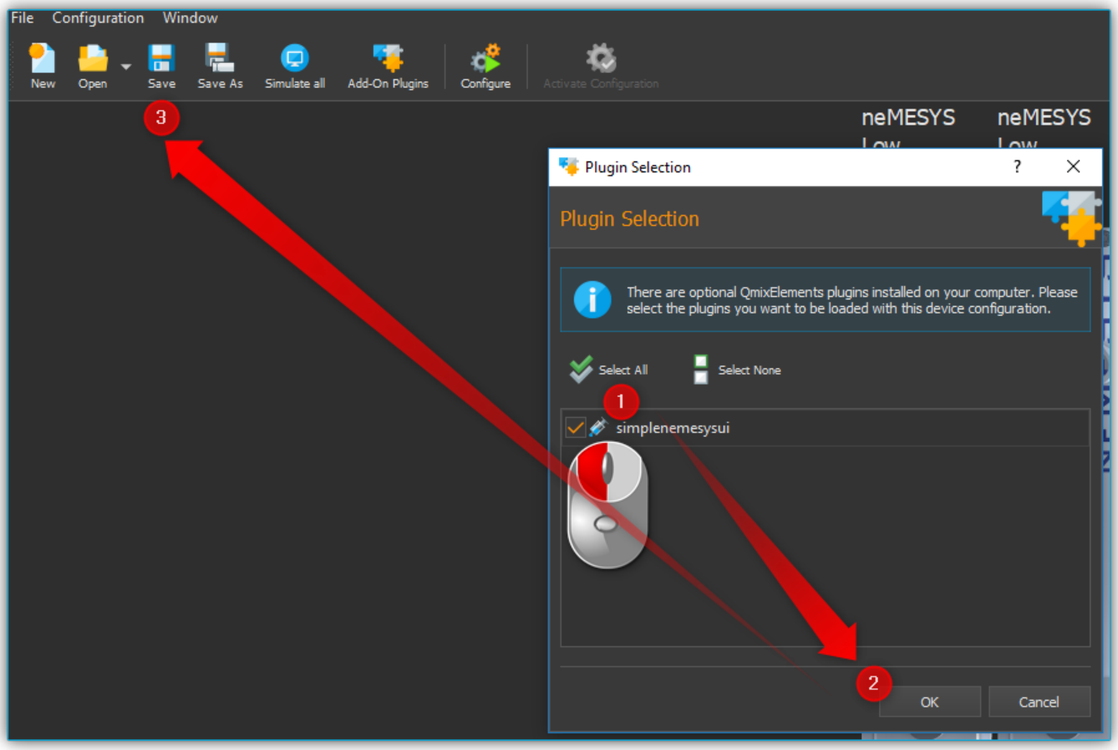


Abbildung 5: Adding an add-on plugin to a device configuration

6 Script System

6.1 Introduction

The software provides a powerful scripting system to set up automated process sequences.

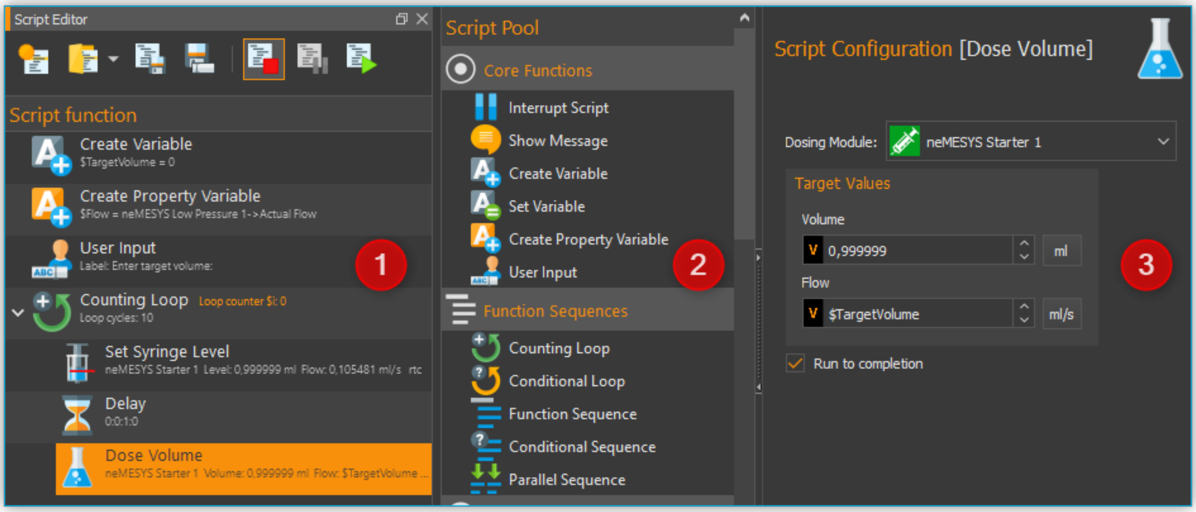


Figure 79: Script system overview

The script system work bench consists of the following three main parts:

1 *Script Editor* – shows the script programmed by the user as a function tree. It also features buttons for controlling script files and their execution.

2 *Script Pool* – contains all available script functions ordered in device categories.

3 *Script Configuration* – is used to configure the parameters of individual script functions.

6.2 Script Pool

Activate the *Scripting* button in the side bar to show the Script Pool.

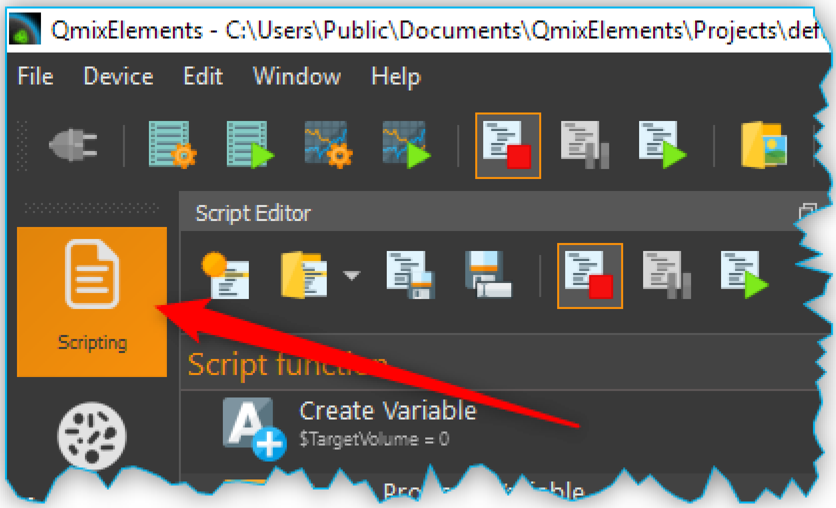


Figure 80: Show Script Pool

In addition to the *Script Pool*, the *Script Configuration* area is also shown. The *Script Pool* contains all script functions that are available for programming scripts. The script functions are grouped into categories. In addition to a set of core functions, each device and each plug-in registers its own specific script functions in a separate category.

The user can open or close the categories in the *Script Pool* at any time. To open or close the function list of a category, simply double-click on the category name (figure below).

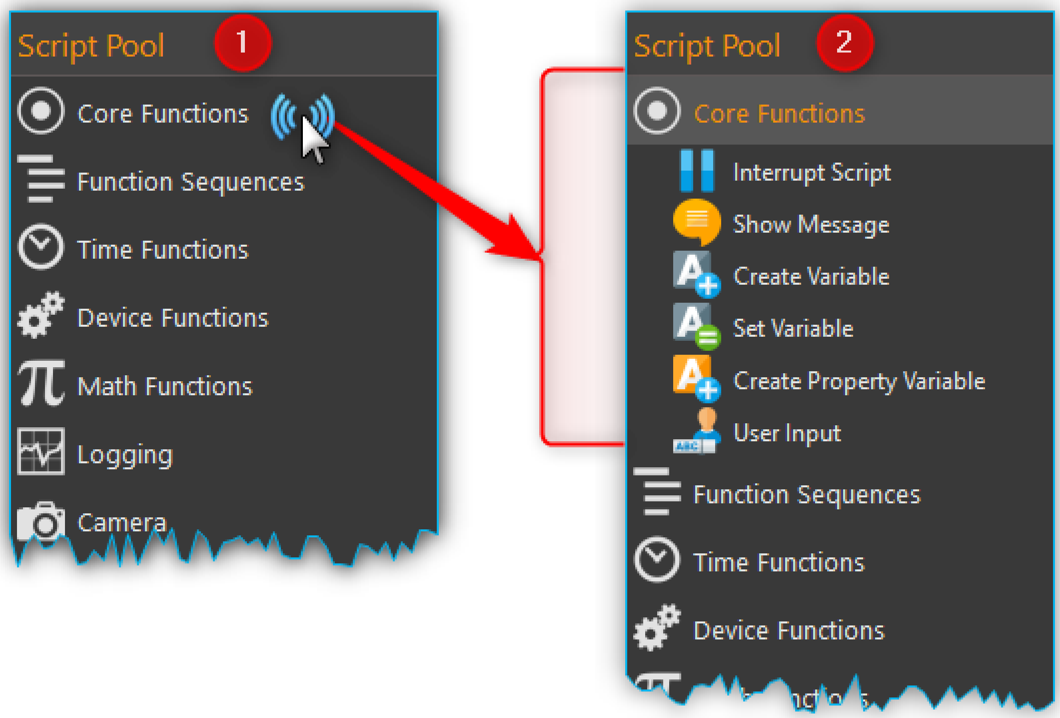


Figure 81: Script Pool – Expanded view (2) of individual category functions.

6.3 Script Editor

6.3.1 Introduction

The *Script Editor* is used for the graphic programming of scripts. The following items are numbered in the figure below:

1 *Toolbar* – is for loading, saving, and controlling the sequence of scripts.

2 *Function tree* – shows a treelike program structure.

3 The currently executed function is highlighted in green.

4 When you select a function by clicking, it is highlighted blue.

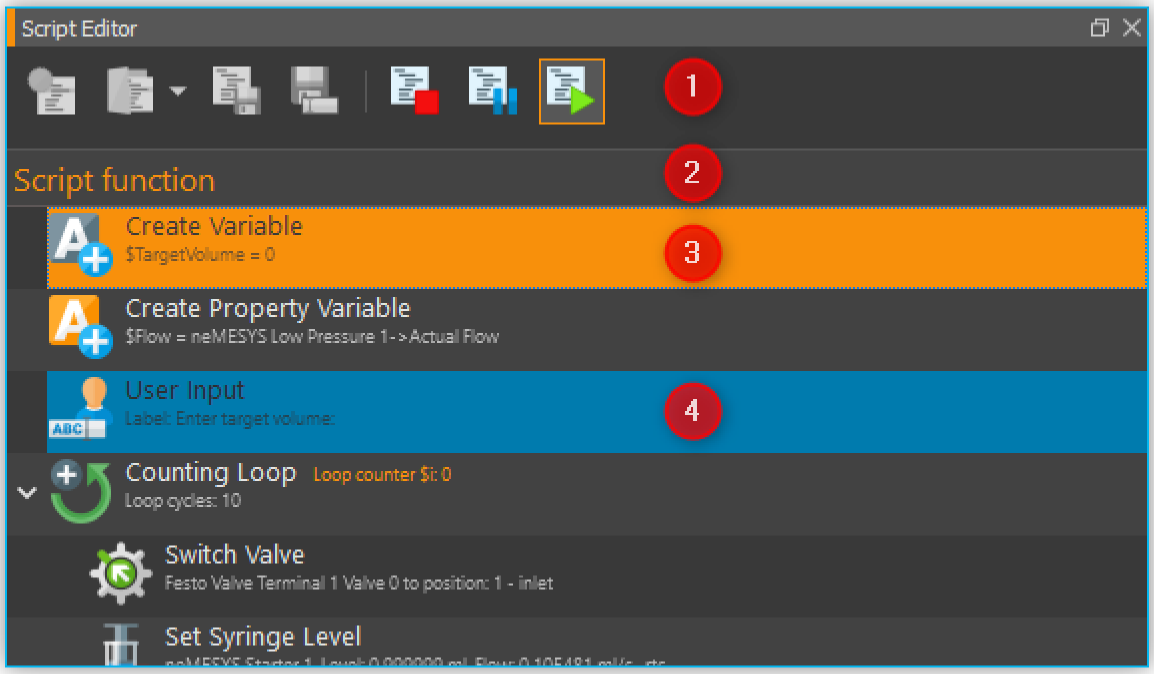


Figure 82: Script editor panel

Each single function is displayed in the function tree in a separate line. In this line all the important function parameters are visible for you (see figure below):

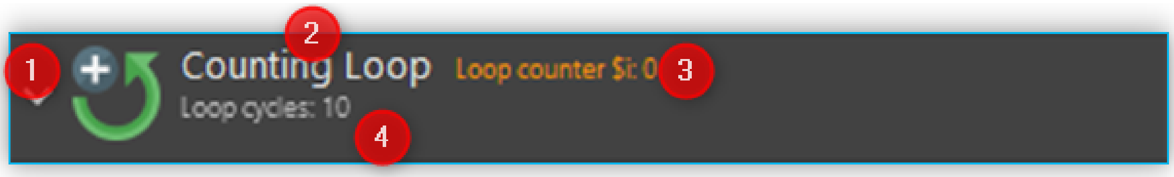


Figure 83: Single function in the Script Editor

On the left side you will find the graphic icon of the function 1. Immediate right of the icon at the top 2 you will find the function name. Status information can be found to the right of the function name 3. For many functions, these status information are visible only during the execution and are subject to change. Directly below the function name is a summary of all important function parameters 4, that you have configured in the configuration area.

The *Script Editor* is a movable and dockable window: you may move and dock the *Editor* to another position within the main software window. To do this, drag-&-drop the window via the title bar to its new location using the computer mouse. If the *Editor* window is not visible, you may first have to activate it via *Window* → *Script Editor* in the main menu (figure below).

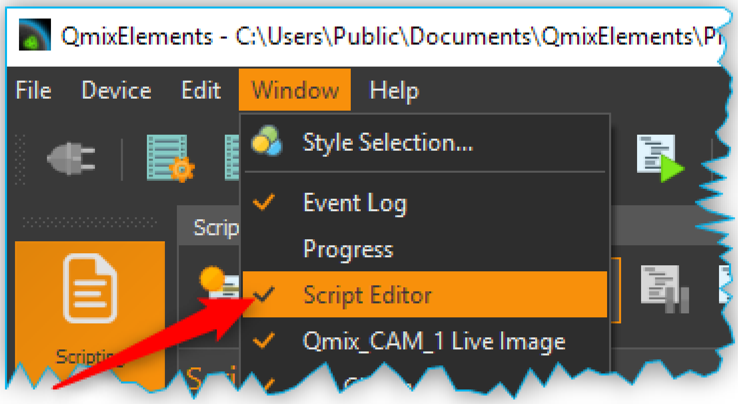


Figure 84: Activating the Script Editor

You can change the scaling in order to increase the clarity or adjust the display of the editor to suit your requirements. To do this, right-click in the editor to open the context menu and select the size of the display in the submenu *Set Item Scaling:*

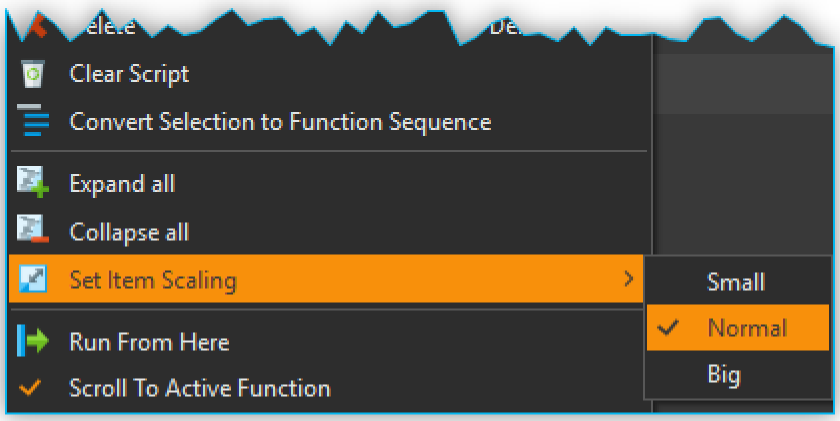


Figure 85: Change script editor scaling

The following sizes are available:

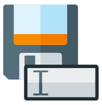
- **SMALL** – very compact display for maximum clarity in complex function processes, the function parameters are no longer displayed.
- **NORMAL** – normal size
- **BIG** – the icons and the function names are enlarged for optimal readability

6.3.2 Toolbar

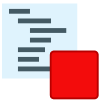
Generates a new, empty script.

Loads an existing script file into the *Script Editor*.

Saves the currently active script.



Saves the currently active script into a new script file.

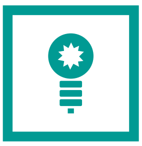


Stops the execution of the current script. By clicking the start button, the complete program will be restarted from the beginning.

Request Script Stop - clicking this button sets the *$StopRequested* global script variable to true. The script can query this variable cyclically, e.g. in the main loop of the script, to react to a stop request and terminate the script. Thus the script has the possibility to bring devices into a safe state (e.g. to stop pumps, to switch valves) before the script execution is terminated.

Pauses the execution of the current script. By clicking the start button, the execution will resume from its current position.

Starts the execution of a script or resumes a script after an interruption.



HINT. You can also load script files easily via drag & drop. Simply drag a script file from your file system over the script editor and drop it there.

6.3.3 Context Menu

If you click with the right mouse button on any function within the *Script Editor*, a context menu appears to quickly access additional actions (figure below).

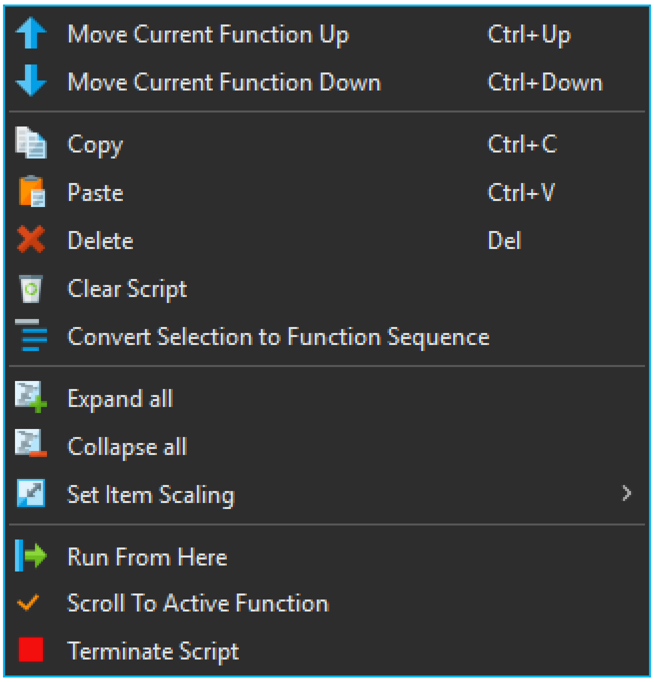
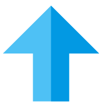
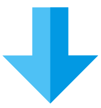


Figure 86: Script Editor context menu

The following functions are available from the context menu:

 Moves the selected function one position up

 Moves the selected function one position down

 Copies the selected functions to the clipboard

 Inserts the functions from the clipboard before the selected function

 Deletes the selected functions

Deletes all functions in the script

Replaces the selected functions with a function sequence and inserts the functions into this sequence. This allows you to quickly group functions together to structure your script, make it clearer and improve readibilty.

Expands all functions in the script

Collapses all functions in the script so that only the topmost function level is visible

Changes the scaling of the display of the script functions. This increases the clarity of complex scripts.

Starts the script execution at the selected function. If variables are previously initialized in the script, it is possible that they are not yet initialized in this case.

Terminates script execution immediately. All devices remain in their current state, are not shut down or stopped, and do not transition to a safe state.

HINT. To start script execution at a specific function from within a script, click the right mouse button on the function of choice and select **Run From Here** from the context menu.

6.4 Script Configuration Panel

6.4.1 Overview

The configuration panel contains all controls for configuring the script function that is currently selected in the *Script Editor*.

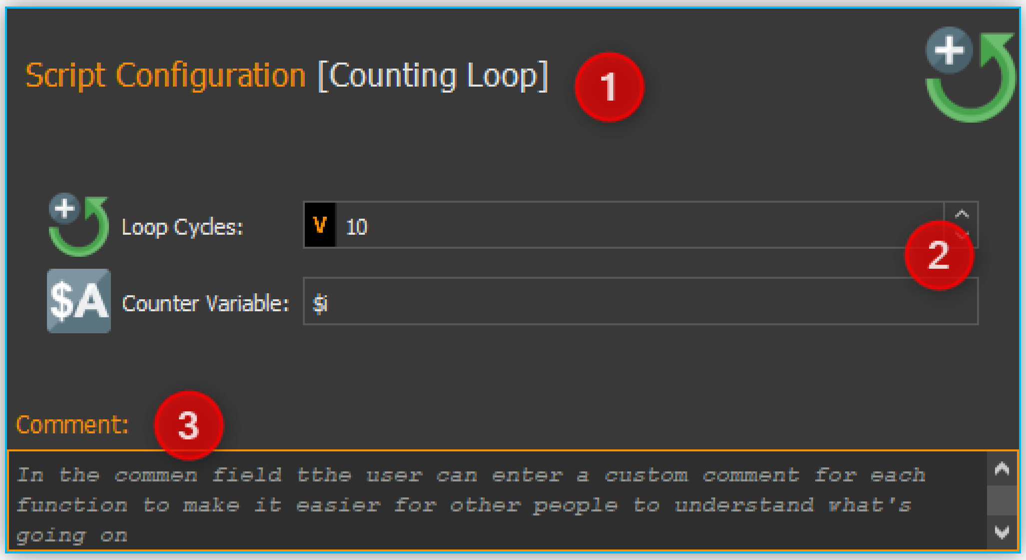


Figure 87: Script Configuration Area

The configuration area consists of:

- 1 header with the name of the currently selected function
- 2 input- and control elements of the function
- 3 comment box to enter any commentary

The input- and control elements 2 are different for each script function. But all functions provide some common controls like the function caption 1 in the header and a comment box 3 at the bottom of the configuration area.

6.4.2 Changing Function Caption

In the header of the configuration area you can change the caption of the function. It allows you to use “talking” function names that make it a lot easier for you or third parties to read and understand your scripts later.

To change the function name, you can either click with the left mouse button on the function name in the header or you can click on the name with the right mouse button and select the context menu item, *Edit Function Caption* (see figure below).

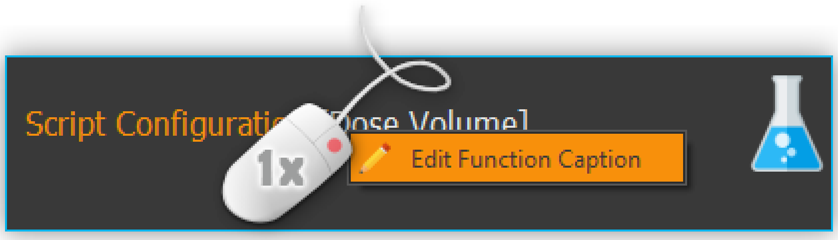


Figure 88: Showing Caption input dialog

In the input dialog that now appears, you can enter a new name for the function.

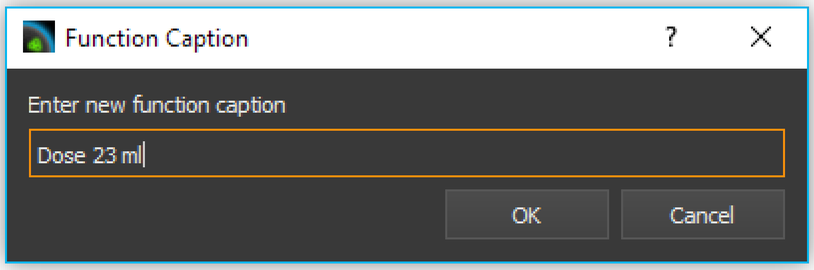
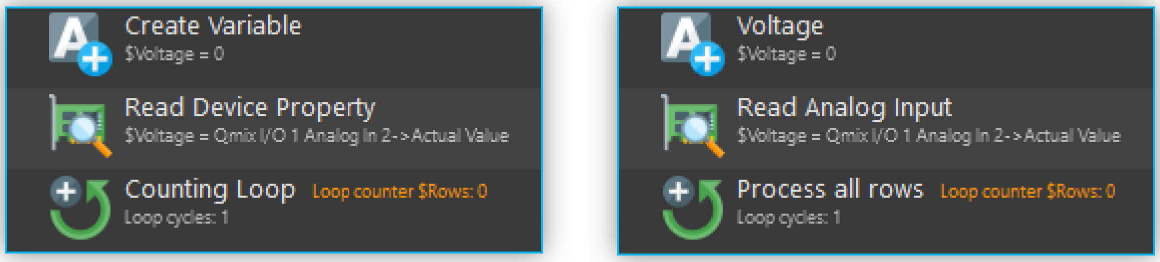


Figure 89: Function Caption input dialog

The following example shows a short program with the default function names on the left side and the same program with own function names on the right side:

Default function names
Application-specific names



HINT. Enhance the readability, understandability and maintainability of your scripts through the use of speaking, application-specific function names.

6.4.3 Enter Comment

In the comment field you can enter a comment that will allow other users to understand your scripts better and to follow the flow of execution easier.

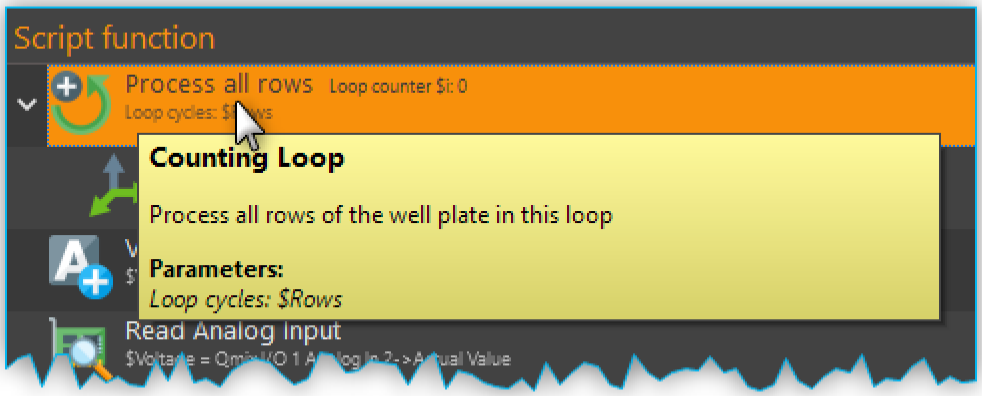


Figure 90: Script function comment

When you move the mouse pointer over a function in the *Script Editor* the comment of this function will be shown in a message box (see figure above). So you can read the comment of a function without having to open the configuration area of that function.

6.5 Programming

6.5.1 Adding Functions

Functions are activated via drag-&-drop from the *Script Pool* to the *Script Editor*. To do this, proceed as follows:

(1) In the Script Pool, left-click on the function that you want to insert 1 and hold down the mouse button.

(2) Move the pointer to the desired position within the *Script Editor* window.

(3) As soon as you release the mouse button 2, the selected function will be inserted into the Script Editor.

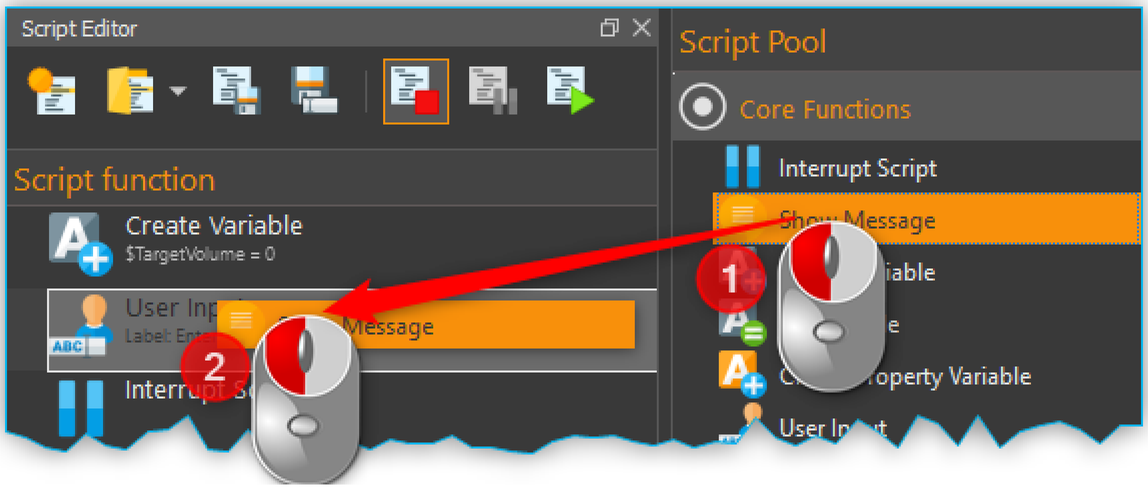


Figure 91: Inserting a function into a script via drag-&-drop

The function is inserted according to where the mouse pointer is positioned when you release the mouse button. The following scenarios are possible (figure below):

1 If you release the mouse button atop an existing function, the new function will be inserted immediately before that existing function.

2 If the mouse button is released atop a function sequence (for example, a loop), the new function will be inserted at the end of that sequence.

3 If the mouse button is released on an empty area at the end of the function tree, the function is added at the end.

6.5.2 Selecting Functions

To move, copy or delete functions, you must first select them. You can either select a single function by clicking on it, or you can select a continuous sequence of functions at the same hierarchy level.

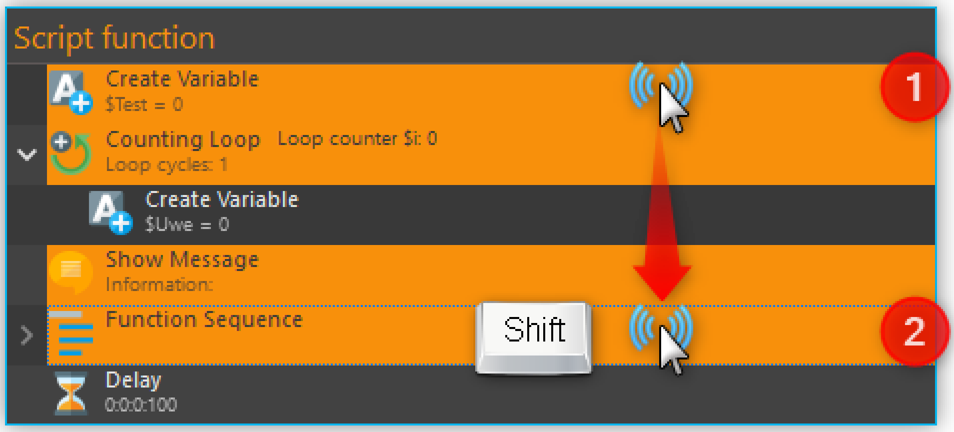


Figure 92: Script Editor contiguous selection

When you select an item in the usual way with the left mouse button, the selection is cleared and the new item is selected 1. However, if you press the *Shift* key while clicking on an item 2, all items between the current item and the clicked item are selected or unselected, depending on the state of the clicked item.

6.5.3 Moving Functions

Analogous to inserting a new function from the *Script Pool*, you can move the functions to new positions within the function tree via drag-&-drop. Again, the same insertion rules apply as above.

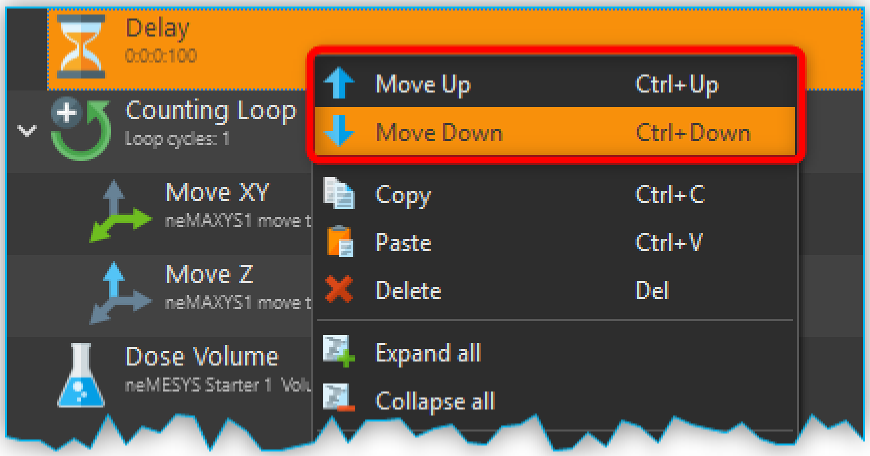


Figure 93: Moving functions within a function tree

There is alternative way for moving functions up and down the list: First, make a right mouse click on the function that is to be moved. This will open a context menu from which you may then select *Move up* or *Move down*, respectively (figure above). Alternatively, you can use the keyboard with the *Ctrl+Cursor up* or *Ctrl+Cursor down* shortcuts.

This latter method may only be used to move functions up or down within the current sequence. If you want to move a function to a completely different position within the function tree, this can only be done via drag-&-drop.

IMPORTANT. *Move up* and *Move down* only moves the current function. Even if several functions are selected, only the current function is moved. If you want to move the entire selection, you can do this by dragging and dropping.

6.5.4 Deleting Functions

There are two ways to delete functions:

1. Select the functions you want to delete and then click the context menu entry *Delete*.

2. Select the functions you want to delete and then press the *Delete* key of your keyboard.

6.5.5 Copying Functions

Similarly, functions can be copied either by using the context menu via the mouse or using key combinations via the keyboard. If you work with the context menu, simply select *Copy* and then *Paste* from the menu (figure below); with the keyboard, use the *Ctrl+C* shortcut to copy and *Ctrl+V* to paste.

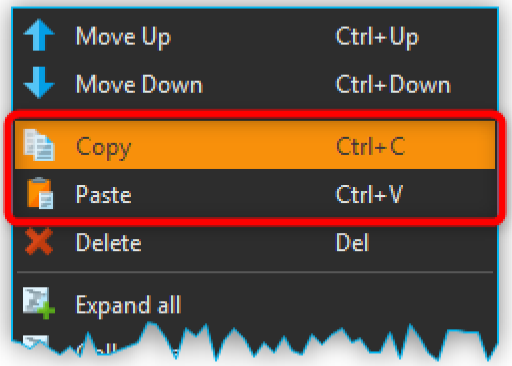


Figure 94: Copying a function

This is how you copy a function to a new position:

1. Select the functions you want to move.

2. Copy the functions via *Copy* of the context menu or via the keyboard and *Ctrl+C*.

3. Select the function before which you want to insert the copied function by left-clicking it.

4. Paste the copied functions via *Paste* of the context menu or via the keyboard and *Ctrl+V*.

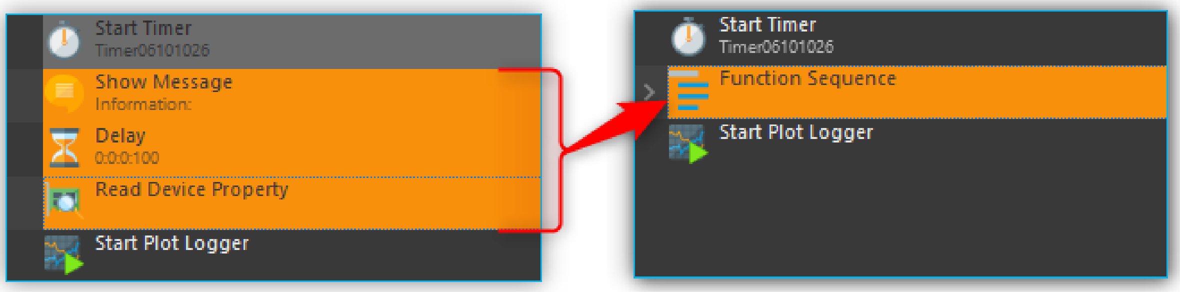
To insert the same functions at multiple points of the function tree simply repeat steps 3 and 4 (above).

6.5.6 Grouping Functions

To improve the clarity and readability of your script, you can quickly and easily group sequences of functions into function sequences. To do this, simply select a contiguous set of functions, and then click *Convert Selection To Function Sequence* in the context menu.

Figure 95: Grouping Functions

The selected functions are now replaced by a function sequence containing the selected functions.



6.5.7 Editing Function Parameters

As soon as you select a function from the function tree, the operating elements appear in the *Script Configuration* area that allow you to configure the selected function. Edit the function parameters as required.

6.5.8 Showing Function Tooltip

If you move the mouse cursor over a function, a tooltip window is displayed for this function after a short time (see figure below). In this window, you will get all information about this function at a glance: function name ❶, user comment or function description ❷ and function parameters ❸.

Figure 96: Showing script function tooltip

6.6 Variables

6.6.1 Introduction

The script system of supports the use of variables. Within script programs, variables serve as containers for calculated and device values. Their values are generated during program execution from, for example, loop counts or sensor data. Each variable is defined by a name. Script functions that support the use of variables may then use the values stored in these containers, *e.g.*, to trigger value-dependent events.

6.6.2 Setting Variables

Before variables may be used by a script, such variables need to be defined. There are two possibilities to setup variables:

1. Explicit: Variables are defined explicitly by the user, *e.g.*, via the function *Create Variable*.
2. Implicit: Implicit variables are created via functions that offer variables by default, such as the counter of the *Counting Loop* function.

6.6.3 Naming Variables

There are some important requirements to keep in mind when naming variables: Every variable is called within a program script via an essentially freely definable name. This name serves to unequivocally identify that variable; different names signify different variables. The $-prefix clearly identifies a name and its use as a variable. The scripting system is case sensitive: *$Var* is different variable than *$var*.

Additionally, the following rules apply when naming variables:

- Variable names have to start with the Dollar symbol ($) and must not themselves contain a $-symbol.
- Only alphanumeric characters are allowed (a-Z, 0-9).
- Special characters (such as, $, &, /, -, ...) are not allowed.
- Variable names must not start with a number.

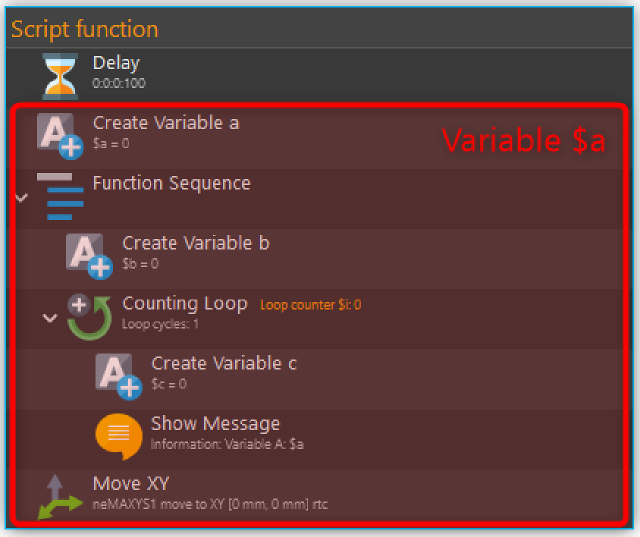
HINT. You can display the contents of variables using the Show Message function, e.g. to check the results of calculations.

6.6.4 Visibility Range of Variables (Scope)

The visibility range (scope) of a variable is that part of the program within which that variable is visible, *i.e.*, available. Qmix scripts are trees with an essentially unlimited number of branches and levels; a variable, *i.e.*, the value it returns, is only visible, *i.e.*, available to be used, at that level at which it has been defined plus all its sub-levels.

The following example illustrates the scope of variables:

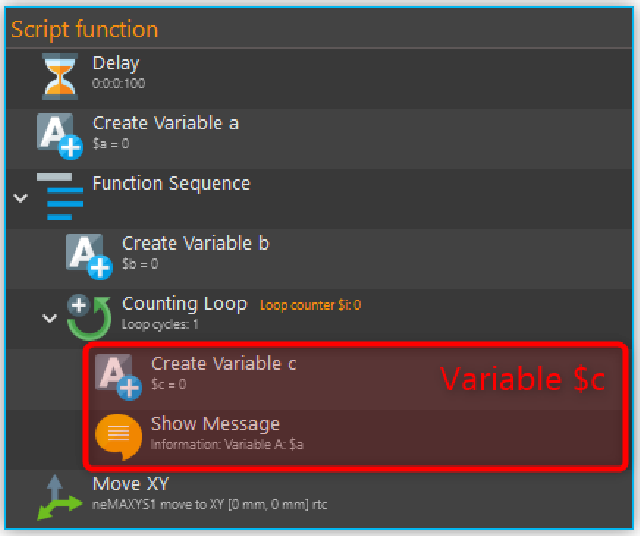
In the following example program the variable `$a` is visible in the red marked area - i. e. usable by script functions (figure below):

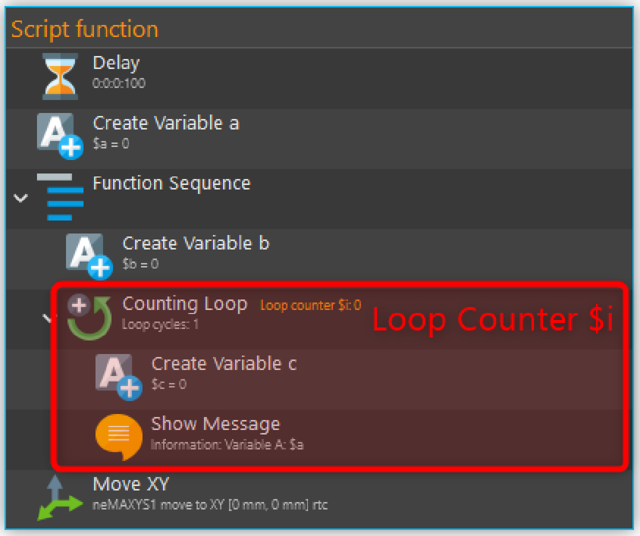


Variable `$b`, on the other hand, is only visible within a specific function sequence (figure below).

The counter variable `$i` of the counting loop is only visible in those functions that are in the counting loop:

The scope of variable `$c`, which has been setup within the counting loop, is only available within that counting loop, too, as no other sublevel has been added at this point:





It is important to note that, if two variables have the same name, the variable that has been defined later (i.e., at a lower level) will overwrite the variable defined earlier (i.e., at a higher level). In the example above, if `$c` would have been named again as `$b`, the later-defined value (i.e., 2) would replace the earlier-defined value (i.e., 1).

IMPORTANT. If a lower-level variable has the same name as a higher-level variable, the lower-level variable will supersede the higher-level variable. That is, functions at the lower level cannot access the value of the higher-level variable of the same name and will use „newer" value instead.

6.6.5 Using Variables

Variables can be used with all functions that support them. Calling a variable to, *e.g.*, set or calculate a value, requires the use of the dollar symbol ($) as a prefix: To use (call write to) the variable *a*, the required syntax is: *$a*.

IMPORTANT. Variables get assigned a valid value only after they have been assigned a value via being run through a relevant function (e.g., Create Variable). If you are using the action **Run From Here** to start a script, variables may not have been assigned a valid value yet if the respective assignment function follows later in the sequence or has been skipped.

Functions that support the use of variables have the relevant input boxes highlighted by a yellow "V" (see figure below). Here you can insert the name of a variable instead of a numeric value that is to be used subsequently within the relevant section of the program script.

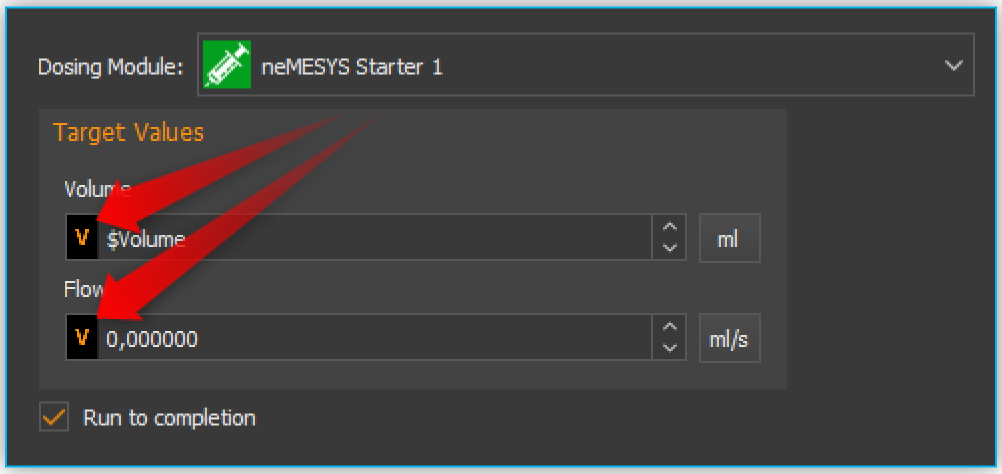


Figure 97: A yellow „V" signifies a function that supports variables

HINT. Nearly all input fields that support variables allow for direct access of device process data via device property identifiers. (see section 6.7)

6.6.6 Auto-Completion of Variable Names

Input boxes that support the use of variables, feature auto-completion to aid the selection and input of valid variable names: Upon inserting the $-symbol, a list will appear that contains all variable names defined so far (see figure above).

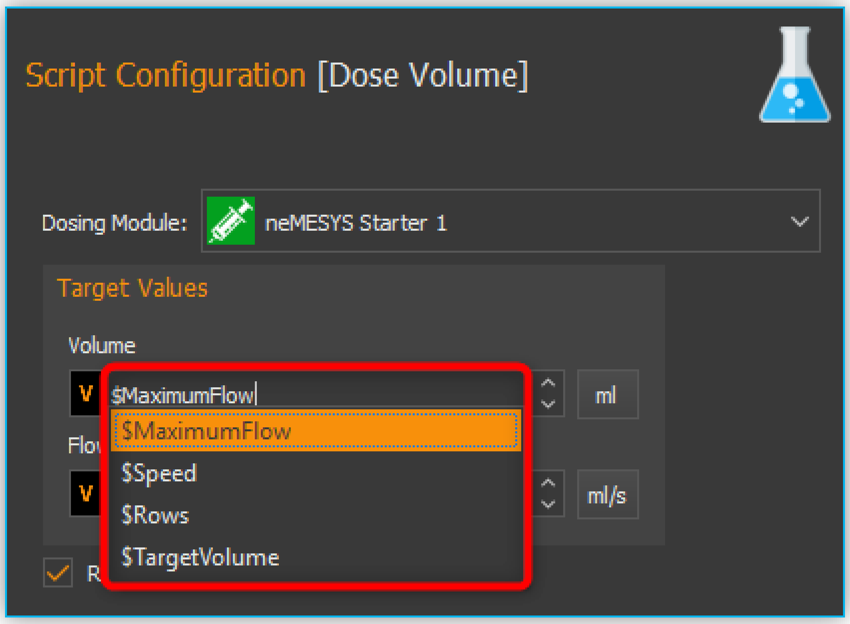


Figure 98: Auto-completion of variable names

Every additional character that you enter will cause a filtering of that list according to the character sequence inserted thus far. You may use either the *up* or *down buttons* of your keyboard or the mouse to select a name from that pre-filtered list. Accept the selection by pressing the *Enter* key.

6.7 Device Property Identifiers

Nearly all input fields that support variables (see section 6.6.5) allow for direct access of device process data via device property identifiers. Just click with the right mouse button in the input field and select the menu item *Insert device property* (see figure below).

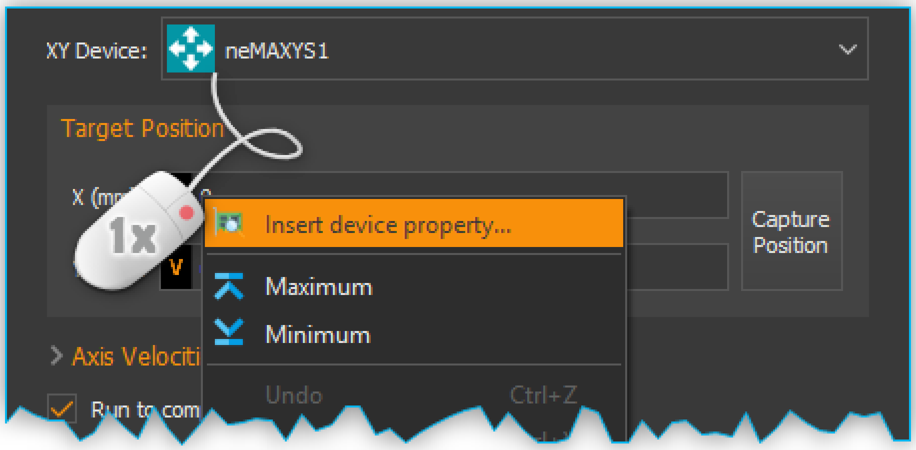


Figure 99: Inserting device properties into input fields

A dialog for selecting the process data is displayed (see figure below).

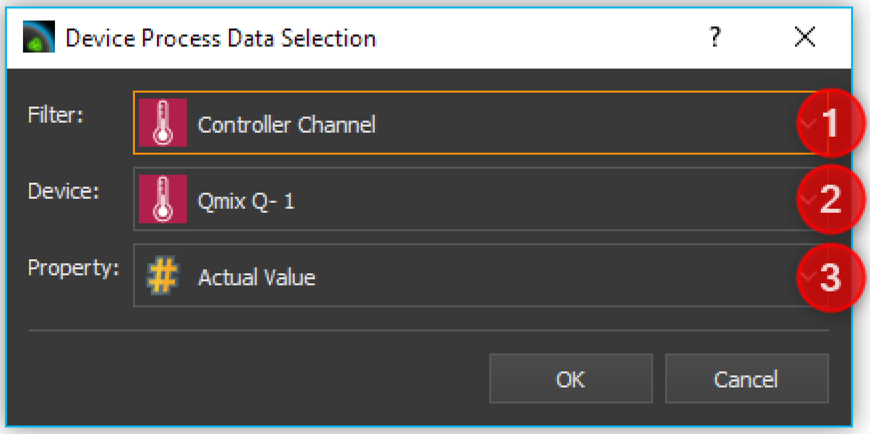


Figure 100: Selection dialog for device process data

In this dialog you can select which type of device you want to access in the selection box *Filter* ❶. Select a specific device in the selection box *Device* ❷ which contains the filtered list of devices. Finally select the process data to be accessed in the *Property* ❸ field.

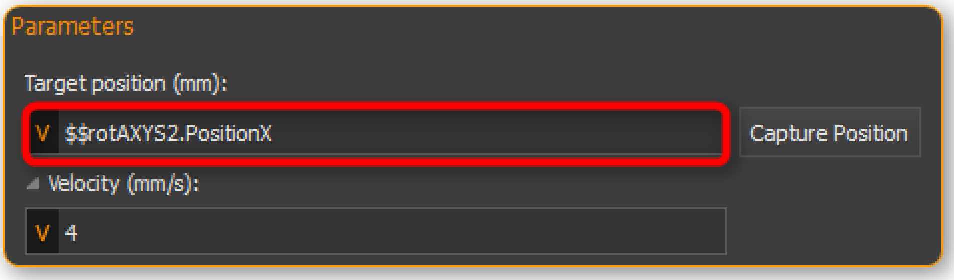


Figure 101: Using process data identifiers in input fields

The selected process data identifier will be entered into the input field. Similar to variable names, the process data identifiers have a particular form:

$$DeviceName.DeviceProperty

Each identifier starts with two dollar signs. A point separates the device name from the device property name. The entire process data identifier must not contain spaces or other special characters.

IMPORTANT. The device name and the name of the process data have a normalized form. All spaces are removed and replaced by underscores. The device name is the unique name of each device and may be different from the device caption that can configured by the user.

When the script function is executed, the process data is read from the device and used as function parameter for the script function.

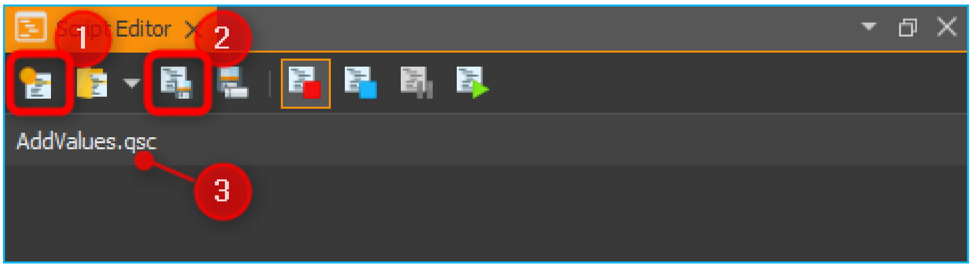
6.8 Programming your own script functions

6.8.1 Create a script function

In addition to the script functions available in the script pool, you have the option of programming your own script functions to use them later in your scripts. To implement an own script function, proceed as follows:

Step 1 – Create a new script

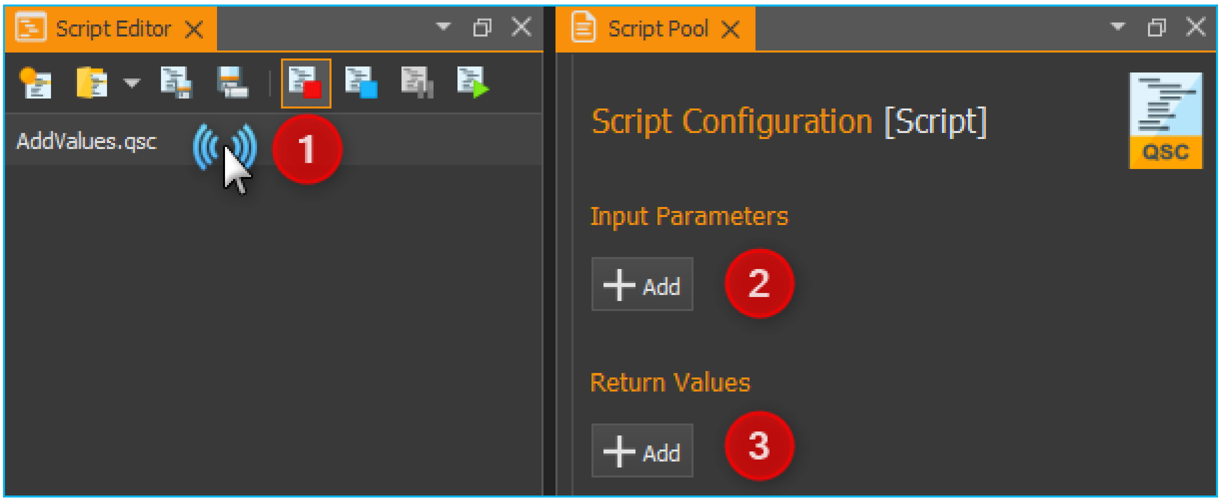
Click on the button *Create New Script* 1 to create an empty script. Then click on the *Save Script* button 2 to give the script function a name and then save it with this name. The name of the script function is then displayed in the script editor header 3. In this example we use the name *AddValues* because we want to implement a simple function that adds two values.



Step 2 – Define function parameters

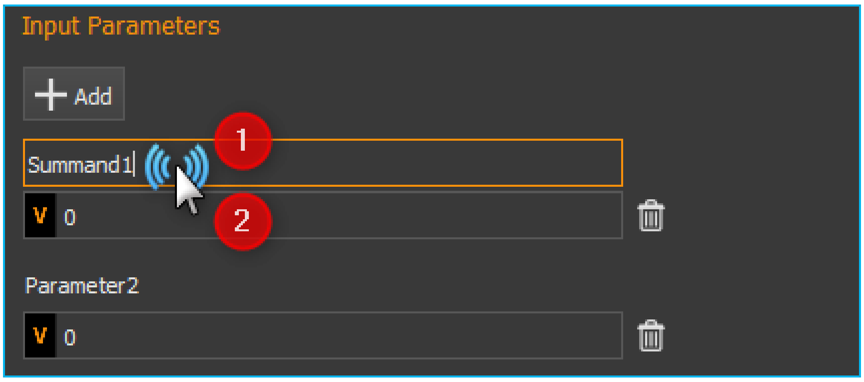
You can define function parameters and return values for your function. Function parameters are values that are passed to the function when it is called. Return values are values (e.g. results of calculations) that are returned by the function to the calling script. Up to 10 function parameters and up to 10 return values can be defined for each function.

To define parameters and return values, click with the mouse on a free area in the script editor or on the script editor's header 1 (figure below), where the name of the function is displayed.



The script pool now shows the configuration window for the script. Here you can add function parameters ② or return values ③ by clicking on the *Add* buttons.

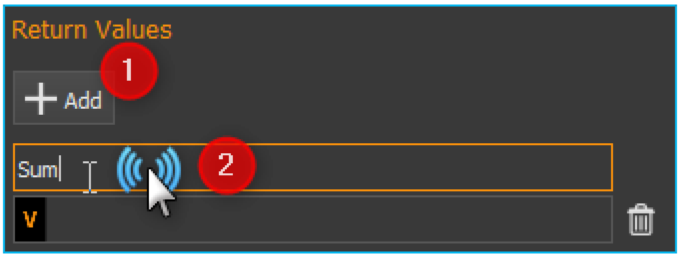
For this example, click twice on the *Add* button ② to add two function parameters. Then click on the first parameter name (figure below) and give it a more meaningful name: *Summand1*:



Then enter the default value 0 ② for the parameter. Now click on the second parameter name and rename it to *Summand2*.

Step 3 – Define return values

Now click once on the *Add* button in the *Return Values* ① area (figure below) to add a return value. Then click on the first return value name and rename it to: *Sum*.

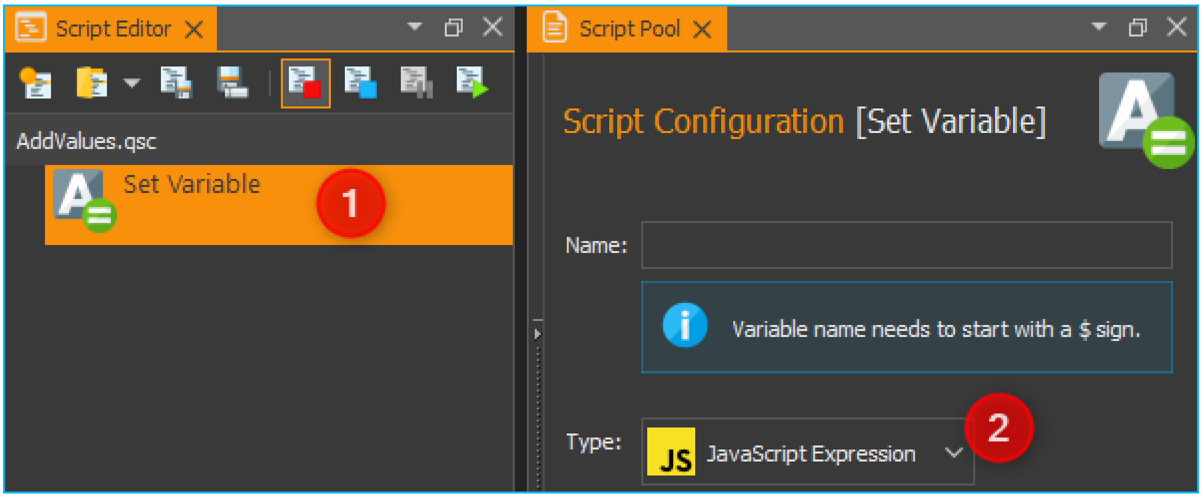


Now save the script function to store your changes. The configuration area of the script function should now look like this:

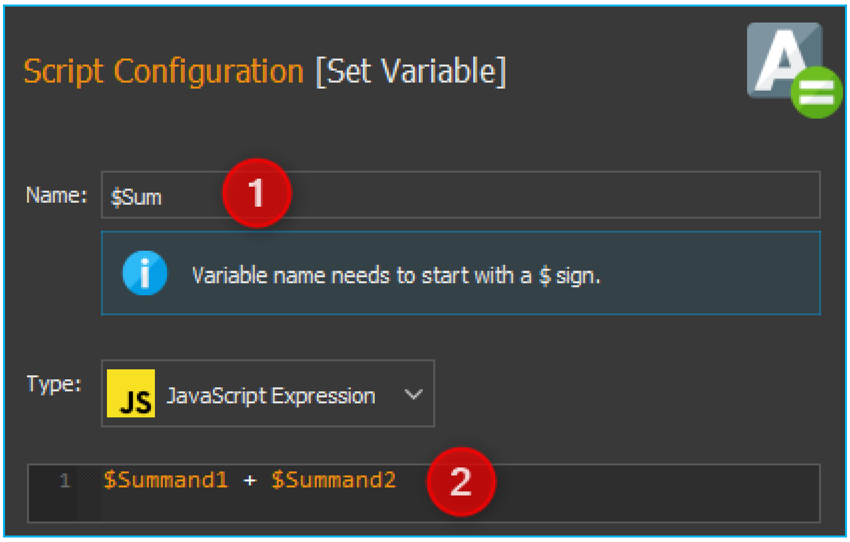
Step 4 – Implement function logic

All function parameters and return values are available within the script as variables that can be read and written. I.e. the script can now read the transferred values from the two variables *$Summand1* and *$Summand2* and save the result of the calculation in the variable *$Sum* and thus transfer it back to the calling script.

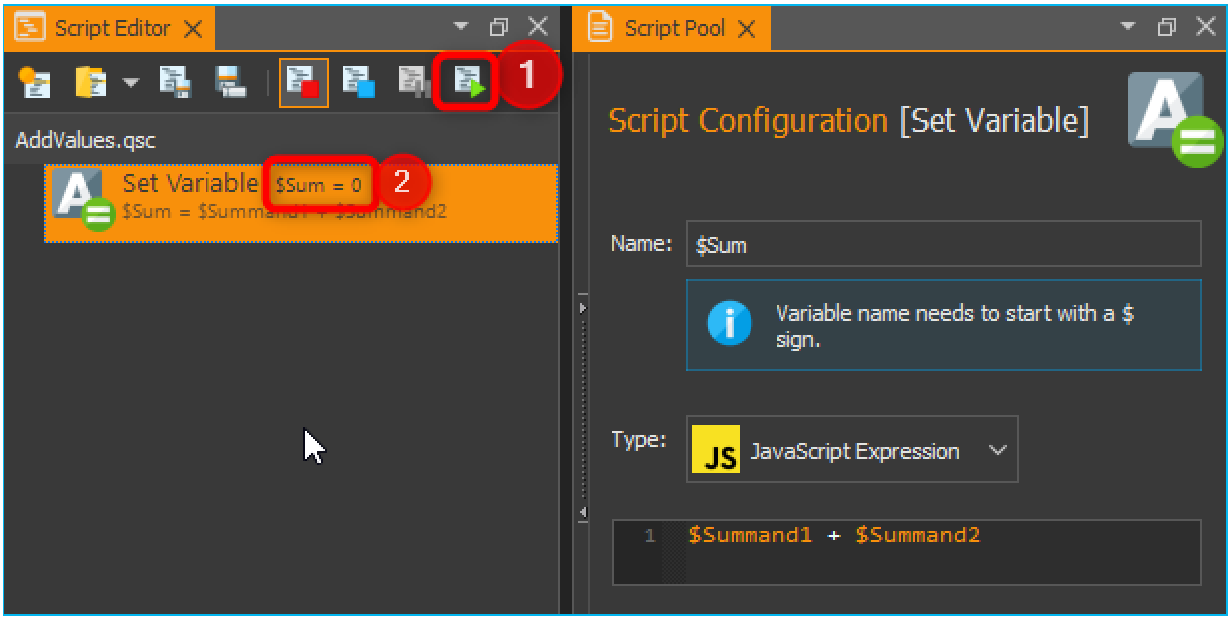
To perform the addition, insert a *Set Variable* script function into the script and set the type of the variable to *JavaScript Expression*.



In the *Name* field, enter the name 1 (see figure below) of the variable which should store the value – in this case, the variable *$Sum*. In the input field for the JavaScript code 2 you can now enter the addition of the two variables *$Summand1* and *$Summand2*.



Now save the script function. Next, you can test the script function. Click on the *Run Script* button 1 (figure below) - no error should occur and the result of the addition should be displayed in the script editor in the *Set Variable* function 2.

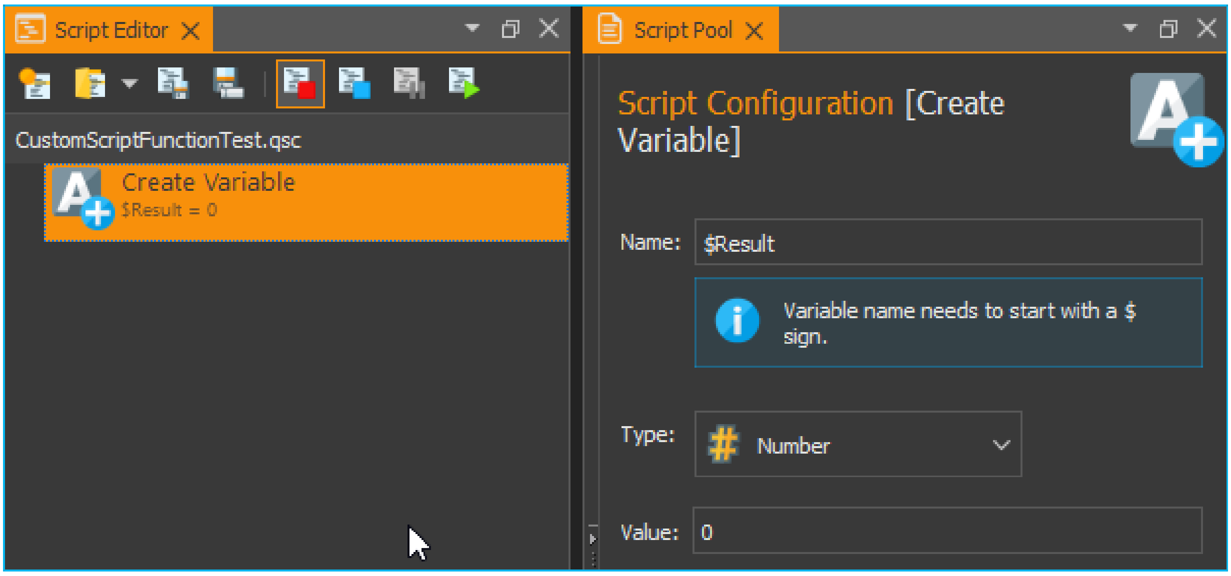


If errors occur, correct them and save the script function again.

6.8.2 Use your own script functions

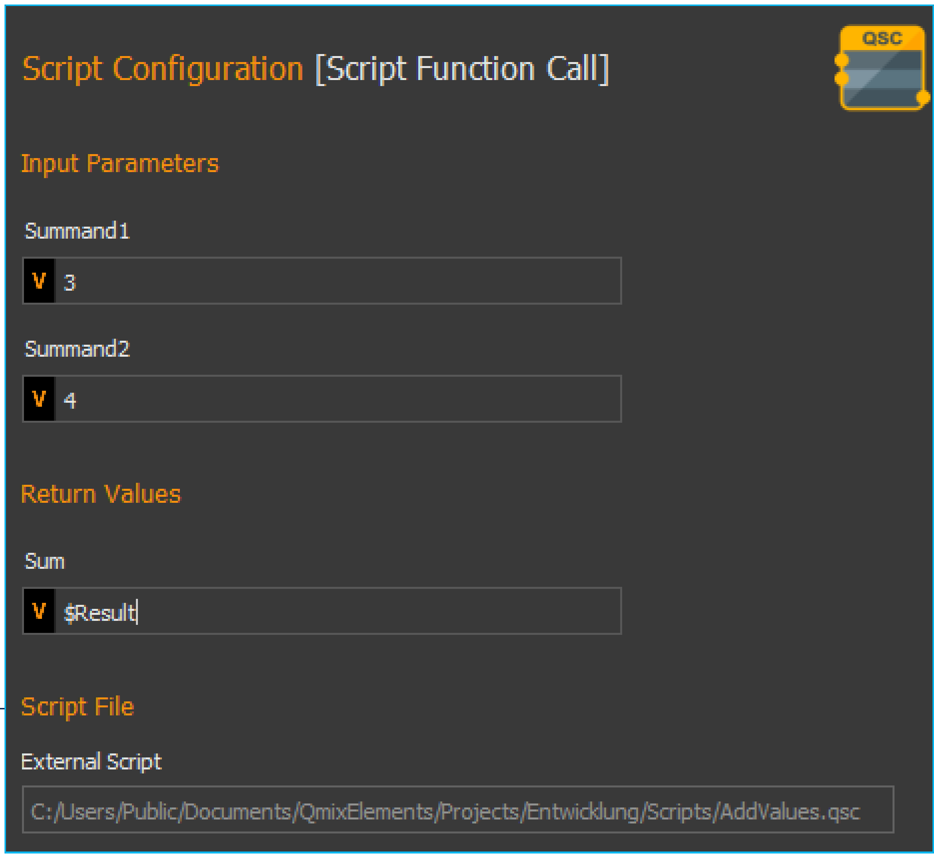
Click on the button *Create New Script* ❶ to create an empty script. Then click on the *Save Script* button ❷ to give the script function a name and then save it with this name. The name of the script function is then displayed in the script editor header ❸. In this example we use the name *CustomScriptFunctionTest*.

Add a *Create Variable* function to the script as the first function and configure the function as follows.

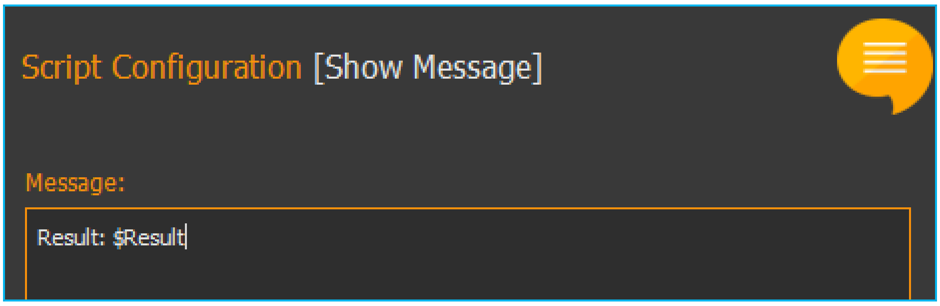


Now insert a *Script Function Call* from the *Core Functions* category into the script as the second function.

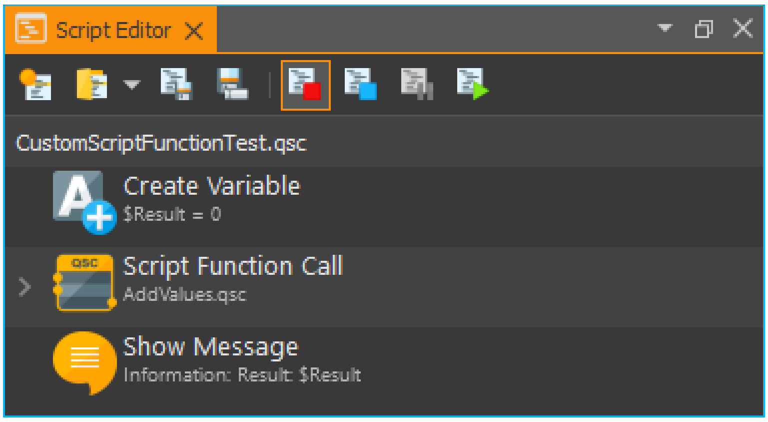
A file selection dialogue pops up where you can select the external script function to be called by the script. Select the example function *AddValues.qsc* that we created in the previous section. For the function parameters *Summand1* and *Summand2*, enter two values as a test, e.g. 4 and 3. You can also use script variables in these fields. Enter the variable *$Result* in the return parameter *Sum*. In other words, the return value of *Sum* is stored in the variable *$Result*. The function should now be configured as follows:



Now add a *ShowMessage* function as the last function to output the value of the *$Result* variable. Enter the following in the *Message* field:



Your script should now look like this:



Now run the script. The *Show Message* function should show you the result of the call to *AddValues.qsc* in a window and in the event log.

By using your own script functions, you can structure your script and break it down into reusable and easily maintainable individual components.

6.9 Script Autostart

The script system can be configured to automatically load and start a script after successfully connecting to the device hardware. The dialog with the corresponding settings can be opened via the menu item *Edit* → *Settings* in the main menu of the application.

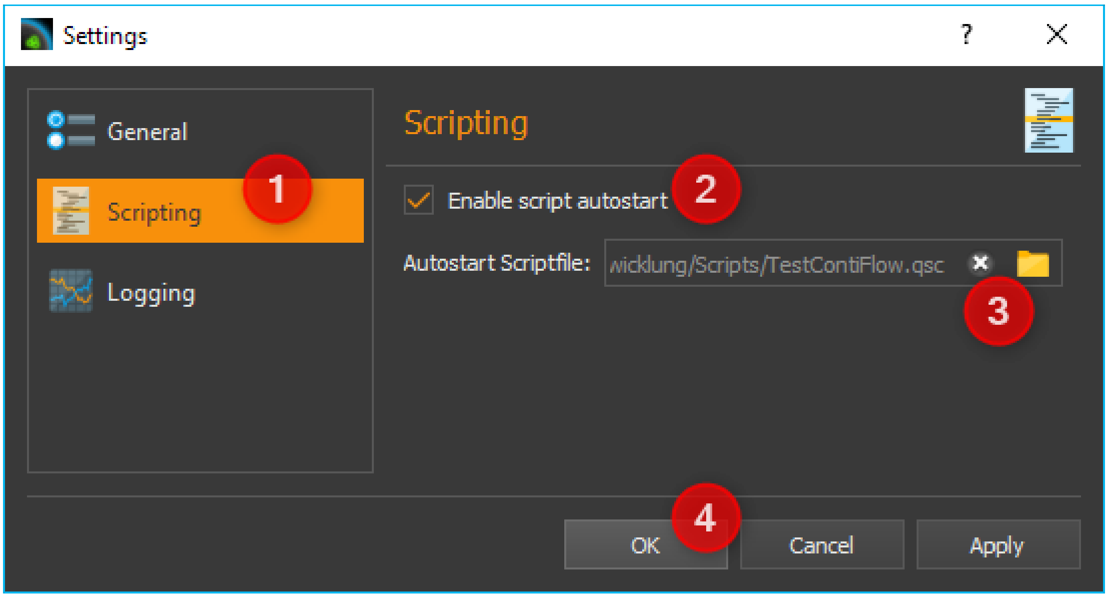


Figure 102: Script autostart configuration

You can now configure the script autostart in the Settings dialog of the application. First select the settings category *Scripting* 1. Now you can activate the script autostart 2 in the right area. You can select the script file to be loaded and executed via the *Autostart Scriptfile* 3 input field. If this field is empty, the script is executed which is loaded when the application is started, i.e. the script which was open when the application was last used. Finish the configuration by clicking *OK* 4.

If you want the software to start up automatically and execute a script after your computer has booted, then proceed as follows:

1. Insert *QmixElements.exe* into the *Windows Autostart* to start the software automatically after booting the computer.
2. Open the dialog with the global settings via the main menu of the application (*Edit* → *Settings*).
3. Select the *General* settings category and activate the *Enable auto connect* option. This will cause the software to automatically connect to the device hardware after start-up.

4. Then select the *Scripting* setting category to configure the script autostart

6.10 Script Error Handling

Errors may occur during the execution of individual script functions, e.g. if parameters are outside the value range or if errors occur during communication with devices. You can configure how the script system should react to such errors in the global settings dialog (select *Edit* → *Settings* in the main menu of the application).

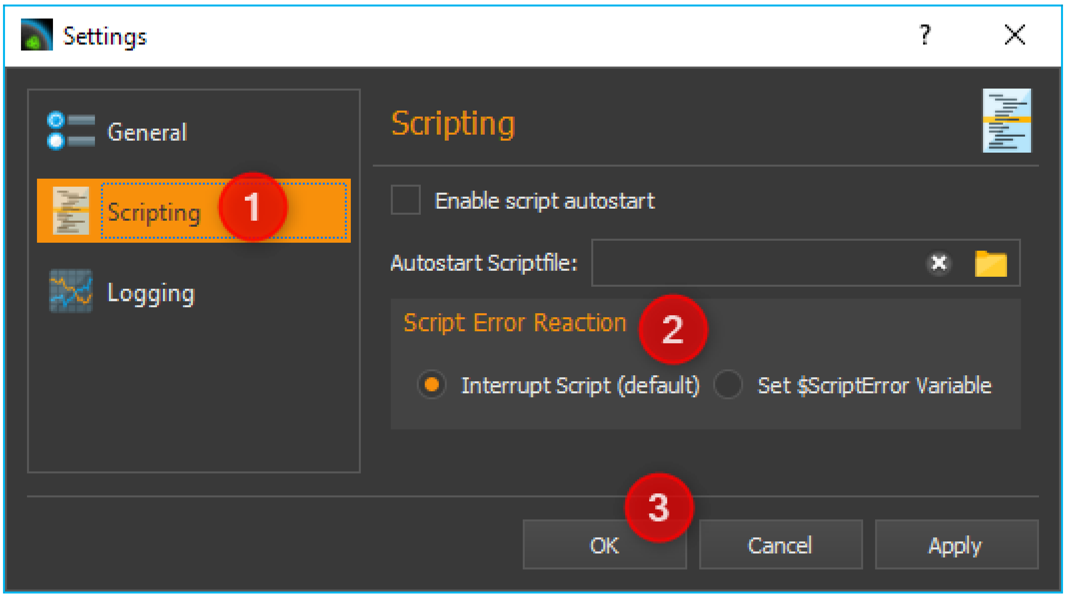


Figure 103: Error Handling Settings

First select the *Settings* category Scripting 1. On the right side of the *Script Error Reaction* 2 you can now configure the reaction in case of an error. You can select the following options:

- **Interrupt Script (default)** – This is the error reaction that is active by default. If an error occurs in the script, the script stops at the function that caused the error and an error message is displayed in the event log. The script can then only be continued by clicking the start button. This may not be desirable for automatic control via the I/Os of a PLC. In this case select the following type of error handling.
- **Set $ScriptError Variable** – If an error occurs, the global script variable *$ScriptError* is set to true and script execution continues. In this case, you must deal with the error handling in the script by checking the status of this error variable after a function call.

Complete the configuration by clicking on OK 3

6.11 Generic Script Functions

The *Script System* comes with a set of generic, or core, functions that control the program flow. These functions are always available irrespective of the activated plug-ins. You will find these functions in the categories *Core Functions*, *Function Sequences*, and *Time Functions* (figure below).

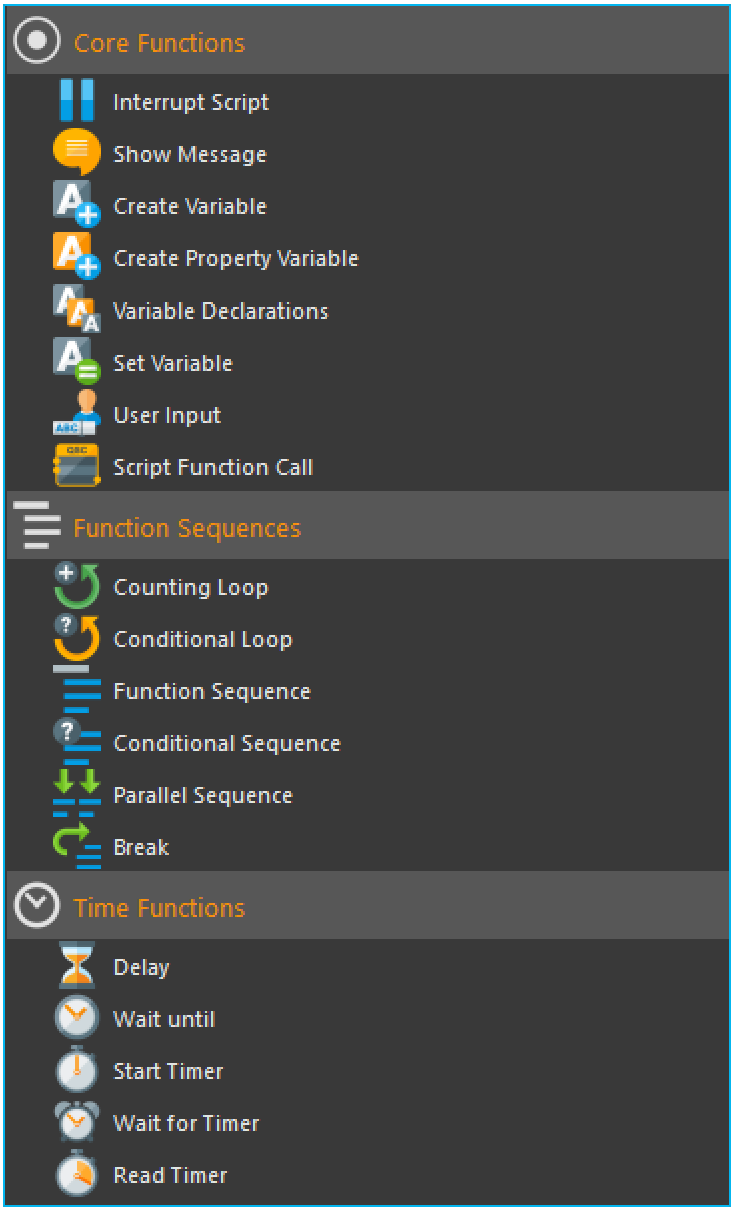


Figure 104: Generic script functions

6.12 Core Functions

6.12.1 Overview

The category Core Functions contains functions that control program flow and the use of variables.

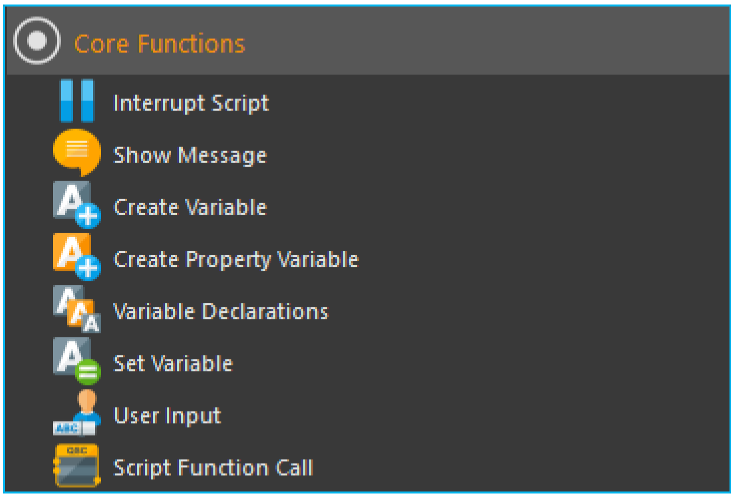


Figure 105: List of core functions

6.12.2 Interrupt Script

This function allows you to pause script execution. The user may then continue the script at a later time.

6.12.3 Show Message

The *Show Message* function allows you to display a message to the user and optionally interrupt script execution. It is well suited for applications that require an intervention by the user at certain times during the execution of a script.

When you run this script function, the message can be displayed both in a message window...

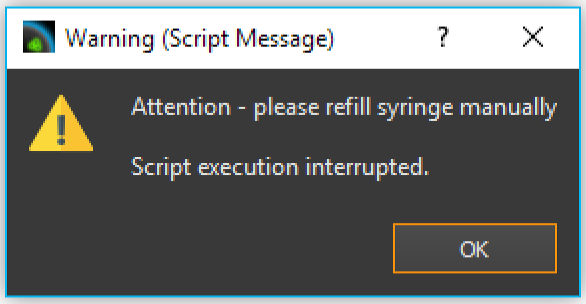


Figure 106: Message window

... and in the event log of the QmixElements software window:

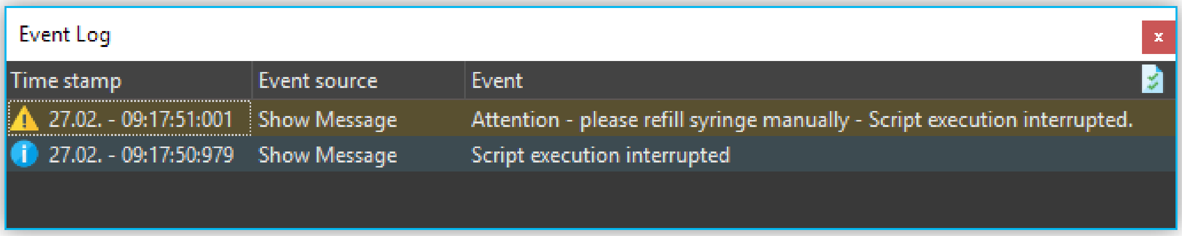


Figure 107: Event log

In the *Event Log*, all messages are listed in chronological order. The optional Message Box always displays the last message.

6.12.3.1 SHOW MESSAGE FUNCTION CONFIGURATION

When configuring the message function, you can configure the following options:

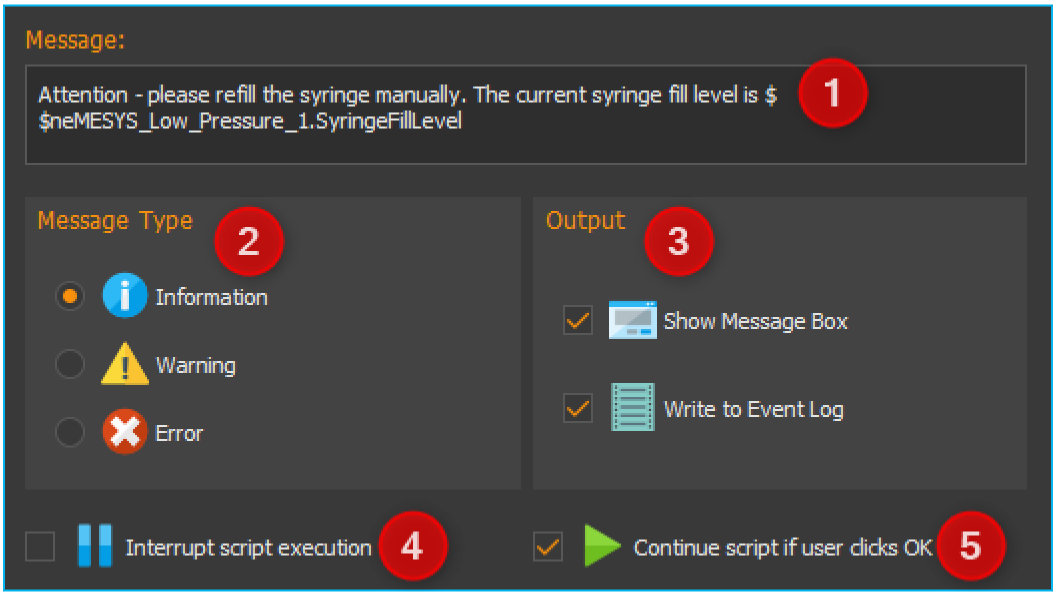


Figure 108: Message configuration panel

1 Enter your message text into the *Message* field.

2 Select the type of message you want to display in the *Message Type* group. Depending on the message type, the relevant icon appears in the event log and in the message window and the message is highlighted with a type-specific color in the event log.

3 In the *Output* pane you can set whether the message should be displayed in a separate message window (*Show Message Box*) and whether the message should appear in the Event Log (*Write to Event Log*).

4 Select whether the message should interrupt script execution. If there is a check mark in this field, the complete script execution is interrupted until the user continues the program execution. If you display a message window with a *Show Message Box*, you should not select this option and use option 5 instead.

5 Here you can select whether the script execution will be continued after clicking *OK* in the message box window.

Depending on the options selected, this function works like this:

SHOW MESSAGE BOX	WRITE TO EVENT LOG	INTERRUPT SCRIPT EXECUTION	CONTINUE SCRIPT IF USER CLICKS OK	RESULT
	☑			A message is written to the event log and then the next function is executed.
	☑	☑		A message is written to the event log and script execution is interrupted. The execution can be continued by clicking the *Run Script* button in the script toolbar.
☑	☑			A message is written to the event log, a message box is displayed and then the next function is executed.
☑			☑	A message box is displayed. When the user clicks OK, the next script function is executed.
☑		☑		A message window is displayed and script execution is interrupted. Execution can be continued by clicking the Run Script button in the script toolbar.

6.12.3.2 DISPLAYING VARIABLES CONTENT OR PROCESS DATA

In the *Message* field of the function, you can also use variables (like *$Volume*) or process data identifiers (like *$$neMESYS1.SyringeFillLevel*). These serve as place holders and are replaced at script runtime by the contents of the variables or the process data of a certain device. As soon as you enter a dollar sign *$* in the message input field, a list of available variables is displayed, from which you can select a variable (see figure below).

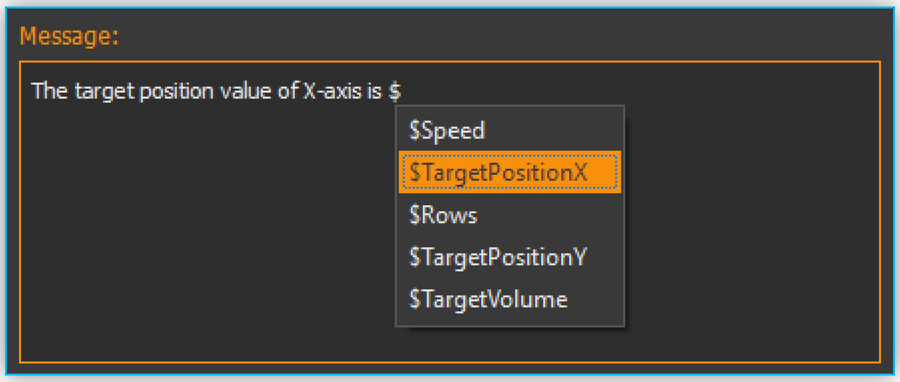


Figure 109: Using variables in Show Message function

```
The contents of variable A is $A.
```

If the variable *$A* contains the value 25.3 at runtime, then the function will display the following message:

```
The contents of variable A is 25.3.
```

HINT. You can display the contents of variables using the Show Message function, e.g. to check the results of calculations.

HINT. Click with the right mouse button in the input field and then select the context menu item **Insert device property** to insert a process data identifier of device process data you would like to print out, when Show Message function is executed.

6.12.4 Creating a Variable

With *Create Variable* you can define a new variable and assign a name, a type, and a value to that variable (see figure below):

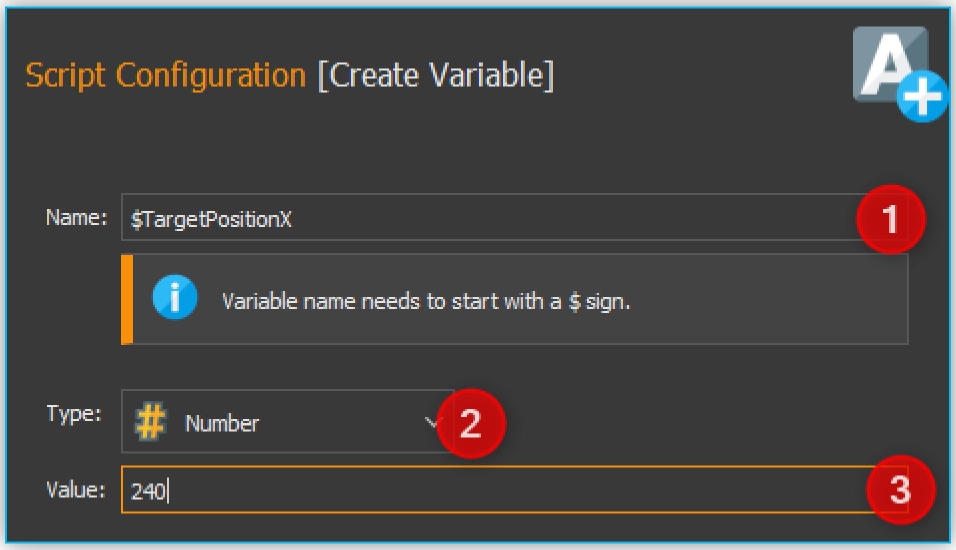


Figure 110: Configuration of a variable

1 *Name* – set a unique name for your variable. For details concerning the naming of variables,

please refer to section 6.6.3.

2 *Type* – choose the kind of variable you want to set.

3 *Value* – give the variable a value or, if the variable is of the type *Calculation*, enter the mathematical expression to calculate the variable.

HINT. To prevent an incorrect variable name, the **Name** field accepts only names that start with a $-symbol.

6.12.4.1 TYPES OF VARIABLES

The variables in the script system are not typed and can hold a wide variety of values. For example, a numeric value can be assigned to a variable when it is created and a text can be stored in the same variable later. The *Type Hint* field in the configuration area is simply a hint as to which type is to be stored and is used to display a corresponding input field depending on the type. All types, *Boolean*, *Number* or *Text* can also be entered via a JavaScript expression. The following table is an overview of a selection of data that can be stored in variables.

Type Hint	Data / Values	Examples
Boolean	Boolean values	`true, false`
Number	Numeric values	`24, 10, -13.23, -128, 0.001`
Text	Strings, words or text	`Value1, Enter a value`
JavaScript	Device References (e.g. for passing devices to functions)	`$$Nemesys_M_1` or `$$Nemesys_M_1_ForceSensor`
JavaScript	Array of numbers	`[3, 7, 8.5, 12, 23.5]`
JavaScript	Array of strings	`[User1, User2, User3, User4]`
JavaScript	Array of device references	`[$$Nemesys_M_1, $$Nemesys_M_2]`

JavaScript	Calculation result	`25 + $TargetPositionX`
JavaScript	Read device properties / process data	`$$Nemesys_M_1.DosedVolume`

6.12.4.2 CALCULATING USING VARIABLES

If you set the variable type to Calculation, you can define an expression in the field *Value* that is to be calculated during run time. This expression may include other variables to use their respective values. Script variables can be accessed via the variable name, which starts with a dollar sign.

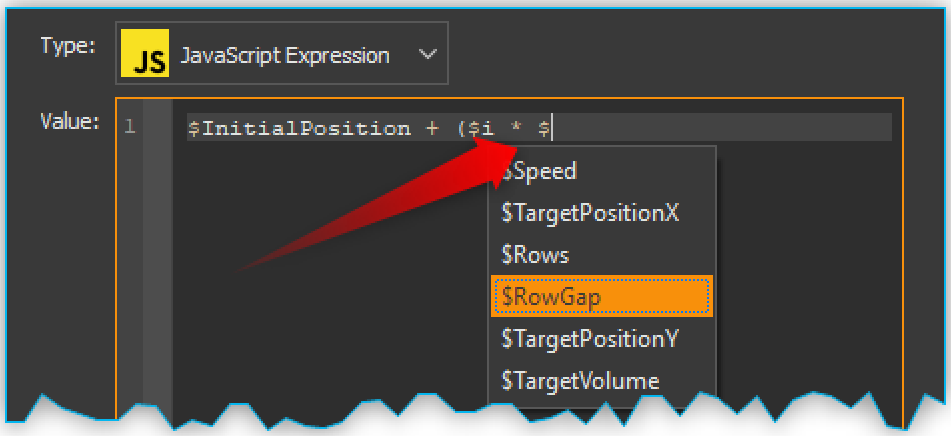
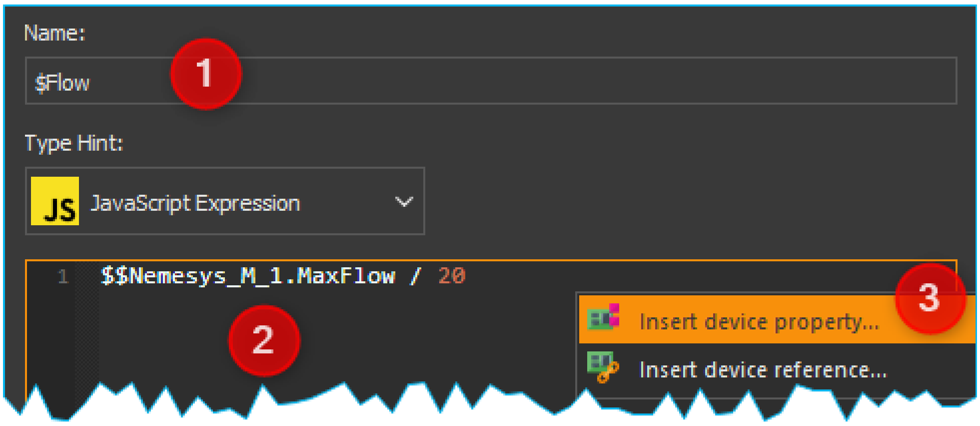


Figure 111: Defining calculation variables

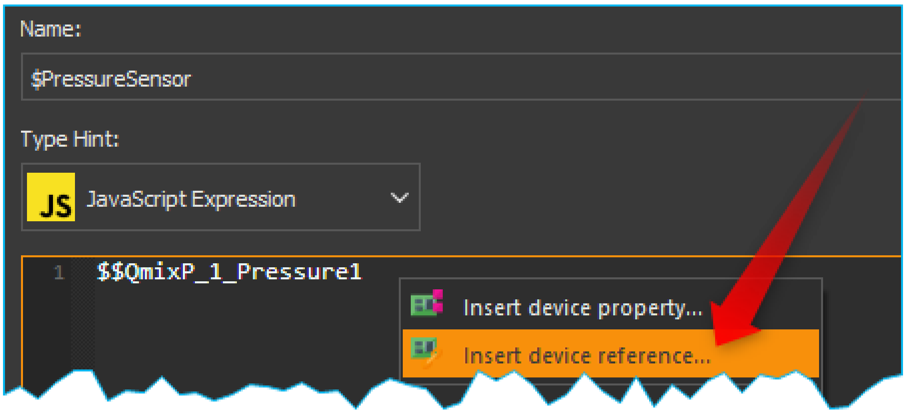
HINT. The auto-complete feature will support you during entering a new expression by listing all previously defined variables.

You can also directly read the properties / process data of a device within a JavaScript expression and include it in a calculation. In the following example, the variable `$Flow` 1 is calculated as a twentieth of the device property `$$Nemesys_M_1.MaxFlow` 2. Device properties can be easily added using the context menu (*Insert device property*) 3.



HINT. Click with the right mouse button in the input field and then select the context menu item **Insert device property** to directly access device process data values in the calculation.

Using the context menu you can also insert device references into the JavaScript code. This makes it possible to store device references (e.g. `$$Nemesys_M_1`) or arrays with device references (e.g. `[$$Nemesys_M_1, $$Nemesys_M_2, $$Nemesys_M_3]`) in variables. In the following example, a reference of the $$QmixP1_Pressure1 pressure sensor is stored in the `$PressureSensor` variable.



The identifiers for device references start with two dollar signs like Device Properties, but consist only of the device name without any additional device property.

6.12.4.3 OPERATORS

The following tables summarize available operators and their syntax:

Arithmetic Operators		Example
+	Addition	`$a + $b`

-	Subtraction	`$a - 1`
*	Multiplication	`5 * $b`
/	Division	`$a / 10`
%	Division and returning the remainder	`$a % 5`

Logical Operators		Example
==	Checks for equivalence	`1 == 2`
!=	Checks for difference	`5 != 6`
>	Checks for Larger Than	`7 > 2`
<	Checks for Smaller Than	`3 < $a`
≥	Checks for Larger or Equivalent	`$b ≥ 3`
≤	Checks for Smaller or Equivalent	`$a ≤ $b`
&&	Links two values with the logical AND	`true && false`
||	Links two values with the logical OR	`$a || (b < 5)`
!	Links a value with the logical NOT	`!true == false`

String Operators		Example
+	Assembles two strings or texts	`text + $a`

6.12.4.4 OPERATOR HIERARCHY

The operators are ranked by a predefined hierarchy. When complex calculations are to be carried out, which combine a number of different operators, these operators will be resolved according to the following order:

1. rank: `||`
2. rank: `&&`
3. rank: `==` `!=`
4. rank: `<` `≤` `>` `≥`
5. rank: `+` `-`
6. rank: `*` `/` `%`
7. rank: `!`
8. rank: `()`

Parentheses allow you to influence the hierarchical order in which the various operators are executed according to your needs.

6.12.4.5 EXAMPLES

...for the use of *Calculation* variables:

EXAMPLE 1

Adding two variables *a* and *b* and storing the result as a third variable *c*:

Name: `$c`
JavaScript: `$a + $b`

EXAMPLE 2

Multiplication of the variable *Speed* with 5 and storing the result as the same variable:

Name: `$Speed`
JavaScript: `$Speed * 5`

EXAMPLE 3

Connecting the two Boolean variables *SwitchedOn* and *CheckOk* with the logical AND and save the result as the variable *TurnOn*:

Name: `$TurnOn`
JavaScript: `$SwitchedOn && $CheckOk`

EXAMPLE 4

You want to assign the current flow rate of the pump *Nemesys_M_1* to the variable *$Flow*.

Name: `$Flow`
JavaScript: `$$Nemesys_M_1.ActualFlow`

EXAMPLE 5

You want to create an array with 4 flow rates and assign it to the *$TargetFlows* variable.

Name: `$TargetFlows`
JavaScript: `[0.12, 0.05, 0.2, 0.21]`

EXAMPLE 6

The *$TargetFlows* variable contains an array of 4 numeric values. You want to assign the third value in the array to the *$NextFlow* variable. The index for the array access is numbered starting from 0. I.e. to access the third element the index is 2.

Name: `$NextFlow`

JavaScript: `$TargetFlows[2]`

EXAMPLE 7

You want to store an array with the two current values of the pressure sensors *QmixP_1_Pressure1* and *QmixP_1_Pressure2* in the variable *$Pressures*.

Name: `$Pressures`

JavaScript: `[$$QmixP_1_Pressure1.ActualValue, $$QmixP_1_Pressure2.ActualValue]`

EXAMPLE 8

You want to store a device reference of the *Nemesys_M_1* pump in the variable *$ProcessPump* to be able to pass it to a function later.

Name: `$ProcessPump`

JavaScript: `$$Nemesys_M_1`

6.12.5 Set the Value of a Variable

With *Set Variable* you can assign a new value to an already existing variable. The configuration of the variable is equivalent to the function *Create Variable*.

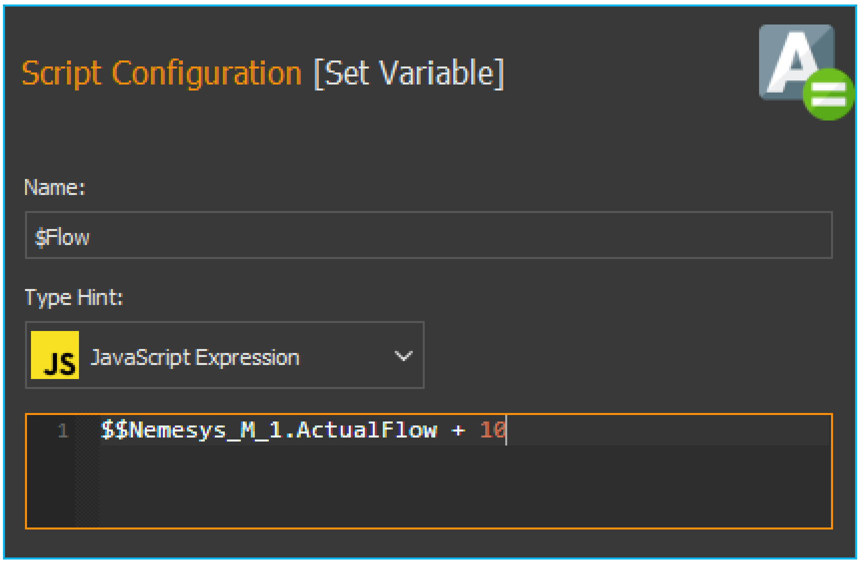


Figure 112: Entering a variable name using the auto-complete feature

HINT. When entering a name, the auto-complete feature will provide you with a selection of all previously defined variable names.

IMPORTANT. To be able to set the value of a variable using the **Set Variable** function, the variable has to be defined beforehand, e.g., by using the function **Create Variable**.

6.12.6 Create Property Variable

Use this function to create a special variable that is associated with a certain property of a certain device. Via this variable you can directly access the assigned device property from within a script, without any additional function calls.

If you e.g. want to read an analog input value using normal variables, you first need to create a variable via *Create Variable* function and then use the *Read Process Data* function to store the analog input value into this variable. Only then you can use this variable for further calculations and tests in script functions. Whenever you need a new analog input value, you must call the *Read Process Data* function again.

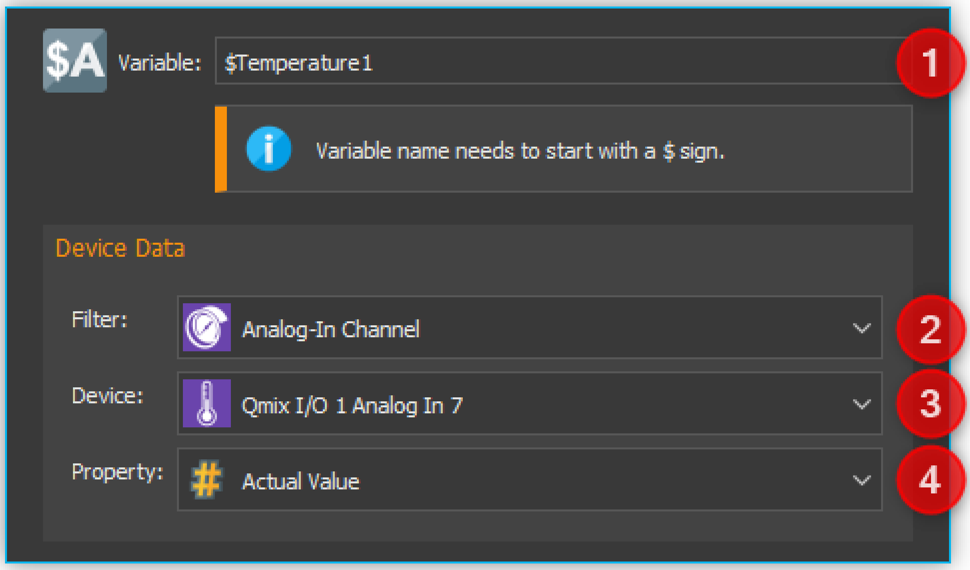


Figure 113: Create Property Variable configuration panel

With property variables that access to device properties (e.g. analog input value) is considerably simplified. If you create a property variable, you use the configuration panel to connect a device property permanently with a variable. Whenever you read the value of this variable from within a script function, automatically the current value of the connected device property is read.

You can set the name of the variable in the input box *Variable* 1 (figure above). Select the device you want to access from the drop-down list *Device* 3, which lists all modules that are part of the current configuration. To simplify the search for a device, you can filter the device list by selecting an appropriate filter (device type) from the *Filter* drop-down field 2.

After the device has been selected, use the last drop-down list *Property* 4 to select the device property, the process data value that you want to access.

In summary, this function can be described as follows:

Assign the device property 4 *of the device* 3 *to the variable* 1 *and return the device property value if a read access on the variable takes place.*

IMPORTANT. Property variables can not be used to store values during calculations—use normal variables for calculations and value storage.

6.12.7 Variable Declaration Sequence

With the variable Declaration Sequence you can combine variables that you declare with Create Variable or Create Property Variable functions. This helps you to improve the clarity of your script. Instead of many individual variables, only the variable declaration sequence is visible and only when this sequence is expanded, the user sees the declared variables.

All variables are visible at the level in which the declaration sequence is located. This means that the variables within the sequence are declared as if they were at the level of the declaration sequence.

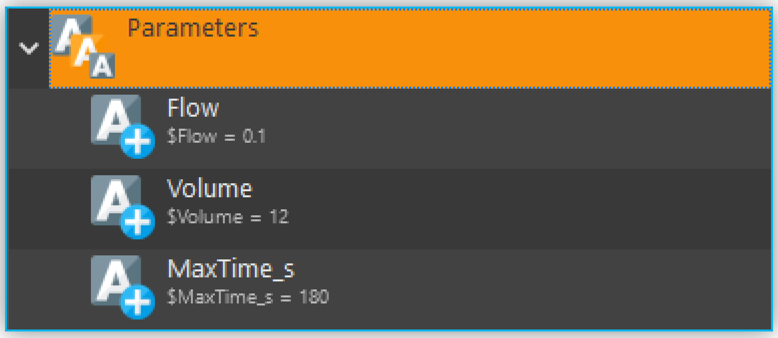
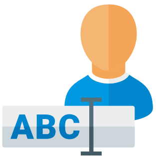


Figure 114: Example sequence with variables

With the sequence you can also group variables that logically belong together, making it easier for the user to find specific variables.

6.12.8 User Input



This function allows the reading of a single input value (number or text) into an existing variable. In the configuration area of the *User Input* function (see figure below), you can configure the following parameters:

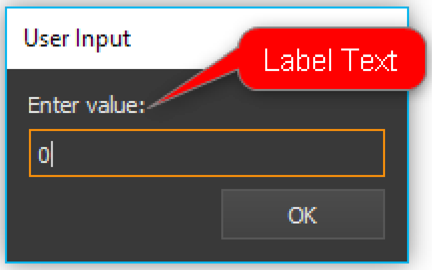
Figure 115: Configuration User Input function

1 *Label Text* – Here you specify a text that will be displayed to the user during the execution of the function above the input field as a label and explanatory text.

2 *Variable* – In this field you specify the variable to store the value entered by the user.

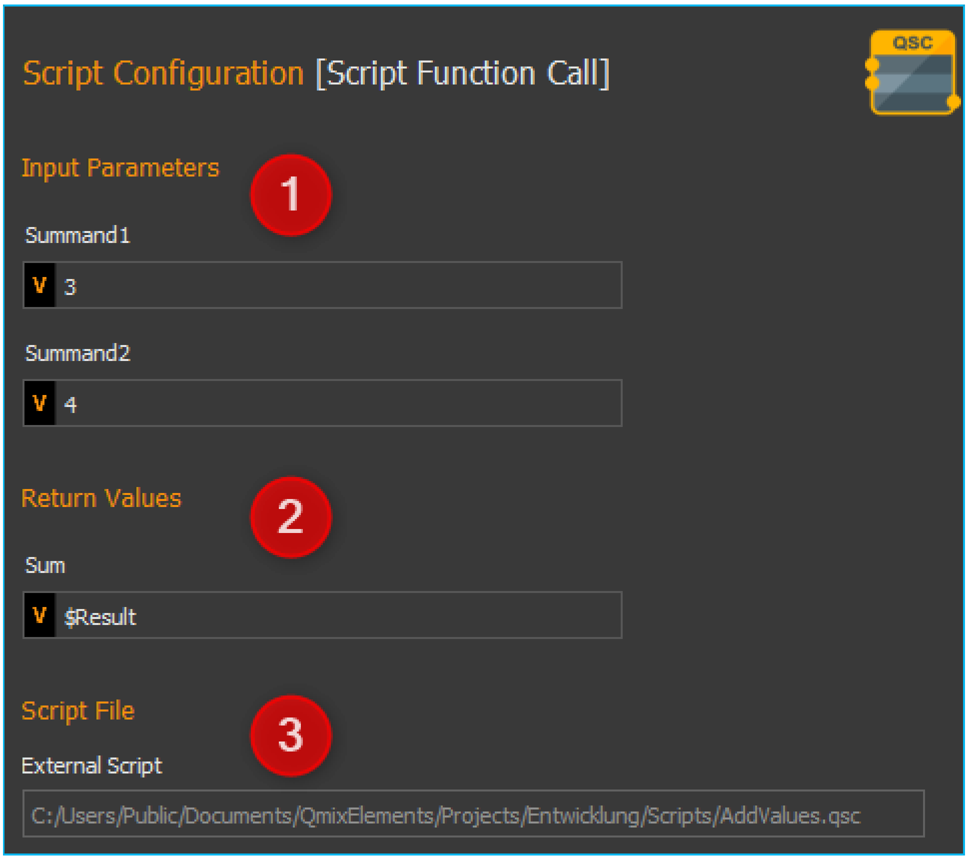
3 *Type* – You can specify the input type, number or text, here. Depending on the selected input type the user input dialog accepts only certain values.

4 *Test Input Dialog* – Clicking this button allows you to test the input dialog to check how the user will see the input dialog later during execution of the script function (see figure below).



6.12.9 Script Function Call

Script Function Calls allow you to integrate your own scripts as functions. I.e. just as you create functions as reusable modules in other programming languages, you can also call your own functions in the script system in the form of external scripts.



External script functions can have 0 – 10 input parameters ❶ and 0 – 10 return values ❷. This allows you to pass values to the script (e.g. numeric values or variables) and read return values from the script and store them in script variables. *Script File* ❸ shows you the path to the external script file that is executed when the script is invoked.

In the section *Programming your own script functions* you will find detailed instructions on how to create your own script functions.

6.13 Function Sequences

6.13.1 Overview

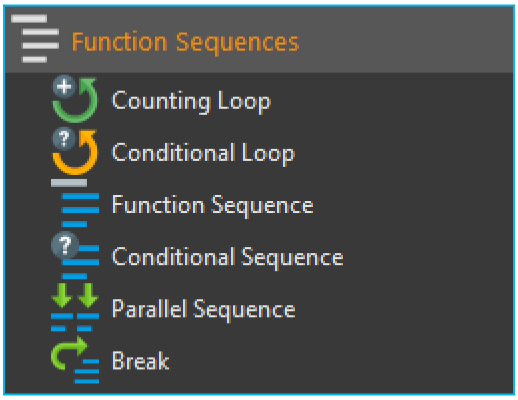


Figure 116: List of function sequences

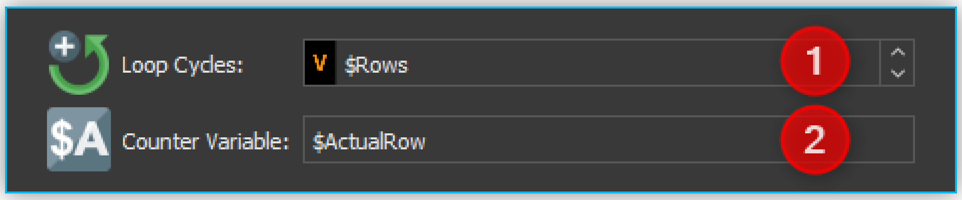
The *Function Sequences* category contains various types of function sequences. Function sequences are script functions that may contain a number of different functions. When a *Function Sequence* is called, all script functions contained within this sequence are executed.

6.13.2 Counting Loop

The *Counting Loop* function allows for multiple sequential executions of the same function(s) within that loop. The number of times the loop is to be run is defined in the configuration area (*Loop Cycles*).

In addition, you may set a name for the count variable (*Loop Counter*) of that loop. Within that loop, functions that support the use of variables may then use this variable to execute specific tasks or to perform additional calculations.

The number of loops may be defined explicitly, by setting a number, or implicitly, via a variable that returns a number (figure below, 1).



Additionally, you can assign a custom name to the Counter *Variable* of the present loop function (figure above, 2). Within this loop, other functions may access the current loop count via this variable and use it for further calculations. This may be used, *e.g.*, to control incremental moves of positioning systems.

IMPORTANT. The counting variable of a counting loop always starts with the value 0 (zero). This is, a loop count set to 10 contains the values 0 to 9 for the successive loops.

IMPORTANT. You should always insert a Delay Function in a loop to prevent the loop from blocking the application interface by consuming 100% processor power. A small delay of 10ms – 100ms is sufficient.

You can use the count variable (*Loop Counter*) to end the loop prematurely (abort). If you set the value of the count variable to a value >= the number of *Loop Cycles* within the loop using the *Set Variable* function, the loop is terminated at the next loop run.

HINT. You can end the counting loop prematurely by changing the counting variable.

6.13.3 Conditional Loop

6.13.3.1 INTRODUCTION

This conditional loop function is used for multiple execution of functions within the loop as long as the loop condition set at its beginning is met. The termination condition in the loop header is checked before each iteration of the loop. The loop execution is stopped or skipped if the termination condition is false.

Spelled out, the *Conditional Loop* function reads as follows:

Repeat the loop and execute the function(s) contained within that loop as long as the loop condition holds true.

QmixElements provides two alternatives to define a *Conditional Loop* – a simple and an advanced mode that are to be selected in the configuration panel.

6.13.3.2 CONFIGURATION - SIMPLE MODE

In the configuration panel, you may choose the simple mode to comfortably set the loop conditions directly via the Graphic User Interface – simply uncheck *JavaScript Expression* 1. The following parameters may be set:

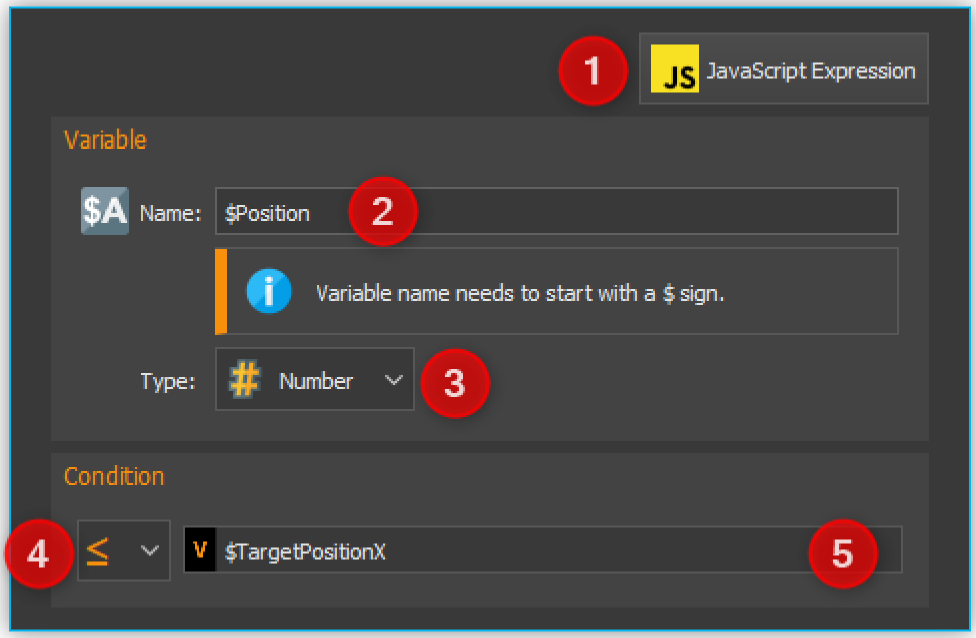


Figure 117: Configuration panel for conditional loops – basic mode

1 *JavaScript Expression* – this button is to toggle between the simple and advanced mode (see section below).

2 *Variable Name* – here you insert the name of the variable the value of which is to be tested against. The variable must start with the Dollar sign ($) that is required to signify the name as a variable within QmixElements.

3 *Variable Type* – select the required variable type. The selection will determine the kind of condition to be selected next.

4 *Comparison operator* – this drop-down menu lists all operators available for the set function type.

5 *Comparison condition* – the value or a variable that the variable from 2 is to be compared with.

SAMPLE CONFIGURATION: SIMPLE MODE

The loop condition to be checked against is whether a the variable `$Position` is smaller than or the same as `25.000,31`.

2 Variable Name	3 Type	4 Operator	5 Condition
`$Position`	`Number`	≤	`25.000,31`

The script editor shows the complete mathematical expression:

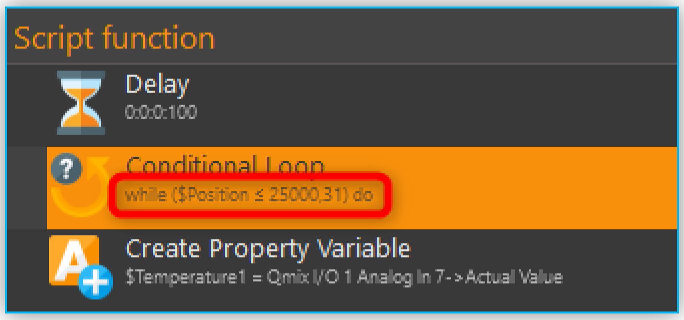


Figure 118: Example for a Conditional Loop

The resulting expression effects that the loop will be rerun as long as the value of the variable $Position is smaller than or the same as 25.000,31.

6.13.3.3 CONFIGURATION - ADVANCED MODE

Whereas the simple mode provides the user with a graphical interface, the advanced mode allows to directly insert an equation with variable names and mathematical and logical expression.

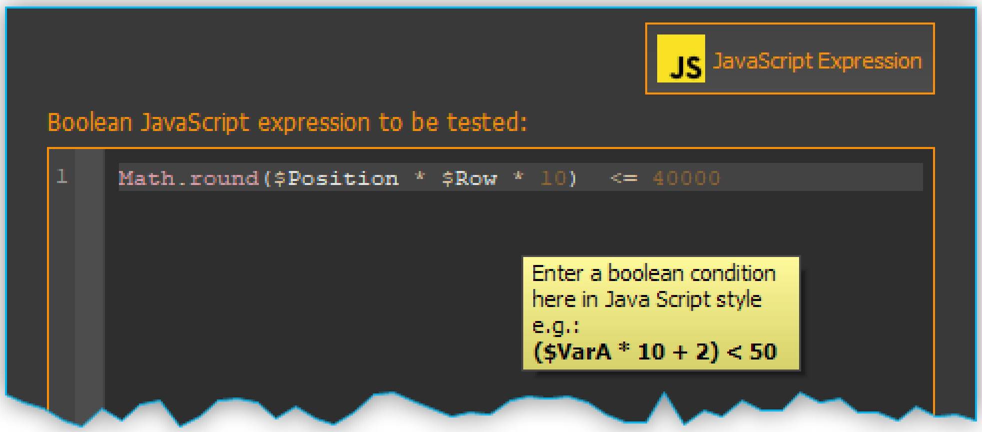


Figure 119: Configuration panel for conditional loops – advanced mode

In advanced mode, a JavaScript editor is displayed with syntax highlighting and line numbers. Here you can set the loop condition using JavaScript expressions. As shown in the example above, you can insert rather complex mathematical expressions or multiple variables to set conditions. JavaScript functions like Math.round() can also be used.

IMPORTANT. You should always insert a Delay Function in a loop to prevent the loop from blocking the application interface by consuming 100% processor power. A small delay of 10ms – 100ms is sufficient.

6.13.4 Function Sequence

A *Function Sequence* may be used to to group multiple successive functions. For one, this simplifies navigation within the function tree: The content of a function sequence may be hidden if required, thus making more of the entire tree visible within the *Script Function* panel. Also, a function sequence simplifies the setup of program scripts. Function sequences that consist of several functions may be copied to other positions within a script. This means, if you want to copy a number of successively executed functions, for instance to use them at various places in the program either sequentially or in parallel, then move them to the relevant section of your function sequence.

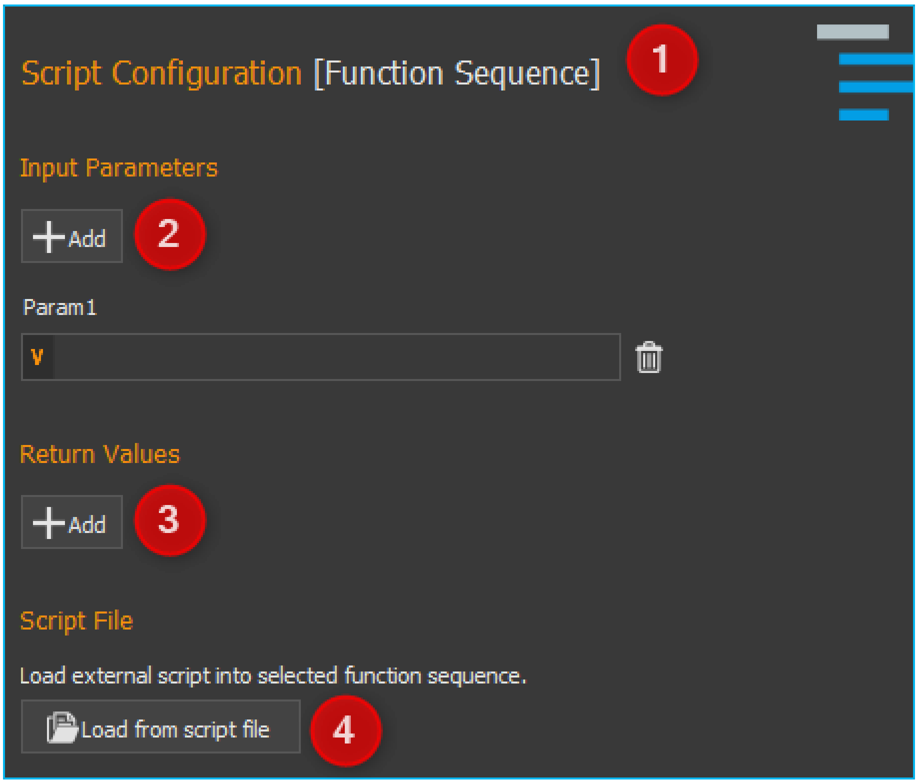


Figure 120: Function sequence configuration

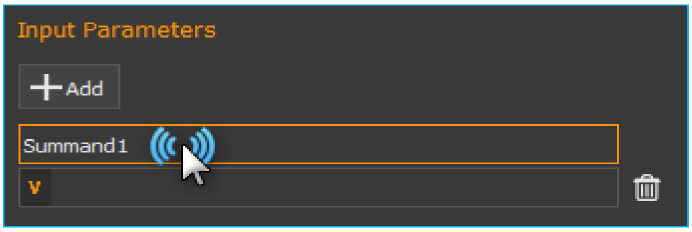
6.13.4.1 NAMING A FUNCTION SEQUENCE

A function sequence may be given a name; that name will then be shown in the function tree. This simplifies navigation in particular with larger numbers of function sequences. To do this, simply enter the name in the *Caption Field* (Figure above 1).

6.13.4.2 FUNKTIONSPARAMETER UND RÜCKGABEWERTE HINZUFÜGEN

You can access variables defined outside the function sequence from within the function sequence. Alternatively, you can also define function parameters and return values. Function parameters are values that are passed to the function when it is called. Return values are values (e.g. results of calculations) that are returned by the function to the calling script. Up to 10 function parameters and up to 10 return values can be defined for each function sequence. The functions in the sequence then only access the function parameters and not the external variables. This improves encapsulation and makes the function more independent of external variables and thus easier to maintain.

You can add function parameters 2 or return values 3 by clicking the *Add* buttons. Assign meaningful parameter names by clicking the parameter names above the input fields (figure below):



6.13.4.3 LOADING A FUNCTION SEQUENCE

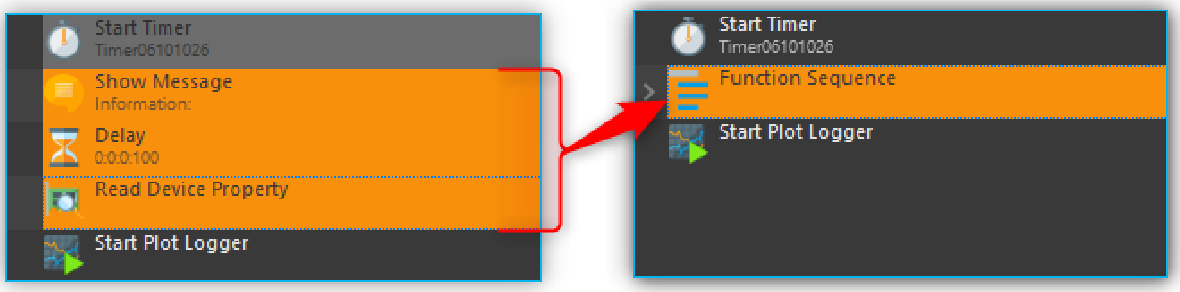
The *Function Sequence* also allows the loading of pre-stored scripts: You may load a script, which you have created and saved previously, into your current function sequence. To do this, just click on the *Load from script file* button (figure above, 4) and select the desired file. The scripts are loaded into the current script. I.e. you can make changes to the loaded function sequence without affecting the external script.

6.13.4.4 CONVERT SELECTION TO FUNCTION SEQUENCE

To improve the clarity and readability of your script, you can quickly and easily group sequences of functions into function sequences. To do this, simply select a contiguous set of functions, and then click *Convert Selection To Function Sequence* in the context menu.

Figure 121: Grouping Functions

The selected functions are now replaced by a function sequence containing the selected functions.



6.13.5 Conditional Sequence

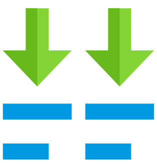
A *Conditional Sequence* provides the user with a number of script functions the contents of which will only be carried out if the condition set in its header is true. If the condition is not true at the time when it is tested, the function(s) contained within that conditional sequence will be skipped.

Spelled out, the Conditional Loop function reads as follows:

If the condition is true, carry out the function(s) contained within that sequence; else continue after that sequence.

The configuration of the Conditional Sequence is analogous to the configuration of the Conditional Loop as described above in Section6.13.3.2.

6.13.6 Parallel Sequence



The script function *Parallel Sequence* allows the concomitant execution of functions or function sequences. The configuration of this script function is identical to the configuration of a normal function sequence. In contrast, however, to a standard function sequence, the individual functions are not executed sequentially but in parallel. In other words, when executing this function, all direct child functions start simultaneously.

If you want to execute complex workflows in parallel, you should combine them in a function sequence. The following figure shows you a parallel sequence in which the three individual processes (function sequences) run in parallel:

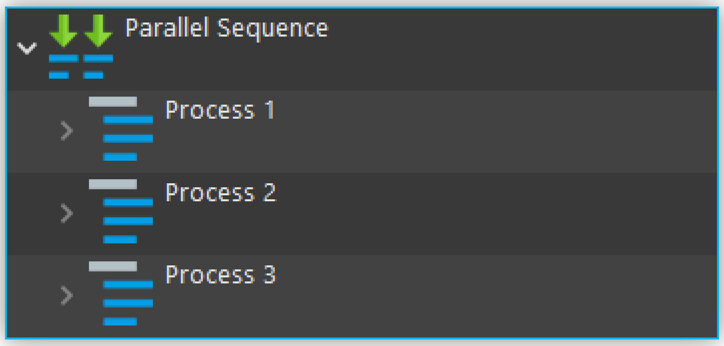
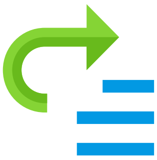


Figure 122: Example of a parallel sequence

The parallel sequence finishes only after all parallel child functions have been completed. Thus, the function is suitable for execution of parallel processes in which it is important to wait for the end of the process that takes the longest before proceeding to the next step.

HINT. Parallel sequences increase complexity and are difficult to debug, which means that it can be difficult to find errors in parallel sequences.

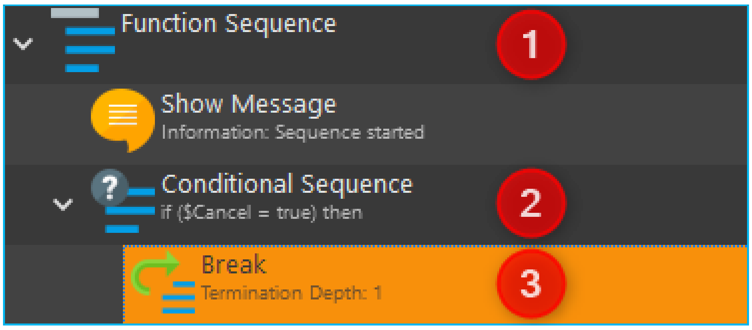
6.13.7 Break



This function causes the enclosing function sequence to terminate. This works with all function sequences from the *Function Sequences* category and with the script itself.

With the *Termination Depth* parameter you can control up to which level the enclosing function sequences are terminated. With the default value 0, only the sequence containing the break function is terminated. If the value is 1, the sequence that is one level higher is also terminated, and so on.

If, for example, you want to terminate a function sequence 1 when a certain condition occurs, insert a conditional sequence 2 into the function sequence to check the termination condition and a break function 3 with the termination Depth 1 into this conditional sequence. The break function 3 will then terminate the conditional sequence 2 and the function sequence 1 above it if the condition is true.



6.14 Time Functions

6.14.1 Overview Time Functions

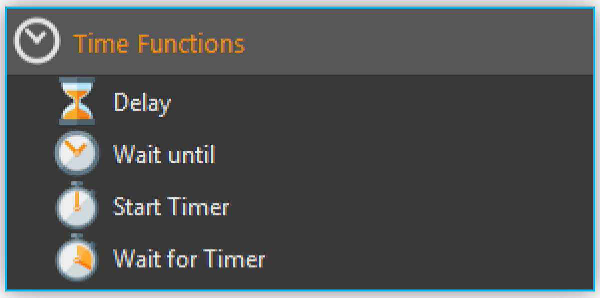


Figure 123: List of timing functions

This category contains functions to enable a time-controlled program flow. The execution of scripts may be suspended for a certain amount of time or can be interrupted to wait for certain timer events.

6.14.2 Delay

The *Delay* function interrupts program execution for a configurable length of time. You can set the delay time (hours, minutes, seconds, or milliseconds) in the configuration area (see figure below).

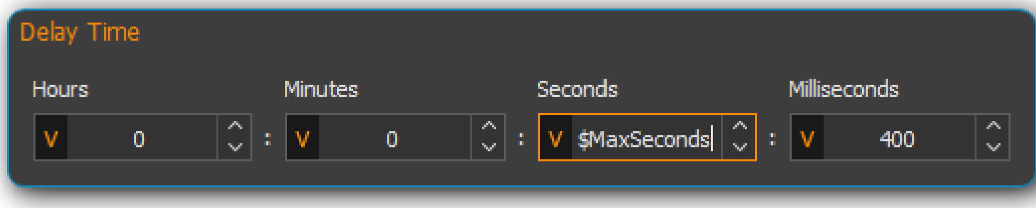


Figure 124: Configuration of Delay function

All input fields support the use of variables. You can freely and flexibly assign any values to the input fields and you can freely mix variables and values with each other.

6.14.2.1 SAMPLE CONFIGURATION

If you e.g. would like to set a delay of10 minutes and 15 seconds, you can do so by very different assignments of the input fields:

HOURS	MINUTES	SECONDS	MILLISECONDS
0	10	15	0
0	0	615	0
0	0	0	615000
0	10	0	15000

6.14.3 Wait Until

You can delay program execution until a precisely defined date and time with the *Wait until* function. Program execution is not resumed until the configured date and time has been reached.

IMPORTANT. Please note that the Wait until function can only be correctly executed once. If the script is executed again, and the time point thus lies in the past, this function will simply be ignored. In other words, date and time of a Wait until function must be re-configured before restarting a script so that an effective delay can again be executed.

6.14.4 Start Timer

This function starts a millisecond timer which can be waited for with the *Wait For Timer* function (see below) or whose current millisecond value can be read with the *Read Timer* function. For configuration, assign a unique name for the timer (*Timer Name*).

If this function is called with the name of a timer that has already been started, this timer is restarted with the configured time span.

HINT. If you want to measure the execution time for a particular process or sequence of functions in your script, simply start a timer before the sequence and read the milliseconds after the sequence with the *Read Timer* function.

6.14.5 Wait for Timer

This function interrupts the further program execution until the *Timer Running Time* configured in this function has elapsed for the selected timer. A timer with the configured timer name must have been started before via the function *Start Timer*. If the timer has already expired when this function is called, the program execution continues without interruption.

With the two above functions, *Start Timer* and *Wait for Timer*, it is very easy to execute functions at fixed intervals. For this, you can, for example, run a function or function sequence along a timer function in a *Parallel Sequence*.

6.14.6 Read Timer

With this function you can read out the current millisecond value of a particular timer and store it in a variable. You can then use the Show Message function to display this variable value. This allows you to measure and display the times of program sequences in your script, for example.

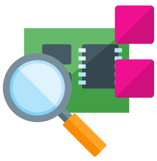
6.15 Device Functions

6.15.1 Introduction

Figure 125: Generic device functions

The script system comes with a number of generic script functions for accessing devices (device process data, device properties). These functions are available independently of the loaded plugins. You find these generic device functions in the category *Device Functions*. Read Process Data

6.15.2 Read Device Property



The function *Read Device Property* allows you to read various process data from a module and save them in a variable.

Select the device you want to access from the drop-down list *Device* 2, which lists all devices that are part of the current configuration. To simplify the search for a device, you can filter the device list by selecting an appropriate filter (device type) from the *Filter* drop-down field 1. After the device has been selected, use the last drop-down list *Property* 3 to select the device property that you want to access.

IMPORTANT. Only devices and device properties are displayed that allow read access.

In the input field *Variable* 4, enter the name of the variable into which the value is to be saved during script runtime (figure below).

In abbreviated form, the set-up follows the following logic:

Read from device 2 *the value of parameter* 3 *and save it as variable* 4.

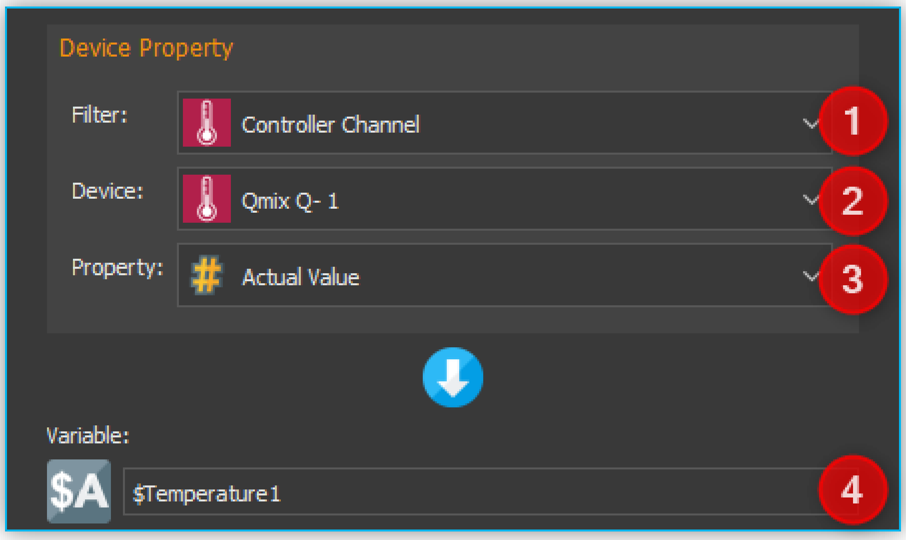


Figure 126: Reading device property set-up panel

HINT. When entering a name, the auto-complete feature will provide you with a selection of all previously defined variable names.

6.15.3 Write Device Property

This function is for writing process data from devices. In the input field 1 enter the value to be written. You can also use variables or *Device Property Identifiers* in this field.

In the configuration panel (figure below), select the device and the appropriate device property that you want to write 2 as described for function *Read Device Property*. The device selection controls 2 only show devices and device properties, which allow write access.

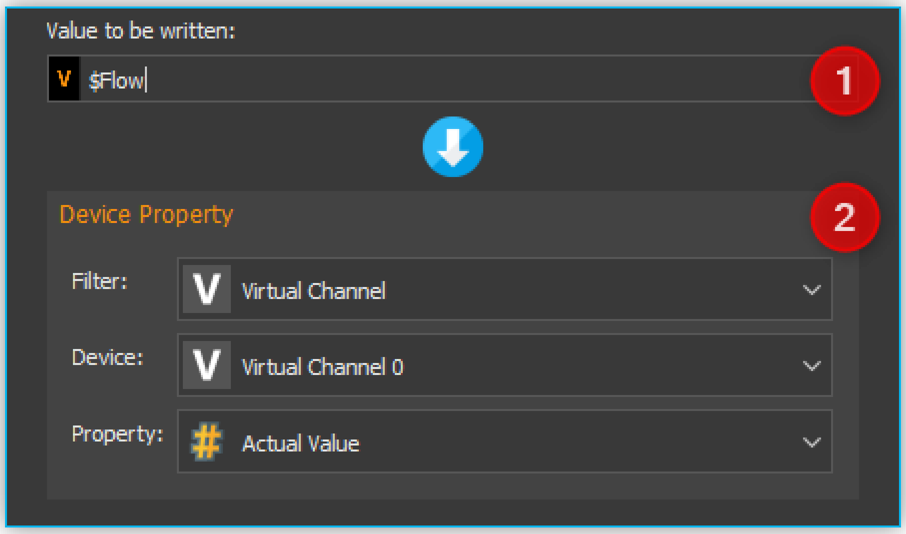
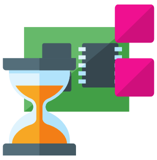


Figure 127: Scriptconfiguration Write Device Property

The value in the input field 1 is then written at run time in the selected device property.

6.15.4 Wait For Device Property



This function allows a script to wait for a specific device property to reach a predefined condition. Once this condition is fulfilled, the script will resume.

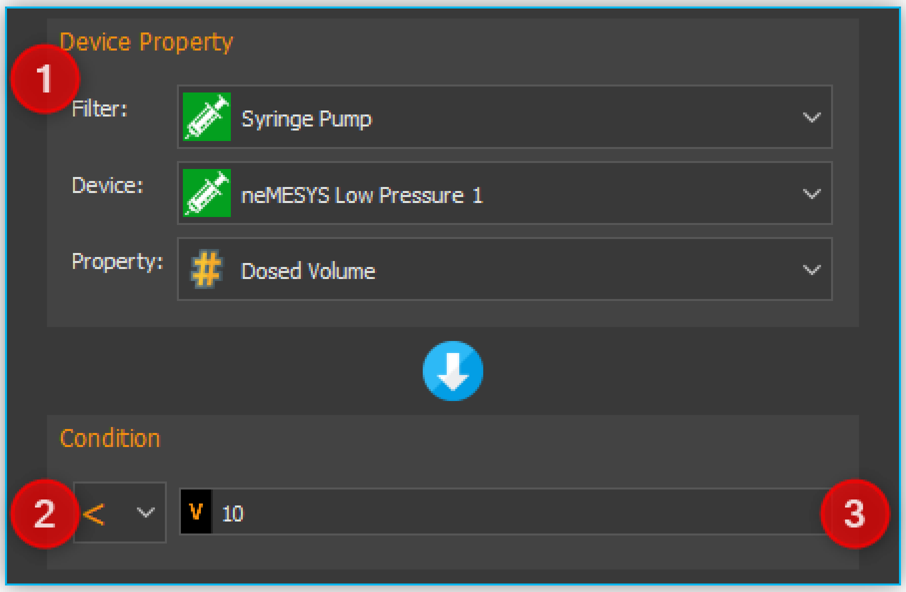


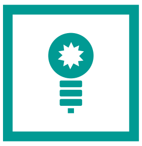
Figure 128: Script configuration Wait for Device Property

In the configuration panel (figure below), select the device and the appropriate device property that you want to monitor 1 as described for function *Read Device Property*.

In the *Condition* input box you can configure the condition that is to be checked against. First define an *operator* 2 and then the *target value* 3 the device parameter is to be compared with. In the value field 3, you can also use variables to set a test condition.

In short, the above set-up procedure reads as follows:

Continue script execution, if the selected property of the device 1 *meets the condition* 2 3.

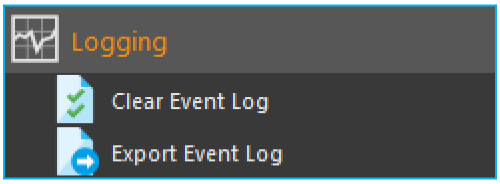


HINT. As a test / comparison condition variables can be used.

1.1 Logging

1.1.1 Overview

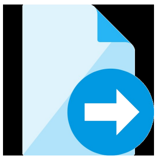
The Logging category contains script functions for controlling various log functions and accessing the event log.



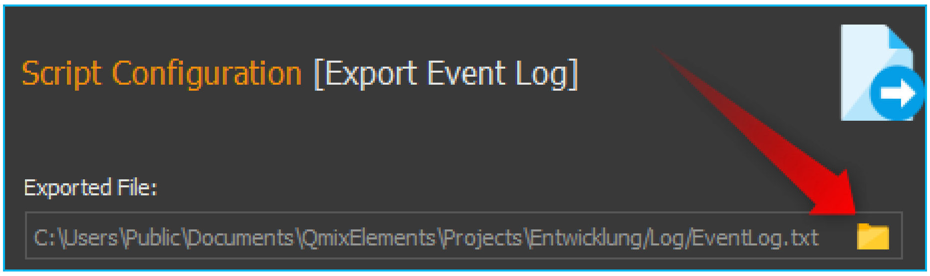
1.1.2 Clear Event Log

This script function deletes all entries from the event log window of the application.

1.1.3 Export Event Log



The export function exports the contents of the Event Log window to a text file. Click on the folder icon in the configuration area (see figure below) to select a file name and location. We recommend that you keep the location in the Log folder of the current project.



The function saves the exported file with a timestamp, e.g. *EventLog_20220121_161653.txt*.

7 Nemesys Plugin

7.1 Introduction

The Nemesys Plugin is for controlling the CETONI Nemesys syringe pumps.

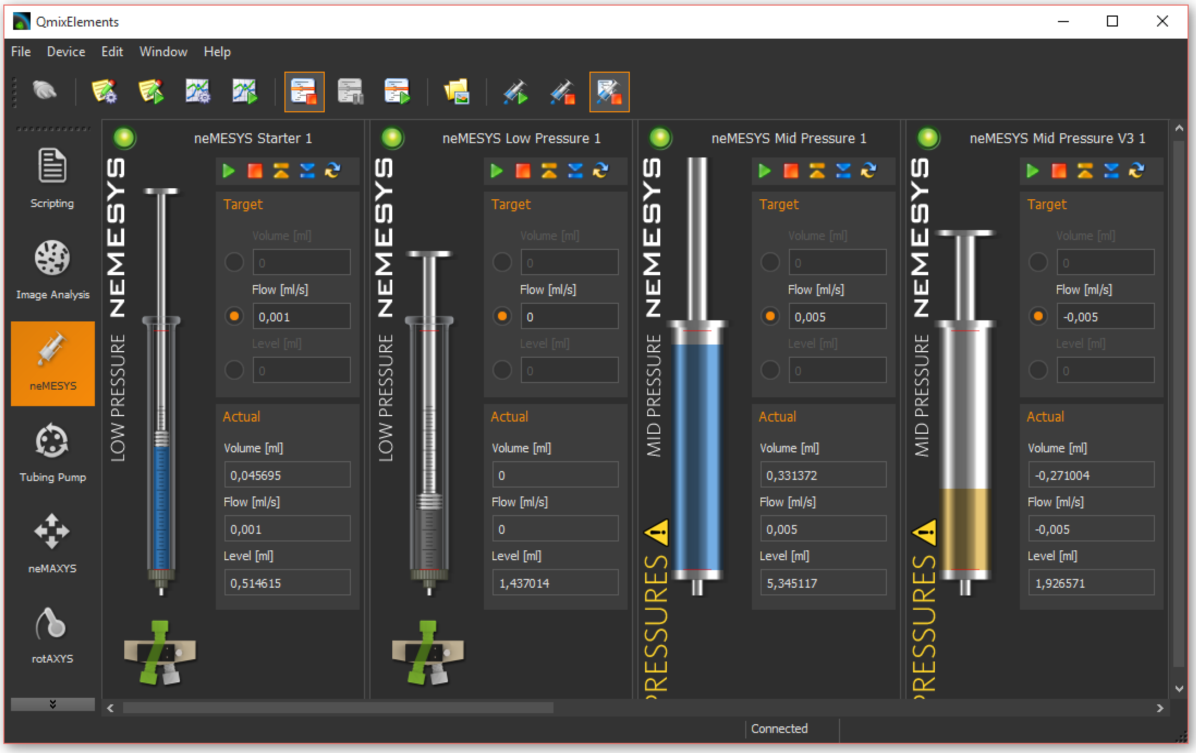


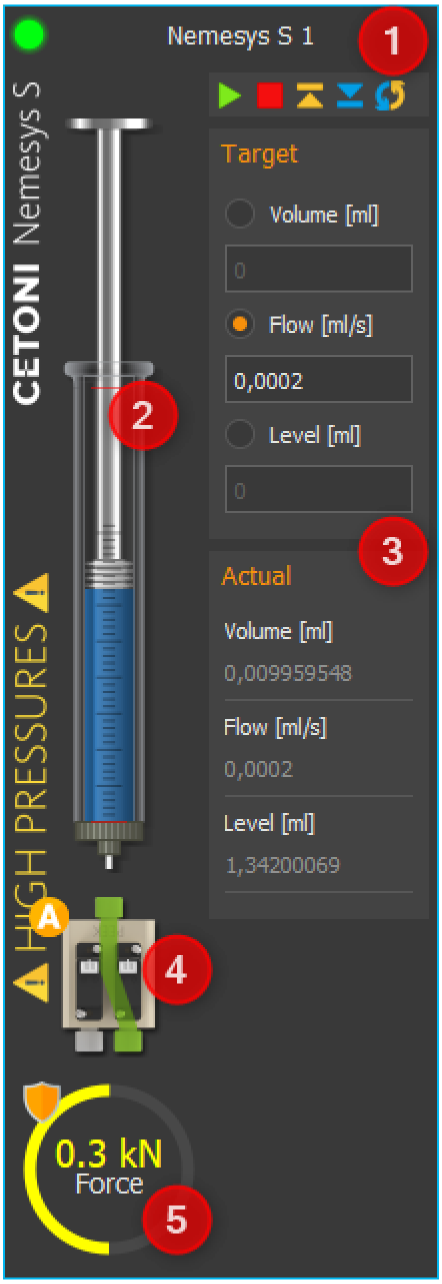
Figure 129: Nemesys workspace

You see operating panels for all the existing, configured syringe pumps displayed in the working area of the Nemesys plugins (see Figure above).

7.2 Operating Panel of Dosing Modules

7.2.1 Overview

Each operating panel is used to control, configure and visualize a Nemesys dosing module. It contains all the control elements for configuring and controlling the pump parameters and for visualizing the current status.



The panel consists of

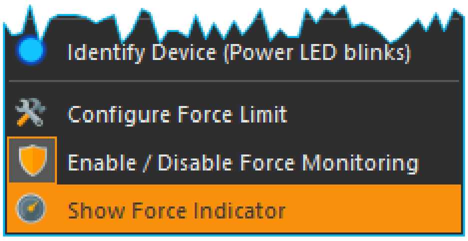
1 Caption of the syringe pump

2 Level indicator

3 Control elements (depending on operating mode)

4 Valve switching and status

5 Display of the internal force sensor if supported by the device (e.g. for Nemesys M and S). The display is hidden by default. To show the display, select *Show Force Indicator* from the context menu.



7.2.2 Changing module names

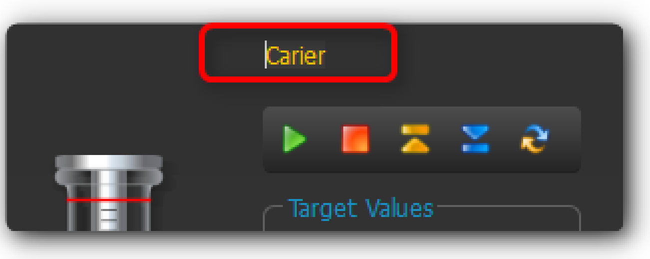
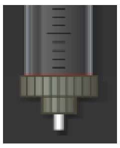


Figure 130: Changing module name

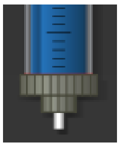
You can assign a unique name to each dosing unit. For example, you can indicate which liquid a dosing unit doses or what it is used for. To assign a new name, simply make a left mouse click on the name of the dosing unit. You can then enter a new name, and confirm the entry with *Return* (Figure below).

7.2.3 Syringe level indicator

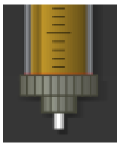
The fill level indicator of the syringe always shows the current position of the drive unit, and thus the current fill level of the syringe used. The colour of the syringe also indicates the current motion of the dosing unit, so you can immediately see the direction of motion of the syringe even with a very low flow rate. The following colours are possible:



Grey - The drive unit is currently inactive.



Blue - The syringe piston is moving in the direction of the syringe clamp (Dispensing).



Orange - The syringe piston is moving away from the syringe clamp (Aspirating).

This color coding of the direction of motion is also used in all other parts of the Nemesys plugin.

HINT. The colours of the syringes help you identify the direction of motion quickly, especially with very low flow rates, because it is almost impossible to see the position of the syringe piston changing.

The thickness of the syringe in the display indicates the approximate size of the currently clamped syringe. You should be very cautious with the use of higher flow rates, especially with very large syringes, because very high pressures can develop very quickly, which could damage the device (valve) or your application.

ATTENTION. Risk of high pressures damaging the connectors in the liquid path. To avoid high pressures causing damage, never make a reference move with a high-pressure syringe clamped.

7.2.3.1 SOFTWARE LIMITS

The travel path of a syringe is limited in the software by a minimum and a maximum value in order to prevent irreparable damage to the syringe or the piston being pulled out of the body of the syringe. These values are indicated by two red marks in the fill level indicator (Figure below), and they have to be defined for each syringe in the syringe configuration (section 7.9).

7.3 Force Monitoring

7.3.1 Introduction

The latest pumps in the Nemesys range (Nemesys M and Nemesys S) have an internal force sensor to monitor the force with which the syringe plunger is pressed into the syringe. This serves to minimise the risk of damage to the device, the syringes and the application.

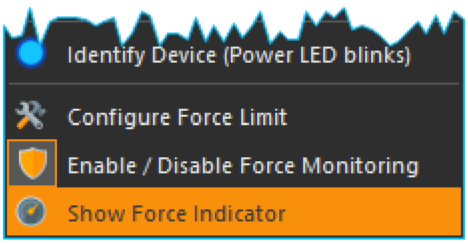
Each pump has a fixed maximum force value at which dosing stops:

- Nemesys S: 0.48 kN
- Nemesys M: 1.3 kN

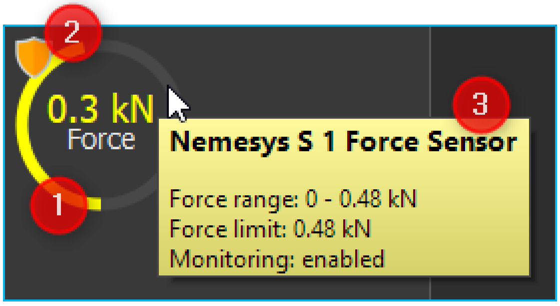
In addition, the maximum force can be further limited by a user adjustable force limit.

7.3.2 Operating Controls

In the operating panel of the pump you will find the display of the force monitoring ❶ below the syringe display: If the display is hidden, you can show the display again via the context menu of the pump. In the context menu, select the menu item *Show Force Indicator*:



The protective shield symbol ❷ (see figure below) in the upper left corner of the force display indicates whether the force monitoring is active or inactive (symbol greyed out).



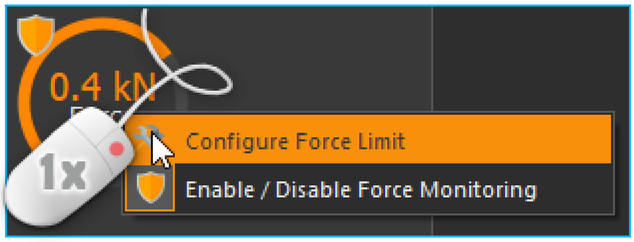
Dosage is only possible when force monitoring is activated. If you move the mouse arrow over the force indicator, additional information is displayed in a small window ❸:

- Name of the force sensor channel (e.g. to query the sensor in the script or to use it in the logger).

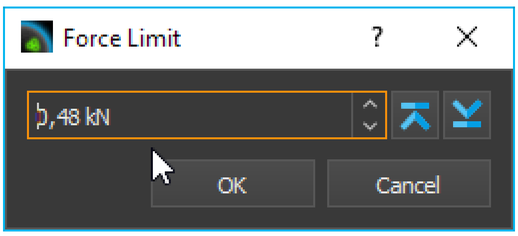
1. Force range - the adjustable range of force monitoring

- Force limit - the currently set maximum permissible force
- Monitoring - indicates whether the force monitoring is active.

To limit the maximum force, right-click in the force display and select the menu item *Configure Force Limit*:

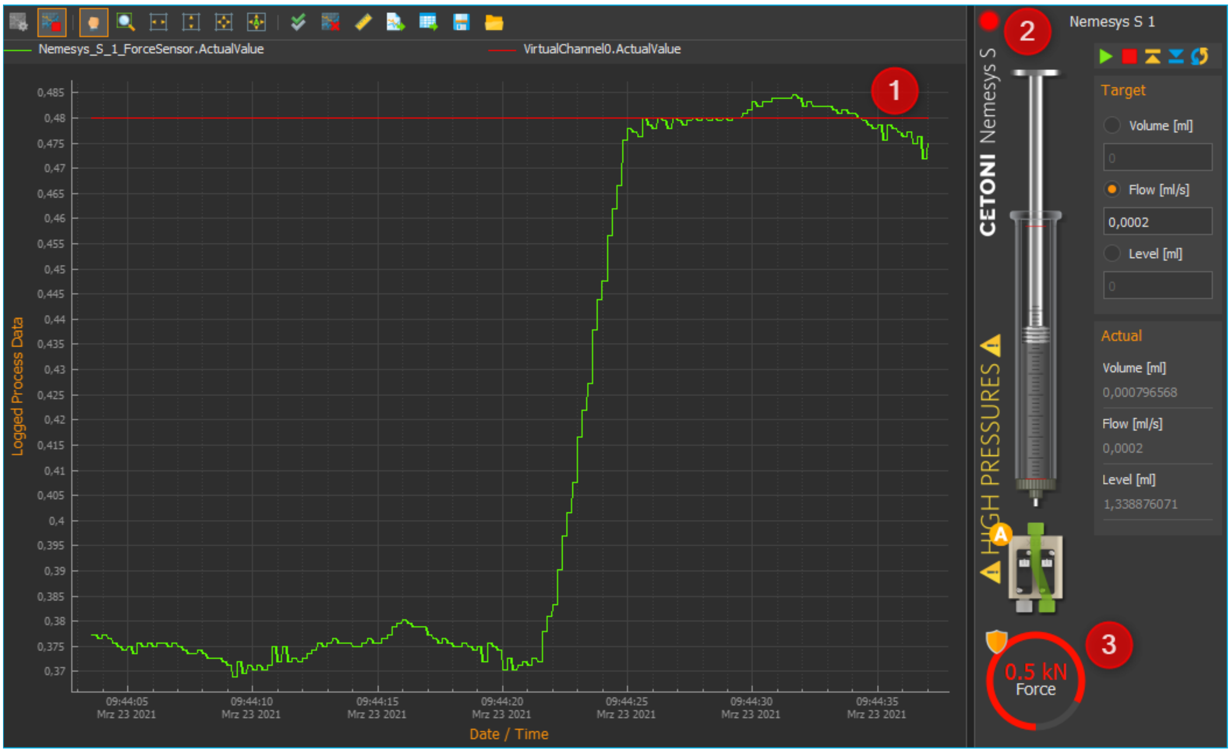


Then enter the desired maximum force and click OK:

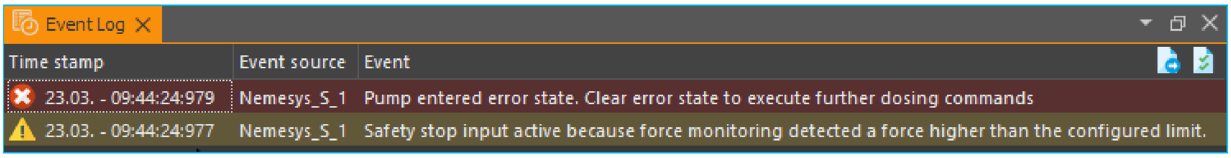


7.3.3 What to do after a force overload stop

If the maximum set force is exceeded while the pump is in use, the pump is stopped immediately and set to a fault state. This is shown in the figure below. The value of the force sensor (green curve) has exceeded the set force limit (red line) ❶.



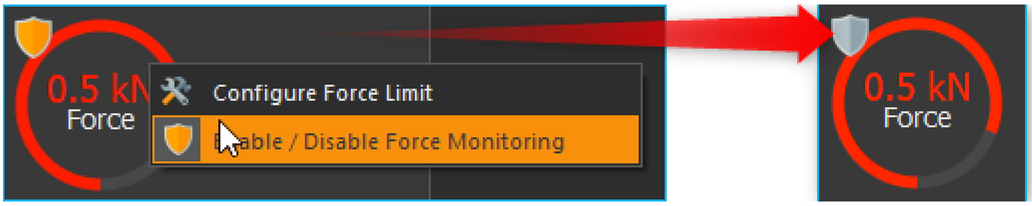
The error status of the pump is indicated by the red flashing LED ❷. The current measured value of the force sensor can be seen in the force display ❸. In addition, the reason for the error is shown in the event log.


Event Log

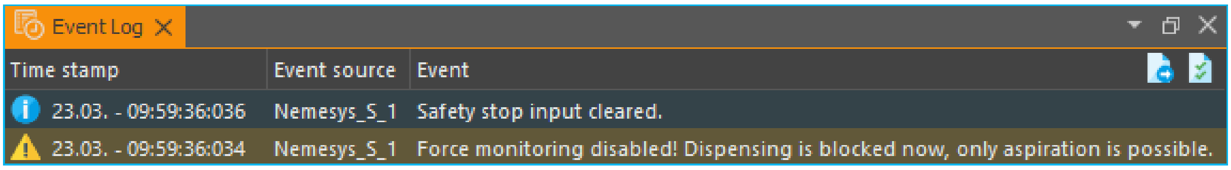
Time stamp	Event source	Event
23.03. - 09:44:24:979	Nemesys_S_1	Pump entered error state. Clear error state to execute further dosing commands
23.03. - 09:44:24:977	Nemesys_S_1	Safety stop input active because force monitoring detected a force higher than the configured limit.

To reset the pump from this "overload" state to normal operating mode, proceed as follows:

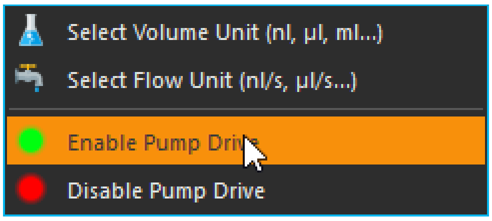
(1) **Disable force monitoring**. Click with the right mouse button in the force display and select the menu item *Enable / Disable Force Monitoring*. The protective shield symbol should then be greyed out.



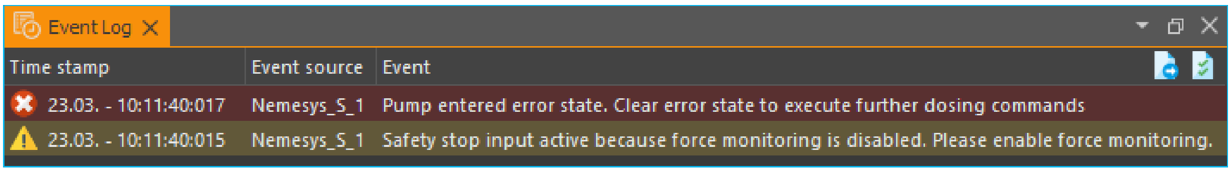
The event log informs you that the safety stop has now been reset and that only aspiration is possible:



(2) **Enable pump**. To activate the pump, right-click on the red status LED of the pump and select the menu item *Enable Pump Drive* from the context menu. The status LED of the pump must then light up green.

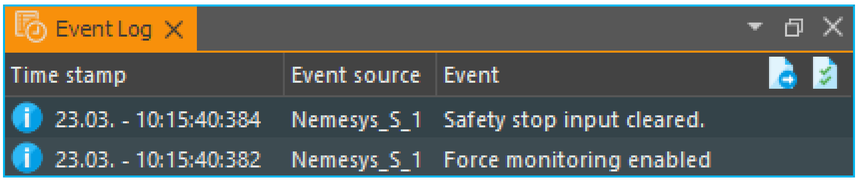


(3) **Reduce force**. You can now reduce the force by drawing up the syringe or starting a dosage with a negative flow rate. You should now see the value in the force display decrease. As soon as the force value falls below a threshold, the pump will stop and return to an error state. You should then see a corresponding message in the event log:

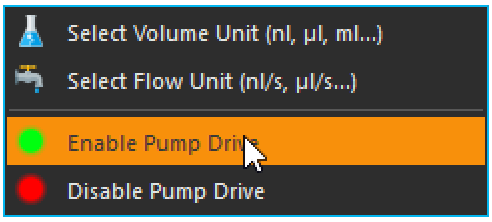


(4) **Activate force monitoring**. Now click with the right mouse button in the force display again and select the menu item *Enable / Disable Force Monitoring*. The force monitoring should then be active again and the shield symbol should be coloured:

In the event log you should now see the information that the force monitoring is active again and that the safety stop has been reset.



(5) **Activate pump.** To activate the pump, right-click on the red status LED of the pump again and select the menu item Enable Pump Drive from the context menu. The status LED of the pump must then light up green.



Now the pump is ready for operation and can be used as usual.

7.4 Carrying out Reference move

HINT. With the current Nemesys M and Nemesys S pumps, a reference move is no longer necessary and you can skip this chapter.

The drives of the dosing units are monitored and controlled by a digital positioning unit. You can make a reference move of the dosing unit to calibrate this positioning unit. A suitable time for this is, for example, when a syringe is changed, because the syringe has to be removed from the dosing unit in order to make a reference move. During the reference move, the dosing unit travels to its lower limit position, and calibrates your zero position when it reaches the limit position.

IMPORTANT. Always make a reference move if you operate the pumps from another PC, because the calibration data are stored on the PC and not in the dosing units.

To start a reference move, make a right mouse click in the operating panel of the dosing unit that is to be calibrated. Select the *Reference Move* menu item in the context menu that will open now.

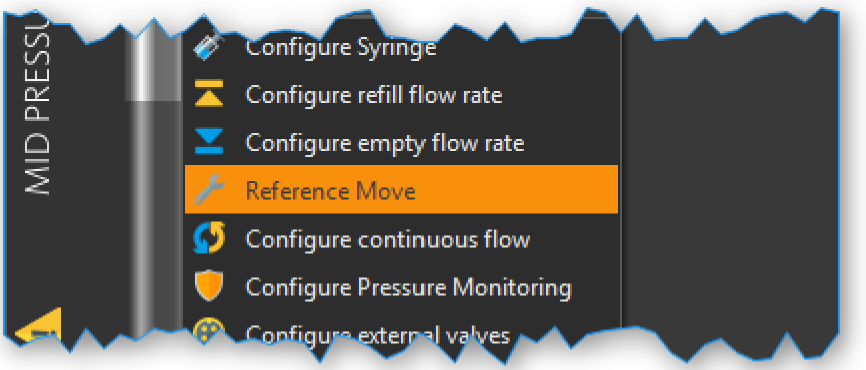


Figure 131: Start reference move

ATTENTION. Danger of damaging syringes! The system may only be calibrated when a syringe is not installed in the dosing unit.

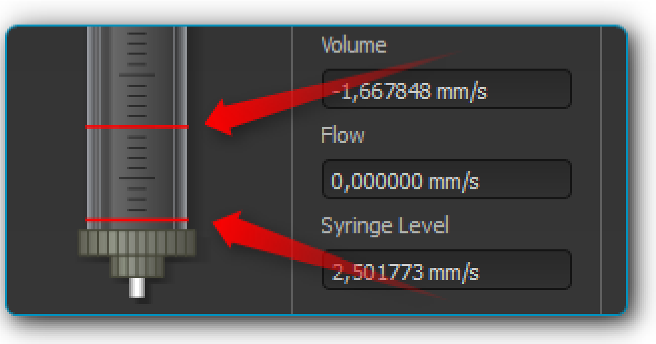


Figure 132: Software limits

ATTENTION. Risk of high pressures damaging the connectors in the liquid path. To avoid high pressures causing damage, never make a reference move with a high-pressure syringe clamped.

7.5 Valve

If your dosing unit has a valve, you can switch the valve or configure the automatic valve switching in this part of the operating panel. The current switching status of the valve is indicated by the valve control element.

HINT. Move the mouse cursor over the valve to get additional information about the valve.

7.5.1 Manual valve switching

To switch the valve, simply make a left mouse click on the valve icon (figure below). You should hear a soft click, and the valve image should switch to the other switching status.

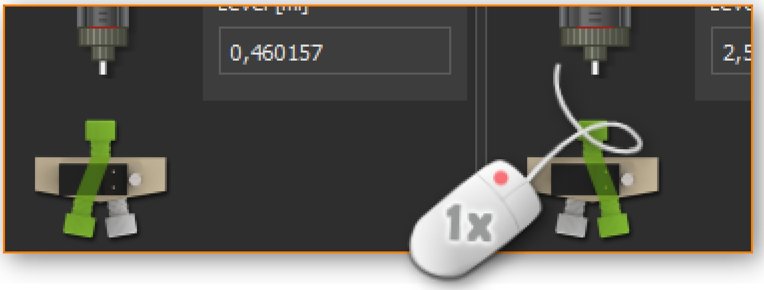


Figure 133: Manual valve switching

If you have a valve with more than two valve positions, you can cycle through all positions by clicking with the left mouse button on the valve image. If you hold down the *Shift* key when you click on the valve, you can cycle through the valve positions in the opposite direction:

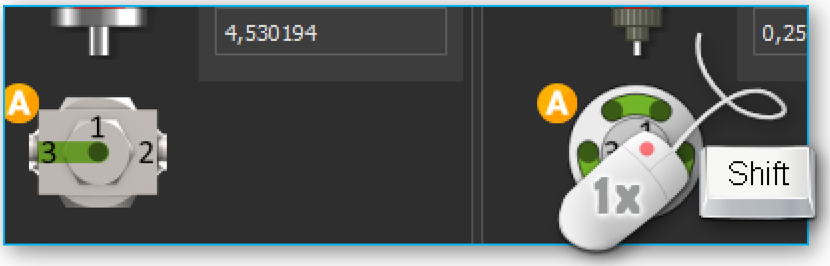


Figure 134: Inverted manual valve switching

Alternatively, you can select the desired valve position directly. Click with the right mouse button on the valve, and then select the context menu item *Switch Valve*.

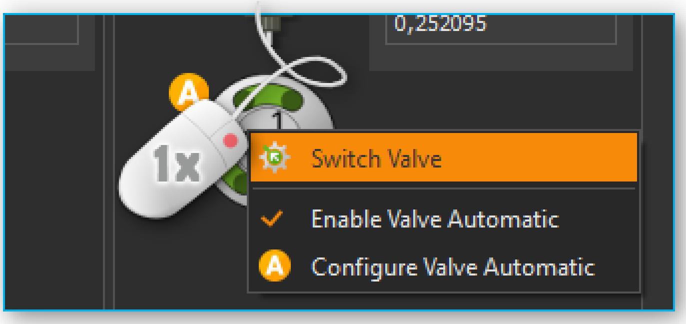


Figure 135: Showing valve context menu

A small dialog window comes up, where you can choose the desired valve switching position from a valve selection box. Once you select a position, the valve is switched immediately. You can finish the selection of a valve switching position by clicking the green check mark (see figure below).

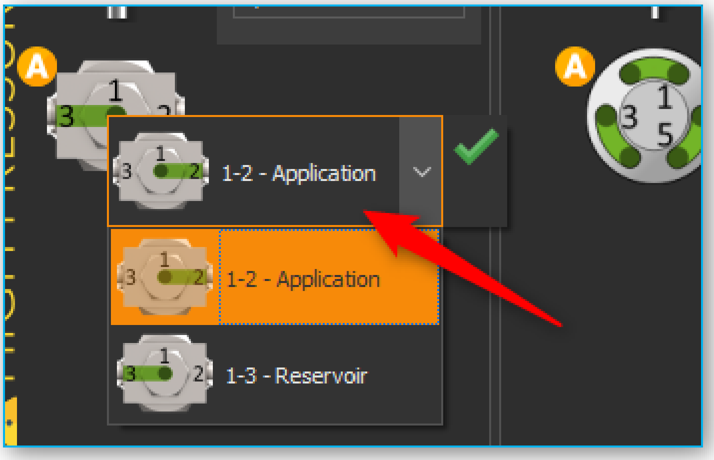


Figure 136: Selection of valve switching position

7.5.2 Automatic valve switching

Each pump has a valve automatic, which automatically switches the associated valve when the pump state changes. The small orange *A-icon* overlay shows you whether the valve automatic mode is active.

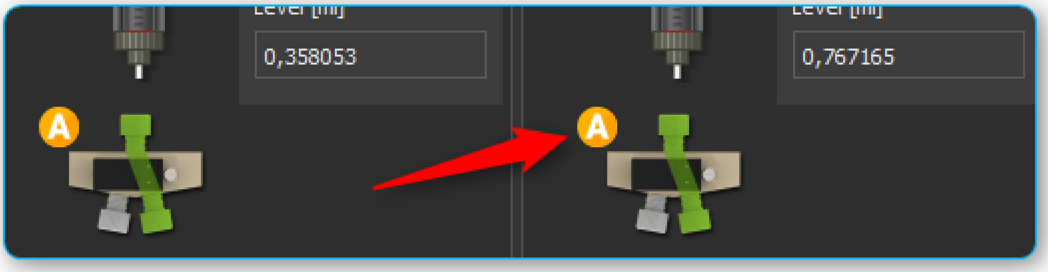


Figure 137: Valve automatic mode active

To toggle the valve automatic, just click with the right mouse button on a valve and then select the menu item *Enable Valve Automatic*.

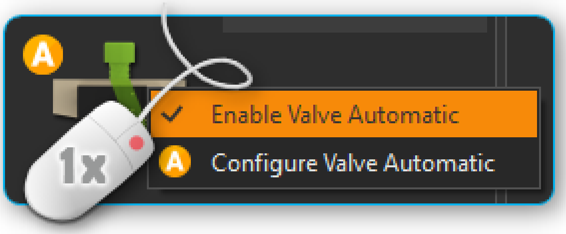


Figure 138: Enable / disable valve automatic

To configure the automatic valve mode, click with the right mouse button on a valve and select the menu item *Configure Valve Automatic*. The valve automatic configuration dialog will then appear.

Figure 139: Valve automatic configuration dialog

You can configure the target valve position for each pump state (*Aspiration, Stopped and Dispensing*) separately. To activate valve switching for a certain state, check the check box 1. It is then displayed a selection box 2 for selection of the valve position. If you want to disable valve switching for a certain state, simply uncheck the check box 3. Finish the configuration by clicking the OK 4 button.

IMPORTANT. In certain dosing modes, such as during continuous dosing with two pumps, the valve automatic is temporarily disabled because the valves are switched by the logic of the respective operating mode.

7.5.3 Assigning a Valve

You can assign a valve to each syringe pump, regardless of whether the pump has its own valve or not. This valve can be e.g. a Qmix V module, an external ball valve or the valve of another syringe pump. You can do this via drag & drop. Simply drag the corresponding valve out of the valve list 1 onto the front panel 2 of the corresponding pump.

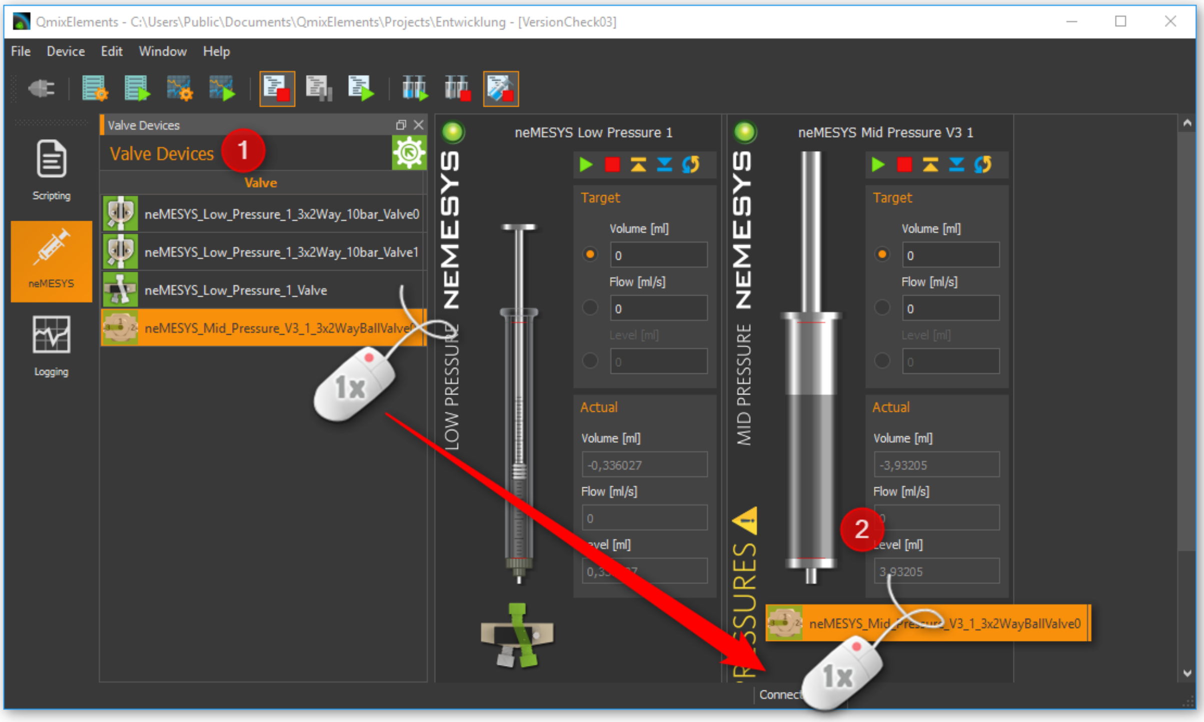


Figure 140: Valve Assignment via Drag & Drop

To assign a valve, select the menu item *Assign Valve* from the context menu of a pump.

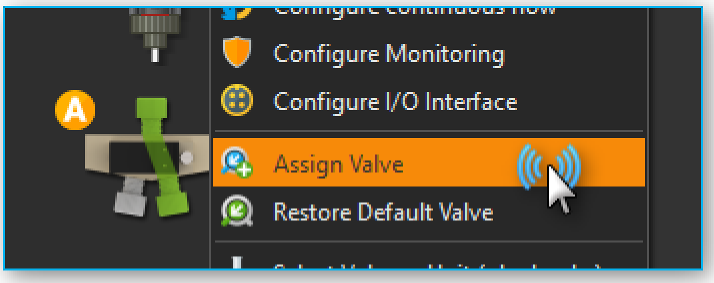


Figure 141: Valve Assignment

Then select the valve from the valve list, which you want to assign.

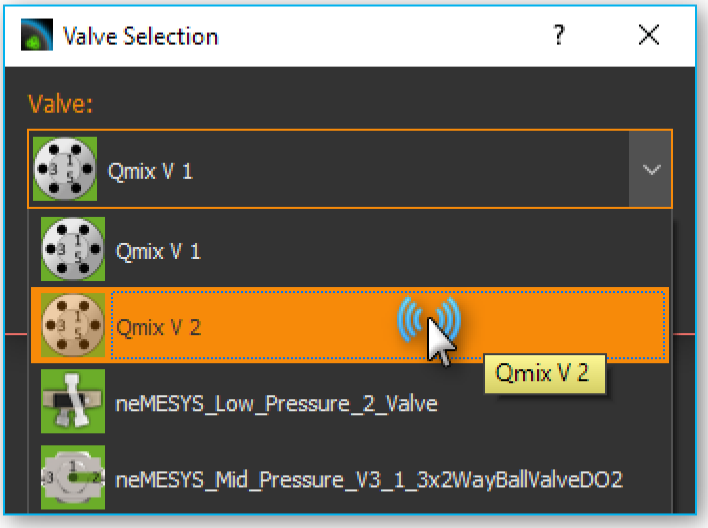


Figure 142: Valve selection dialog

The new valve is then displayed below the syringe. When you configure and activate the valve automatic for this valve, the valve is automatically switched when aspirating and dispensing liquid and when stopping the pump.

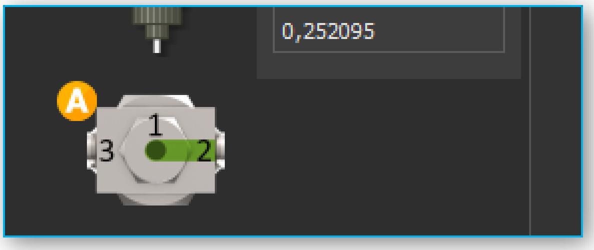


Figure 143: External 3x2 way ball valve

IMPORTANT. You can only assign one single valve to a certain pump.

If you want to restore the original valve configuration later, select the context menu item *Restore Default Valve*.

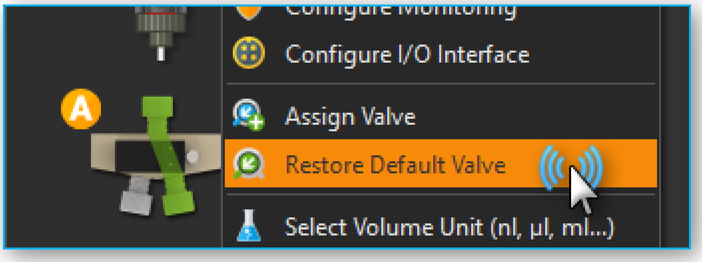


Figure 144: Restore default valve configuration

HINT. In section Connect external valves to I/O interface you will find a detailed guide how to integrate external valve modules connected to the I/O interface (such as external ball valves) in the software.

7.6 I/O Interface

7.6.1 Overview

Various Nemesys pumps offer an I/O interface for connecting external sensors and signals, or for outputting trigger signals (example shown below).

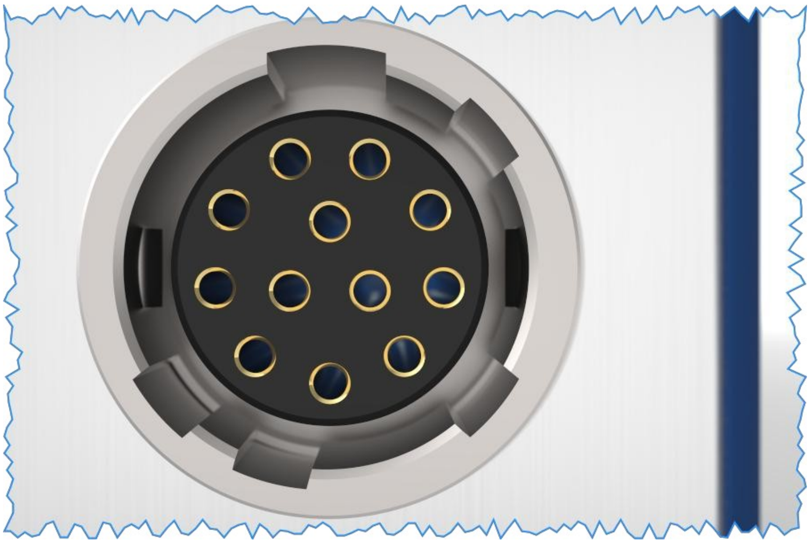


Figure 145: I/O interface Nemesys mid pressure pump V3

Depending on the device configuration the I/O interface features digital inputs and outputs and/or analog inputs. All I/O channels can be found in the software window *I/O Channels*. If this window is not visible, you can show it by clicking the menu item *Window* → *I/O Channels* in the main menu (see figure below).

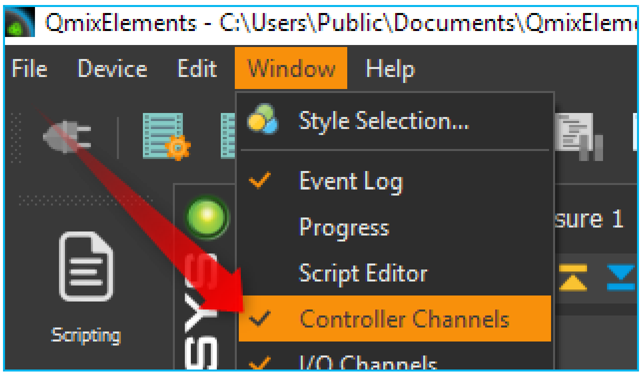


Figure 146: Showing I/O channel window

Each I/O channel has a unique name (e.g. *Nemesys 1 Analog In 1*), which is composed of the pump name (e.g. *Nemesys 1*) and the name of the channel (e.g. *Analog In 1*) like in the figure below.

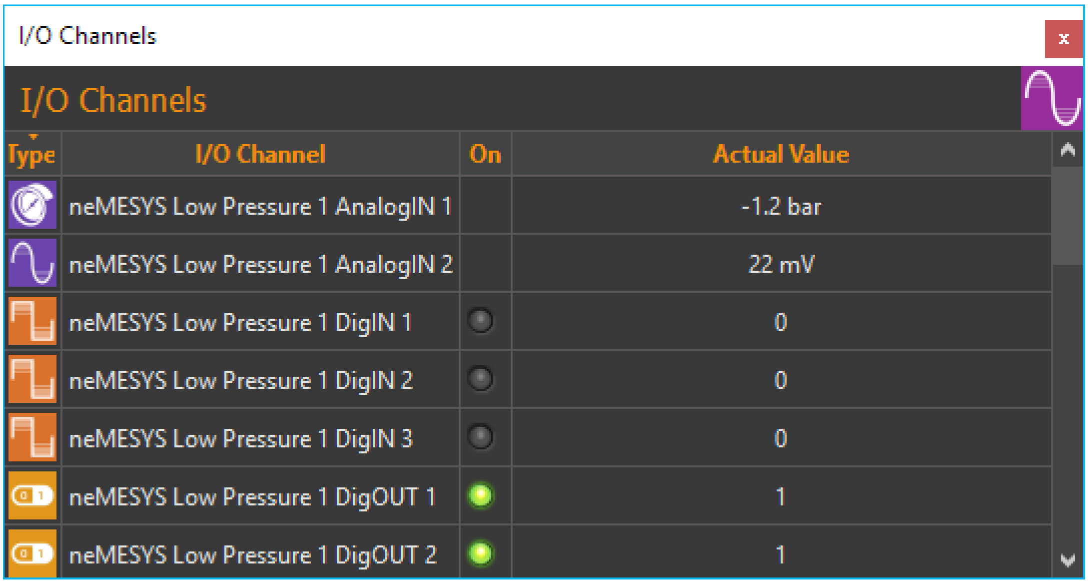


Figure 147: Nemesys I/O channels

7.6.2 Scaling of the analog inputs

The analog inputs of the modules measure the input voltage in the range of 0 - 5000 mV. For each channel, you can specify a separate scaling. This allows you for example to scale the voltage value of 0 - 5000 mV into a pressure value of 0 – 20 bar, if you have connected a pressure sensor. For details on configuring the channels or on scaling, read the documentation of the Qmix I/O plugin.

In the following example, we will show you, how to connect a pressure sensor with the measuring range: 0.5 V - 4.5 V corresponds to 0 - 20 bar. The sensor is connected to analog input 1. We will now define a scaling for this pressure sensor. To do this, click with the right mouse button in the channel row and select the menu item *Configure channel*. The following configuration dialog is then displayed.

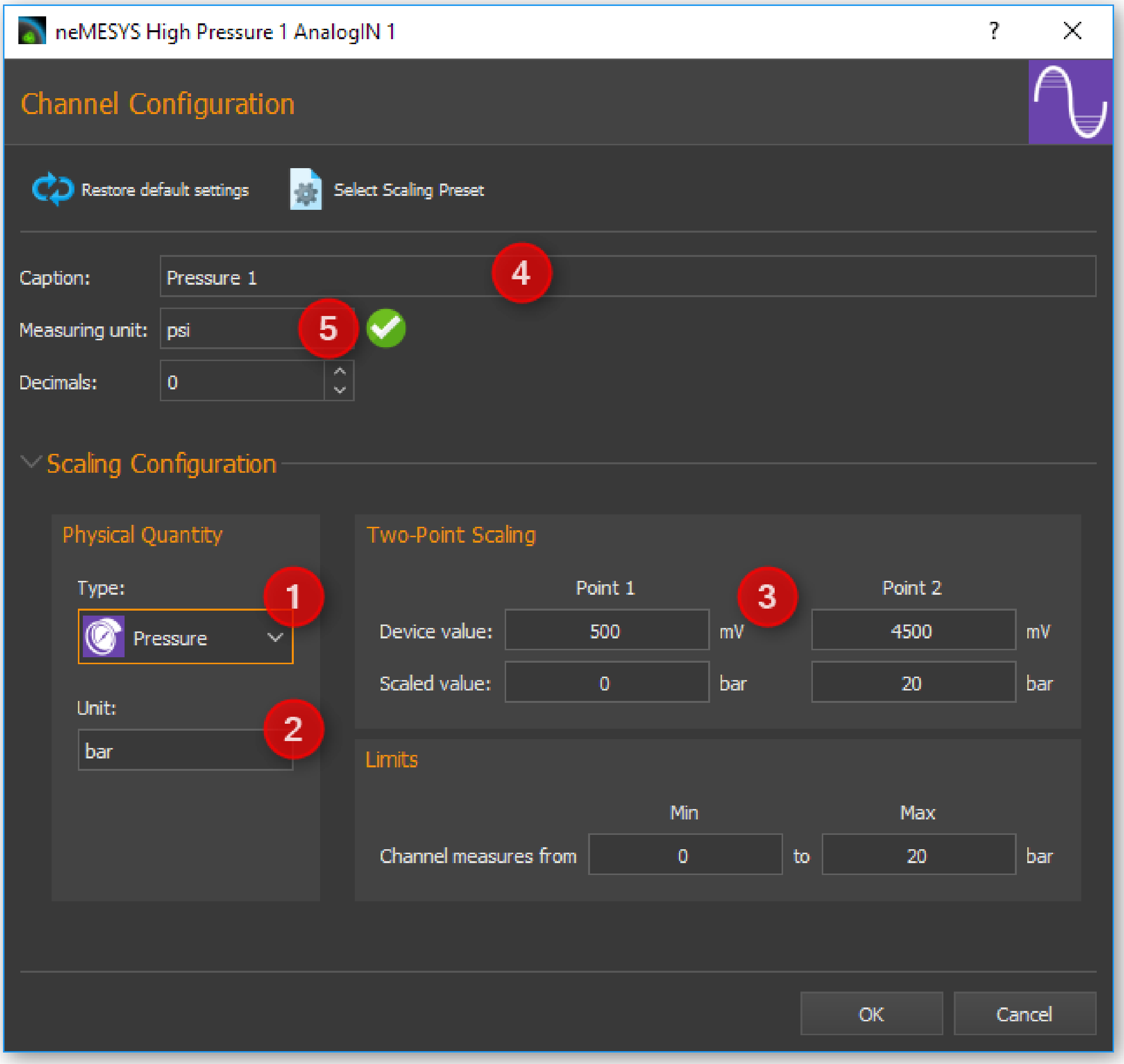


Figure 148: Example: input scaling for pressure sensor

In this dialog, we configure the following values:

- 1 as sensor type we choose *Pressure*
- 2 the basic unit bar corresponds to the unit in the data sheet of our sensor
- 3 in the line *Device value*, enter the measuring range of the analog input: 500 mV to 4500 mV. In the *Scaled value* line, enter the measuring range of the sensor 0 - 20 bar.
- 4 we select a meaningful name for the channel: *Presssure 1*
- 5 As the display unit, we select the pressure unit psi in the measuring unit input field.
- by clicking OK we complete the configuration

The measured pressure of the sensor is now displayed in the I/O Channel window in bar (see figure below).

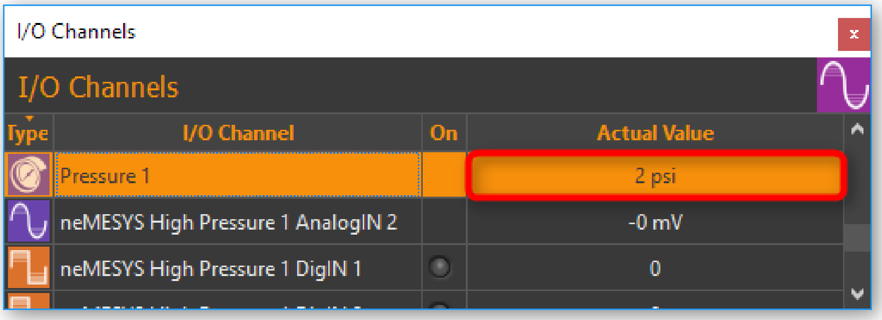


Figure 149: scaled measurement for pressure sensor

HINT. For a detailed description of I/O channels, configuration and scaling, see the [Qmix I/O plugin documentation](#).

7.6.3 Connect external valves to I/O interface

You can connect various external devices, such as external pressure sensors or external valves, to the Nemesys I/O interface connector. To configure an external valve choose the context menu item *Configure external valves*.

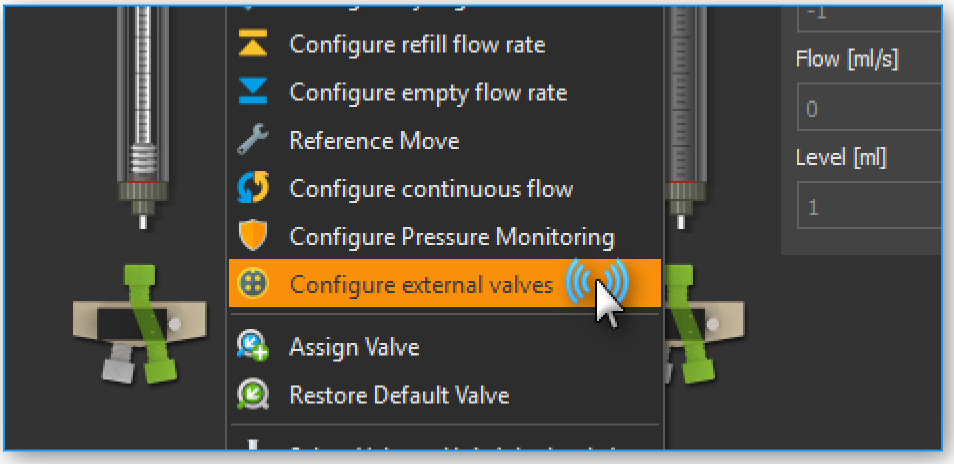


Figure 150: External valve configuration

From the list of external valves, you can now select a device you want to connect to the I/O interface.

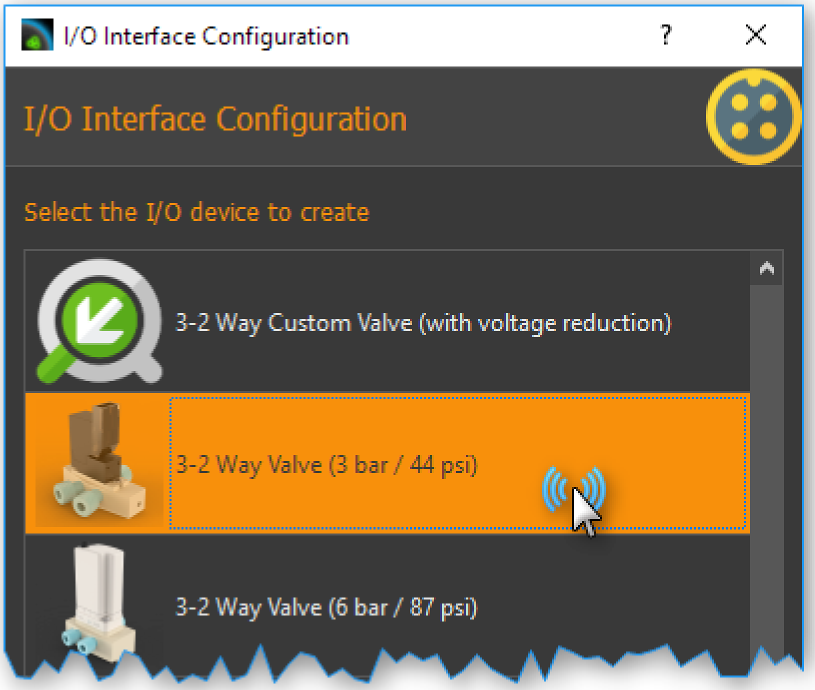


Figure 151: Selection dialog for external I/O devices

If you have selected a device, confirm your selection with *OK*.

After selecting a valve, a dialog appears where you can configure whether the valve is to be a primary valve or secondary valve.

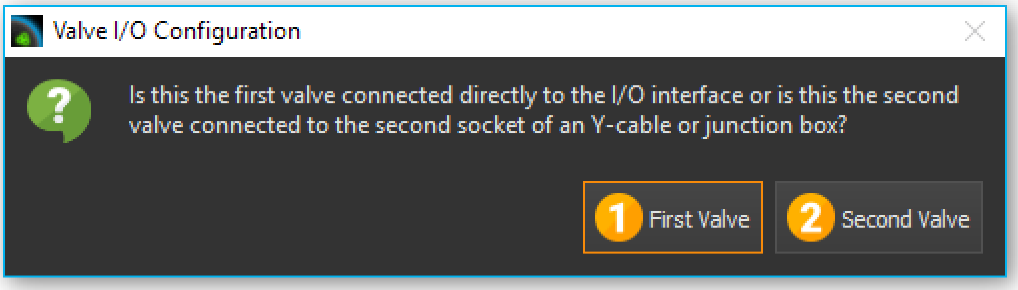


Figure 152: Primary / Secondary valve selection

The *primary valve* can be connected directly to the I/O interface. If you want to connect a second additional valve, you need a Y-cable or a distribution box. Since the second valve is connected to a different digital output, you need to choose 2 *Second Valve*.

If you added a valve, the new valve will appear at the end of the valve list (see figure below). Now you can switch the valve positions or assign it to a pump.

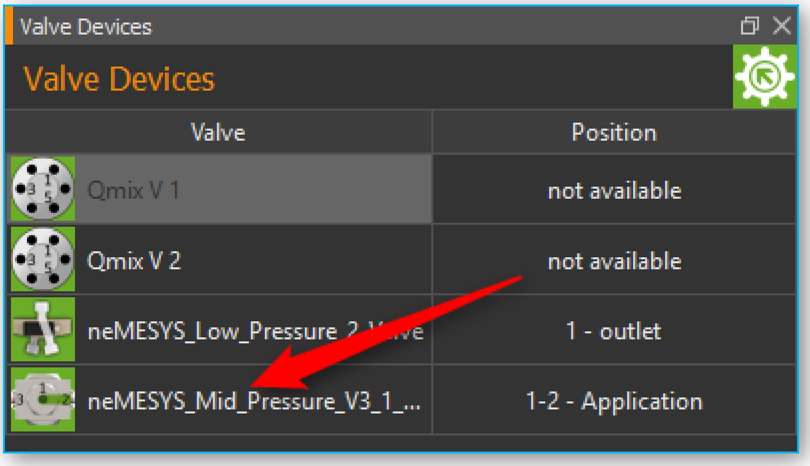


Figure 153: Valve list with external valve as last list entry

Later, if you no longer need a certain valve, you can delete it in the valve list. Click with the right mouse button on the valve list entry to open the context menu and select the menu item *Delete Valve*.

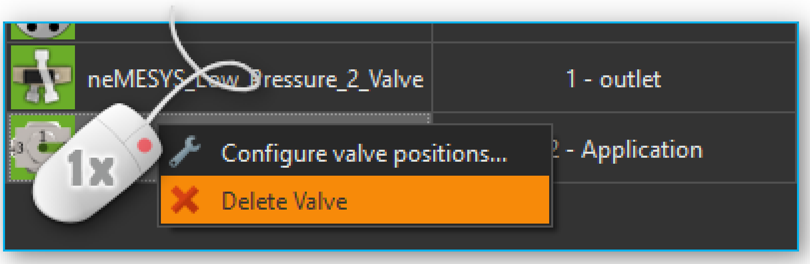


Figure 154: Deleting external valve

The valve is then removed from the list and if you have assigned it to a pump, then it will be removed from the control panel of the pump.

7.7 Pressure Monitoring

Dosing with neMESYS pumps can create high pressure levels, which may cause damage to syringes, the fluidic system or the application. To prevent this from happening, you should monitor the pressure in your system, manually or automatically, using pressure sensors.

7.7.1 Assigning Pressure Sensors

You can assign a pressure sensor to each pump. This sensor will then be displayed on the pump's front panel and can be used for automatic pressure monitoring. The pressure sensor does not have to be electrically connected to the pump. You can use any sensor displayed in the I/O channel list (see image below).

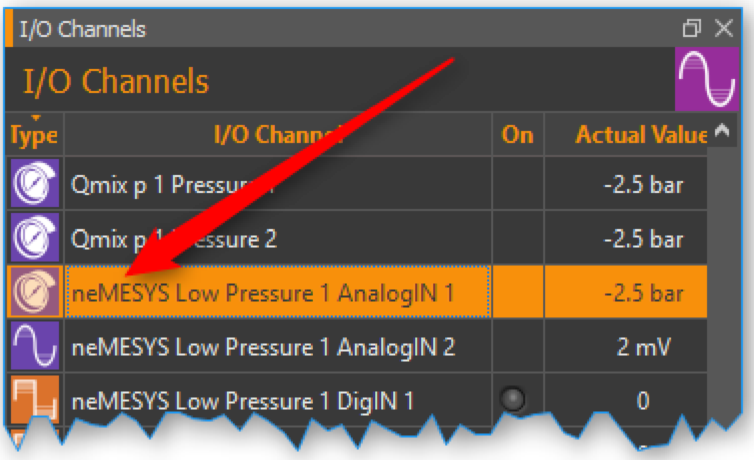


Figure 6: Pressure Sensors in I/O Channel List

TIP: To use any analog input as a pressure measuring input, you have to set it up beforehand. Please refer to the chapter on the Qmix I/O plugin for more on the configuration of I/O channels.

To assign a pressure sensor to a pump, drag the pressure sensor from the I/O channel list and drop it on the pump's front panel.

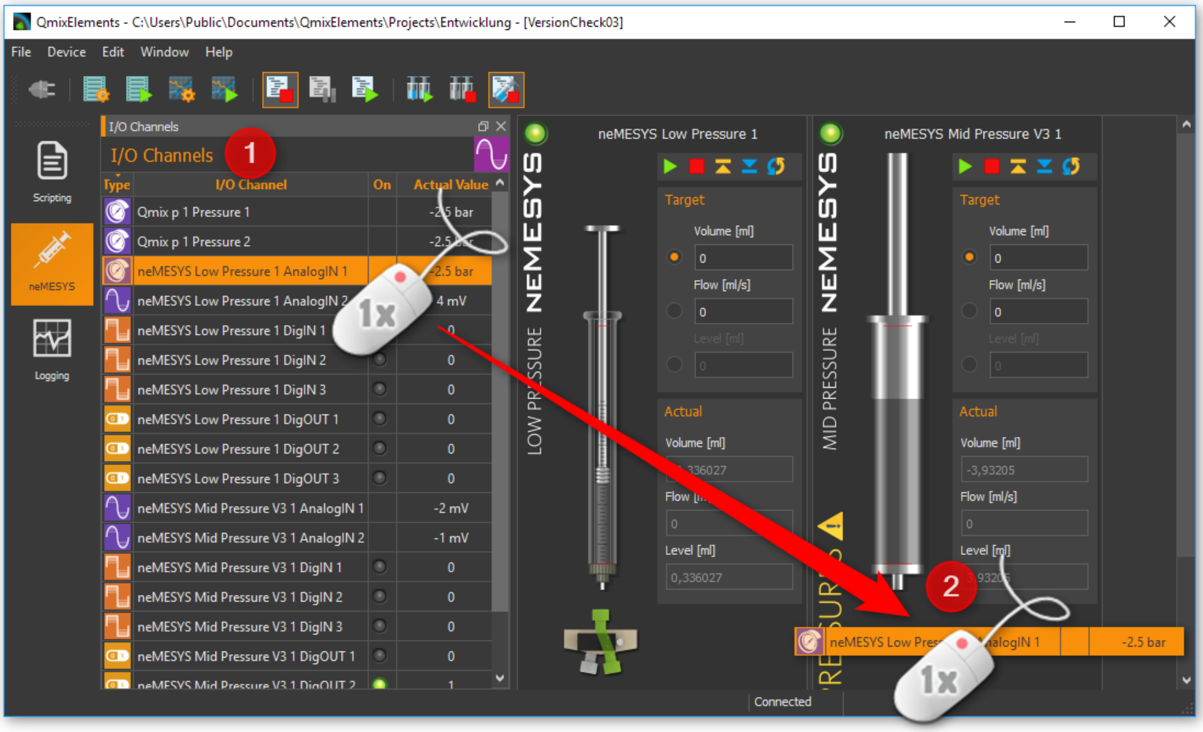


Figure 7: Assigning a Pressure Sensor by Drag and Drop

When you release the mouse button, the pressure sensor is added to the pump's operating panel. The pressure sensor is now displayed in the form of a digital manometer below the syringe, or below the valve, if a valve is present (see figure below).

Figure 8: Pressure Display on the Front Panel

The display changes color depending on the pressure being measured. This gives you a quick indication of how close the pressure level is to the pressure limit.

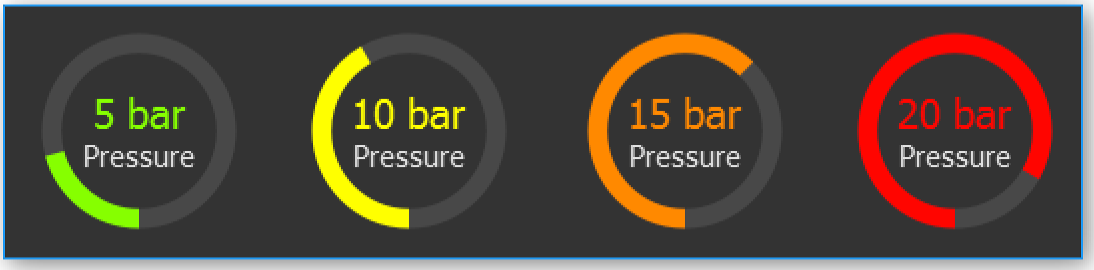


Figure 9: Pressure Sensor Display at Different Pressure Levels

IMPORTANT: You can only assign one pressure sensor per pump.

If you want to restore the original sensor of a pump, select *Restore Default Pressure Sensor* from the context menu on the pressure display.

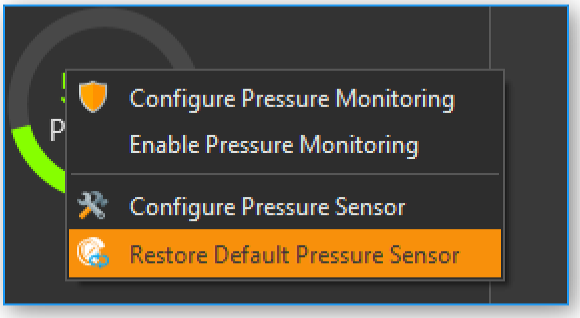


Figure 10: Restoring a Pump's Default Pressure Sensor

7.7.2 Configuring the Pressure Monitoring Function

Once you've assigned a pressure sensor to a pump, you can set up the pressure monitoring function for that pump. When the function is active, it will continuously monitor the current system pressure and trigger certain responses, if the pressure falls below or rises above certain limits. This could include a pump stop, for example.

Please follow these steps to set up the pressure monitoring function:

(1) Perform a right mouse click on the pressure display and select *Configure Pressure Monitoring* from the context menu (see image below).

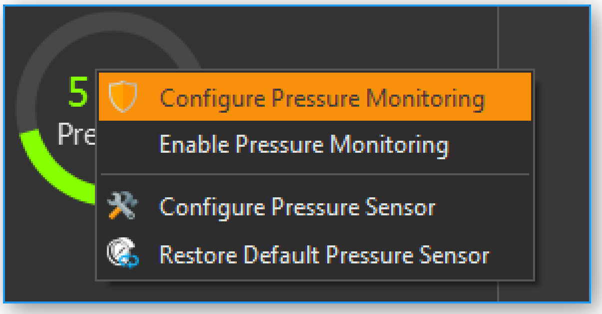
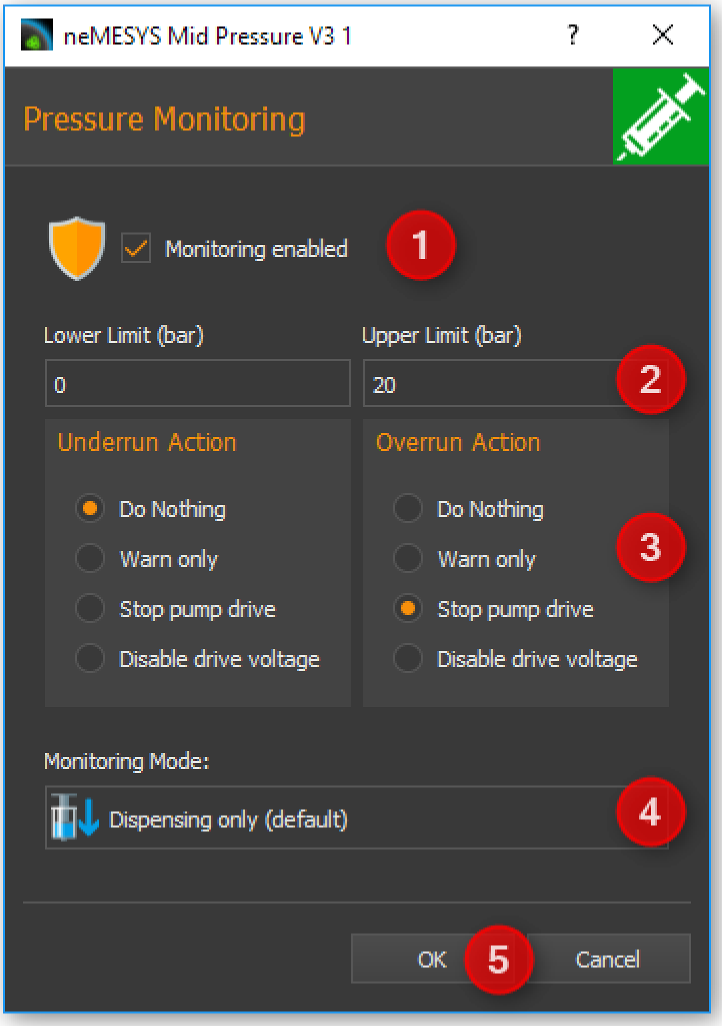


Figure 11: Selecting the Pressure Monitoring Configuration

(2) The dialog window for pressure sensor configuration will be displayed (see image below).



(3) Define a safe range by entering a lower limit and an upper limit. 2. The upper limit is limited by the maximum permissible system pressure. This maximum pressure level is dictated by the maximum rated pressure of the syringe, the valve, the pressure sensor and the maximum force provided by the drive unit.

(4) By selecting the respective button, you can determine what you want to happen when the lower threshold is underrun (u*nderrun action*) 3 or the upper threshold is exceeded (ov*errun action*).

(5) Please select the pumping direction you want the monitoring function to apply to. You can monitor filling, dosing or both. The recommended standard setting is dosing. In this setting the pump gets stopped in case of excessive pressure, while letting you ease the pressure by switching to filling.

(6) Make sure the *Monitoring enabled* 1 checkbox is checked. Otherwise the pressure monitoring function will not be active.

(7) Accept the configuration by pressing *OK* 5.

In the *Overrun* and *Underrun Action* 3 menu, you have the following options:

- *Do Nothing* – safe range departure is ignored. The neMESYS module continues dosing. There is no information or warning of any kind.
- *Warn Only* – a warning message is displayed upon safe range departure. The neMESYS module continues dosing.
- *Stop Pump Drive* – dosing is stopped upon safe range departure. In addition, a warning message is displayed.
- *Disable Voltage* – the dosing module is stopped by cutting off the motor's electrical power. In this case the syringe piston may be moved by the pressure in the system. In addition, a warning message is displayed. The pump must be reactivated to be able to use it again (see section 7.14. Reactivating a Deactivated Pump).

7.7.3 Using the Pressure Monitoring Function

High pressures can quickly occur during dosing, which can damage the valve or the pressure sensor. For this reason, it is particularly important to monitor the overpressure in the dosing direction. Whether the pressure monitoring is active is indicated by the small shield symbol which is shown in the pressure display when pressure monitoring is activated (see figures below). Only if an action has been configured for the overpressure that causes the pump to stop, the shield symbol is shown in orange. If an action has been selected for overpressure monitoring that only issues a warning, the label is grayed out.

Pressure monitoring disabled

Pressure monitoring enabled - Pump is not stopped in case of overpressure

Pressure monitoring enabled - Pump is stopped in case of overpressure

The arrows in the label indicate the pump direction for which pressure monitoring is active:

Filling monitoring

Dosing monitoring

Filling and dosing monitoring

TIP: Once the pressure monitoring function has been configured, you can activate it or deactivate it an any time using the context menu of the pressure display (see image below).

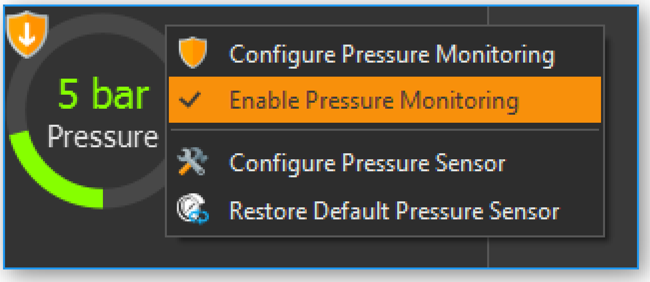
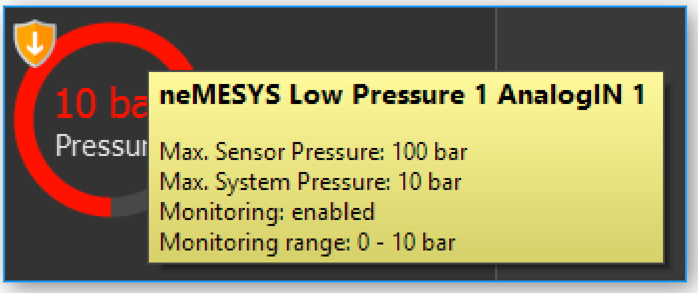


Figure 12: Activating and Deactivating the Pressure Monitoring Function

IMPORTANT: If you remove an assigned pressure sensor, for example by restoring a default pressure sensor, the pressure monitoring function will be deactivated automatically.

If you move the mouse pointer on top of the pressure display, you will get additional information regarding the pressure monitoring function as well as the maximum pressure of the sensor and the system:



7.8 Reactivating a Deactivated Pump

If a neMESYS module has been deactivated by a monitoring function, the color of the LED in the pump's control panel will change from green ❶ to red ❷.

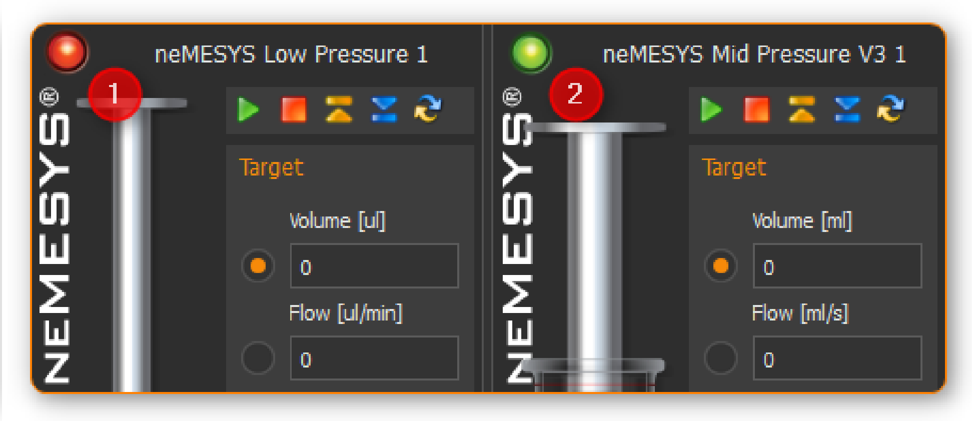
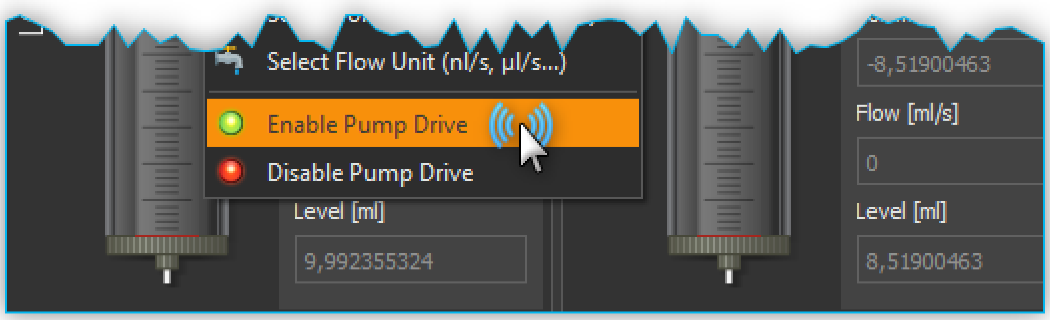


Figure 13: Pump Status Display

To be able to use the pump again, you have to reactivate it. Perform a right mouse click on the pump's operating panel and select *Enable Pump Drive* from the context menu (see image below).



7.9 Syringe configuration

7.9.1 Introduction

The software provides user-friendly ways of configuring and managing your own syringes. The configuration and use of specific syringes in the software thus fulfil two requirements:

1. The software calculates all flow rates and volumes on the basis of the configured syringe.
2. The motion of the piston is limited by the configured syringe length and the set limits.

7.9.2 Opening the configuration dialog

The software constantly shows the current flow rate for each dosing unit. The program needs the parameters for each syringe in order to calculate the flow rates. You should configure these values properly each time a syringe is changed.

To do this, make a right mouse click on the axis of the image of the syringe you want to configure. Select the *Configure Syringe* menu item in the appearing context menu. Then select the syringe you want to configure, as shown in Figure below. The *Configure syringe* dialog then opens.

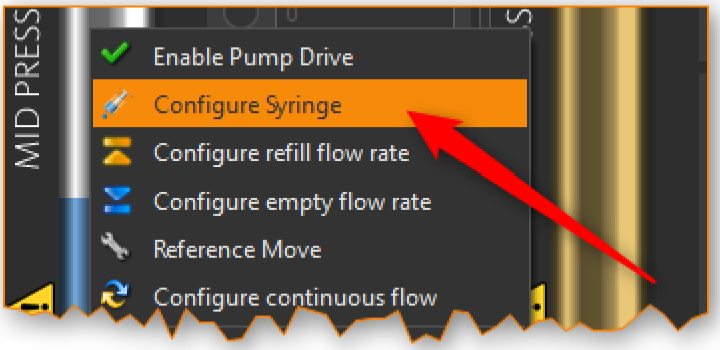
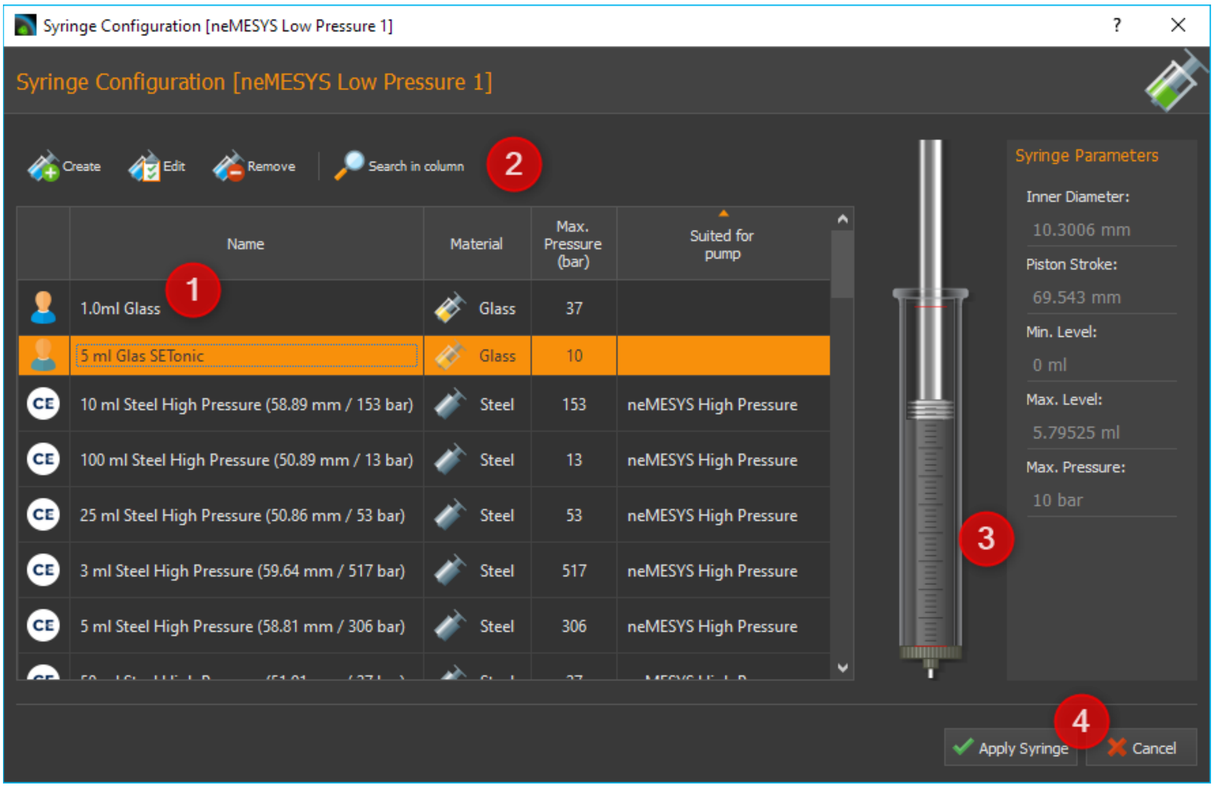


Figure 155: Starting syringe configuration

IMPORTANT. Please note that when a syringe is changed, the fill flow rate and empty flow rate of this dosing unit are reset to 0. This means that you have to reconfigure these values after each syringe change.

7.9.3 Syringe selection dialog

In this dialog, you can also select the desired syringe from a list of available syringes. The following options are available here:



The list view 1 displays all the syringes that you have previously created. If the list contains a suitable syringe, you can select it, and close the dialog by clicking *Apply Syringe*. Alternatively, you can simply select a syringe by double clicking the syringe.

If the list does not contain a suitable syringe, a new syringe can be created by clicking the *Create* 2 button. Clicking the *Edit* button opens a window to edit the syringe parameters of the currently marked syringe. You can remove a syringe from the list by clicking the *Delete* button. A syringe can also be removed by pressing the *Delete key*.

On the right you will see a preview image 3 of the currently selected syringe and all syringe parameters

The selection can be completed at any time by clicking *Apply Syringe* 4 or cancelled by clicking *Cancel*.

7.9.4 List of available syringes

The list of available syringes shows all available syringes in tabular form. It contains both the standard syringes defined by CETONI and the syringes created by the user. The icon in the first column indicates whether the syringe is a standard syringe or a user-specific syringe:

Predefined standard syringe from CETONI

Syringe created by user

IMPORTANT. Predefined syringes can not be edited or deleted.

You will find the following columns in the syringe list:

NAME	Unique syringe name. Syringes with the same name are not allowed.
MATERIAL	Steel or glass – the syringe display is adjusted accordingly
MAX. PRESSURE (BAR)	Maximum syringe pressure – important for pressure monitoring
SUITED FOR PUMP	Shows for which pump a syringe is suited. If this field is empty, it is a syringe that was created by the user.

HINT. If you click in the header of a column, the syringe list is sorted according to this column. This is indicated by a small arrow above the column title. By clicking again in the same column header, you can reverse the sorting.

To search for a specific syringe in the list, simply right-click on a cell in the column you want to search. Usually this will be the *Name* column. From the context menu that is then displayed, choose *Search in column*.

A search field is displayed in which you can enter your search text 1.

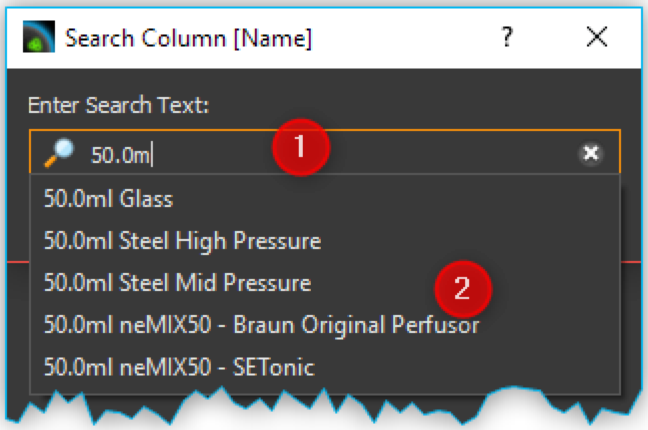
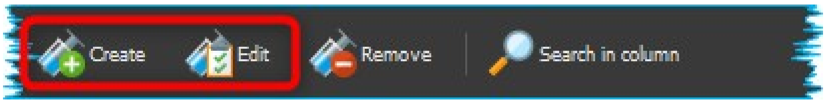


Figure 157: Search dialog for syringe list

While typing, a list of possible hits is displayed 2. As soon as there is a matching entry, select it with the cursor keys or click on the entry with the mouse. The search dialog will be closed and in the list of syringes the found syringe will be highlighted in color.

7.9.5 Configuring the syringe parameters



If you click the *Create* or *Edit* button, the (*Syringe Configuration Wizard*) for configuring the syringe parameters opens. Please configure a syringe with the following steps:

(1) First assign a unique name to the syringe in the field *Unique syringe name* 1 (Figure below). If you have selected a syringe for editing and only change its name, you can generate a copy of the selected syringe. Then select the syringe material 2. Click *Next* to go to the next step.

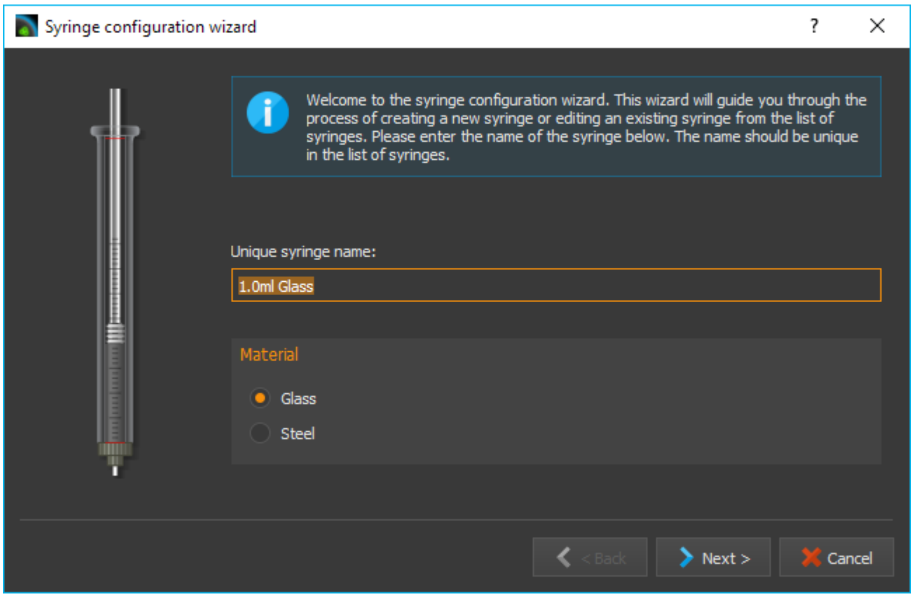


Figure 158: Syringe configuration step 1 - Enter name

(2) You should now configure the inner diameter of the syringe. The inner diameter of the syringe is required in order to calculate flow rates and dose quantities. If you do not know the inner diameter, you can alternatively define the inner diameter via the *Scale length* in mm, and the volume it contains via *Scale volume* in µl. The software then uses these values to calculate the inner diameter of the syringe. In this step, select how you want to configure the inner diameter of the syringe. (Figure below). Depending on the selection, you continue the configuration with point (3) or (4). Click *Next* to go to the next step.

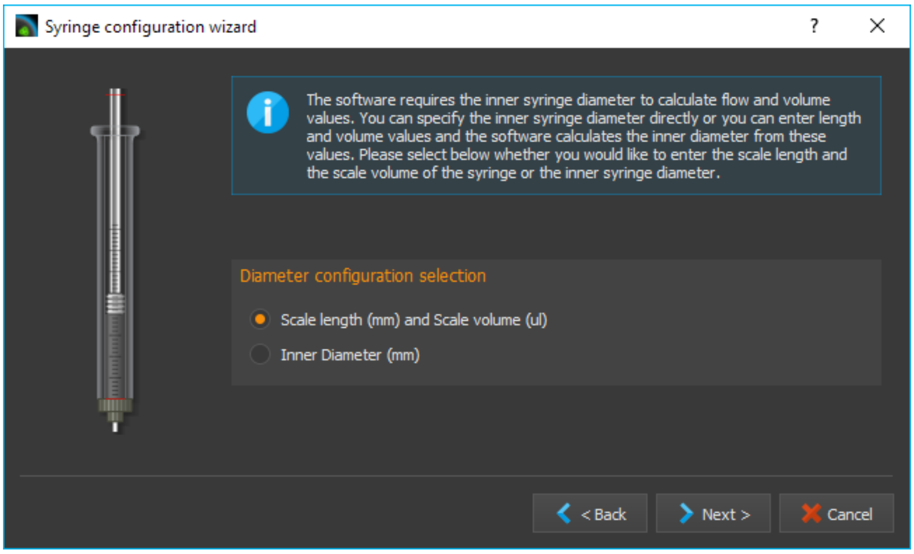
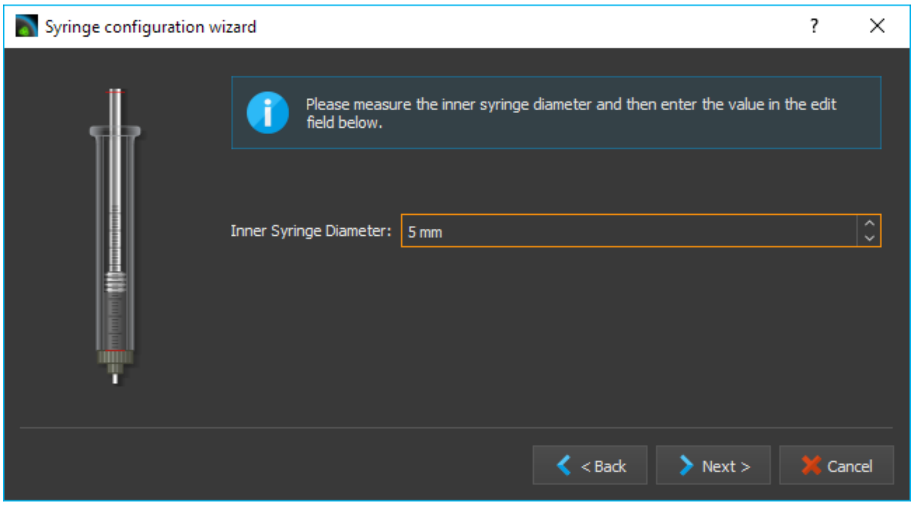


Figure 159: Syringe configuration step 2 - Select method

(3) You configure the inner diameter of the syringe by entering the scale length and volume within this scale length. (Figure below). To do this, measure, the scale with a suitable measuring instrument (calliper gauge), and read the volume shown on the inscription on the scale. Then click *Next* to go to the next step, and read further at point **(5)**.

(4) You configure the inner diameter of the syringe by measuring it with a calliper gauge. Then enter the value in the *Inner Syringe Diameter* field, and click *Next* to go to the next step.



(5) Now configure the piston stroke (*Piston Stroke*) of the syringe (Figure below). The piston stroke is the maximum length through which the piston can be moved in the syringe without it leaking and liquid emerging. It is advisable to use the length of the scale on the syringe as the maximum piston stroke.

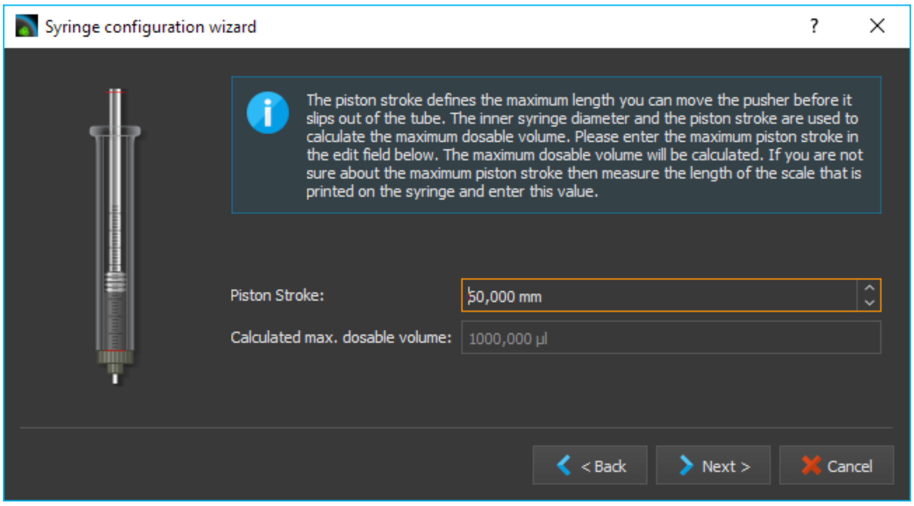


Figure 161: Syringe configuration step 2 – Piston stroke

If syringes of different lengths are used in a multi-syringe holder, the syringe with the shortest piston stroke limits the range of travel of the drive. When you have entered the piston stroke, the software uses this and the inner diameter to calculate the maximum dosable volume.

(6) In this step you can define the limits for the maximum and minimum syringe fill levels. (Figure below). In this way, you can limit the range of travel of the piston by two additional software limits. You can enter these parameters optionally in microlitres or millimetres. You should use these values to limit the range of travel, for example when using a syringe stirrer. If you enter the value in one unit (e.g. millimetres), the value in the other unit (e.g. microlitres) is calculated by the software.

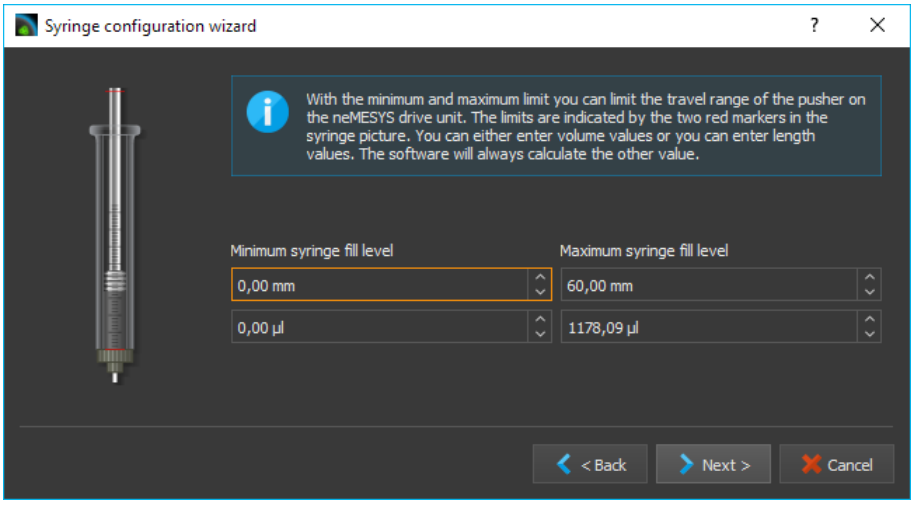


Figure 162: Syringe configuration step 6 - Limits

The syringe preview on the left-hand side always shows you the current configuration of the syringe. The width of the syringe changes when the inner diameter changes. The length of the syringe changes when the maximum piston stroke is changed, and the two red marks on the syringe indicate the two additional limits. The syringe is subsequently also shown corresponding to this preview in the software.

(7) Finally enter the maximum pressure the syringe can be used with. This value is the maximum value you can use for pressure monitoring. (see section 7.13.Pressure- / Analog-In-Monitoring)

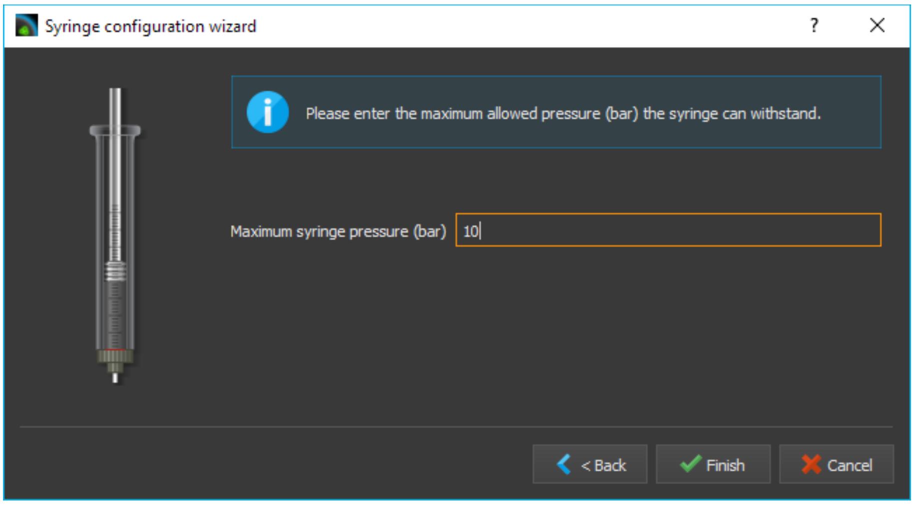


Figure 163: Syringe configuration step 7 - maximum pressure

(8) Close the configuration of the syringe parameters by clicking the *Finish* button. You can cancel the configuration at any time by pressing the *Cancel* button. You can always return to the previous configuration step with the *Back* button.

7.10Configuration of SI units

When dosing liquid flows, the main parameters are the flow rates used and the quantities of liquid to be dosed or already dosed. For optimal adaptation to the particular application, the user can configure the SI units individually for displaying the flow rates and volumes for each individual dosing unit. To show the SI-dialog, click with the right mouse button in the control panel of the pump to display the context menu.

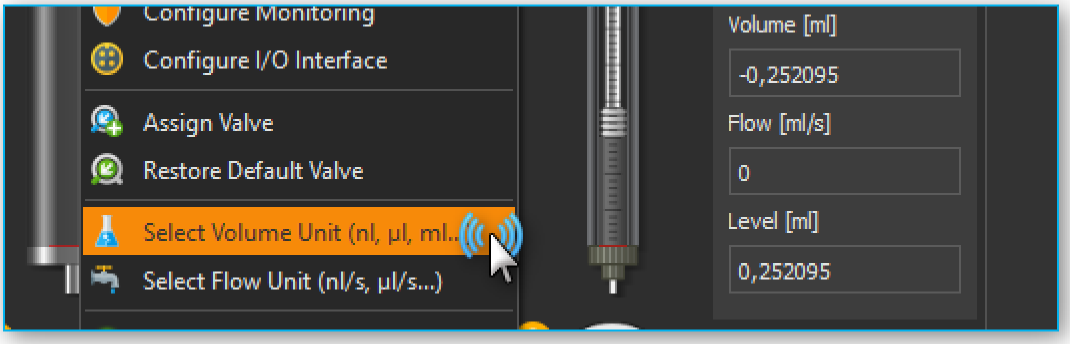


Figure 164: Pump context menu

Then click the menu item *Select Volume Unit* to configure the SI unit of volume or click the menu item *Select Flow Unit*, to configure the SI unit of flow rates. A SI-unit selection dialog window appears.

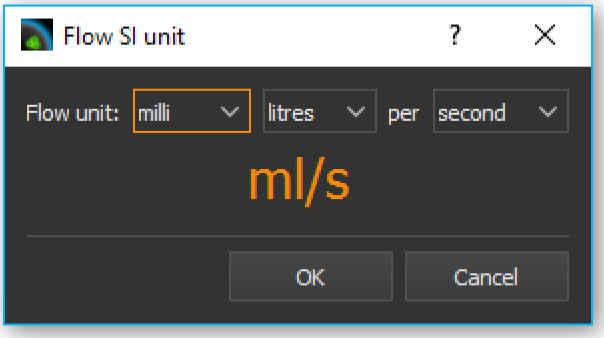
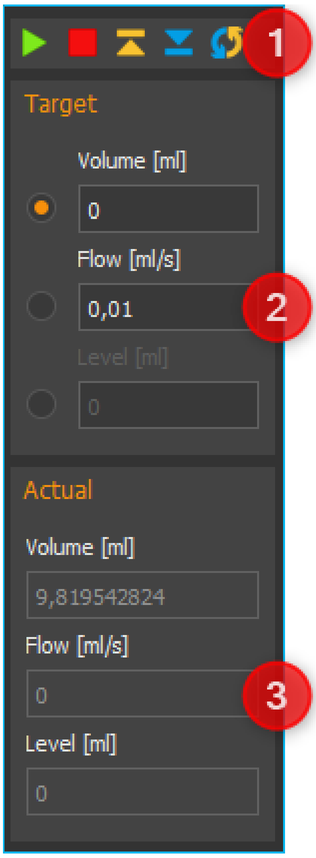


Figure 165: SI-unit selection dialog

Set the desired SI unit, and close the dialog by clicking OK.

7.11 Direct Control

7.11.1 Overview



Use the direct control for interactive control of the dosing units. In this way, you dose precisely defined quantities of liquid with defined flow rates or generate constant liquid flows.

The direct control also displays the current flow rate, the dosed volume and the fill level of the syringe. If you use a high-pressure module, the current pressure measured by the pressure sensor is also displayed.

Following controls and indicators are available:

1 Toolbar
2 Target values of the dosing module
3 Actual values of the dosing module

7.11.2 Toolbar

The toolbar includes buttons for starting dosing processes. The following buttons are available:


Start the dosing with set target values


Stop dosing


Aspirate reagent - Move dosing unit to maximum position


Empty syringe - Move dosing unit to minimum position

Configure the continuous flow

ATTENTION. Danger of damaging the syringe during emptying! When the syringe is emptied, according to the configuration, the dosing unit is moved toward the lower limit position at high speed.

7.11.3 Target value

7.11.3.1 SELECTING DOSING MODE

You define the type of dosing with the selection buttons on the on the left-hand side.

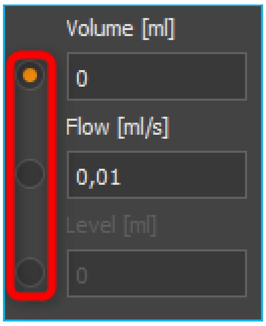


Figure 166: Select dosing mode

The following options are available:

- *Volume* - Select this mode to dose a specific volume at a defined flow rate
- *Flow* - In this mode, you generate a constant flow of liquid. The dosing continues until a limit position is reached or until you stop the dosing process.
- *Syringe Level* - In this mode, you can define a specific fill level for the syringe which is to be reached at a defined flow rate.

7.11.3.2 SETTING FLOW RATE AND VOLUME

Use the *Volume*, *Flow* and *Syringe Level* input fields in order to enter the flow rate, the flow volume or the syringe level. Setting the set-points does not start the drives or change the current flow rate. The new values are not transferred to the dosing unit until a dosing process is started by pressing the *Start* button in the toolbar.

7.11.3.3 VOLUME DOSING

If you want to dose a specific volume, you have to enter the volume to be dosed and the flow rate. The volume is entered relative to the current position of the syringe piston. This means that you enter a negative volume to aspirate reagent and a positive volume to dispense reagent. In this operating mode, the flow rate is always a positive value, and defines the flow rate of the aspiration/dispensing of reagent.

7.11.3.4 CONSTANT FLOW RATE

In order to generate a constant flow of liquid, only the flow rate has to be defined, and the volume input field is disabled for entries. In this operating mode, the flow rate can be either positive or negative. A negative value indicates reagent aspiration, and a positive value reagent dispensing.

7.11.3.5 SETTING SYRINGE LEVEL

You can achieve a precisely defined syringe fill level by entering the values for the syringe fill level and the flow rate. The syringe fill level must be a positive value lying between the minimum and maximum values of the syringe fill level. (see section 7.9 Syringe configuration). In this operating mode, the flow rate is always a positive value, and defines the flow rate of the aspiration/dispensing of reagent.

ATTENTION. Danger of damaging the valve or connections in the liquid path High flow rates can cause high pressures to develop very quickly, which can damage the valve, connections in the liquid path or your application.

IMPORTANT. For dosing very low flow rates, select syringes with small inner diameters in order to ensure pulsation-free dosing.

7.12 Actual Values

The current values reported by the device are shown in the *Actual Values* area.

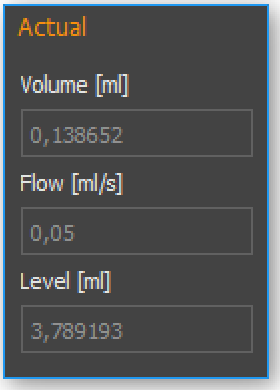


Figure 167: Actual values

The *Flow* field shows the current flow rate in the dosing unit. The *Volume* field shows the volume that has been pumped since the last time the dosing unit was started. If the drive is stopped and restarted, the actual value of the pumped volume is reset to 0. The *Syringe Level* field shows the current fill level of the dosing unit in the configured SI volume unit.

7.13 Synchronous pump start / stop

In the main toolbar, you will find two buttons for simultaneously start / stop multiple pump modules.

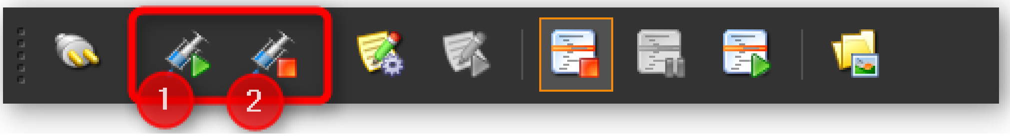


Figure 168: Synchronously pump start / stop

When you click the *Start* button 1, a selection dialog appears in which you select the pumps that you want to start at the same time (see figure below).

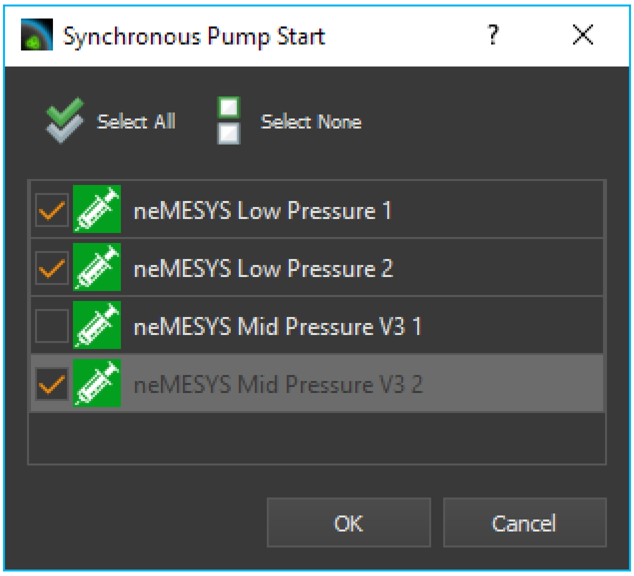


Figure 169: Pump selection for synchronous start

Set a checkmark for each pump you want to start. Dosage starts as soon as you klick the *OK* button.

IMPORTANT. Before starting, you need to configure the dosing parameters (volume, flow) of all selected pumps in the control panel of each pump.

Before the software starts the dosage, it checks the configured parameters of all selected pumps. If the software found invalid parameters (eg flow or volume values of 0), the synchronous start is not performed. In the application *Event Log* you will get information about the problems encountered.

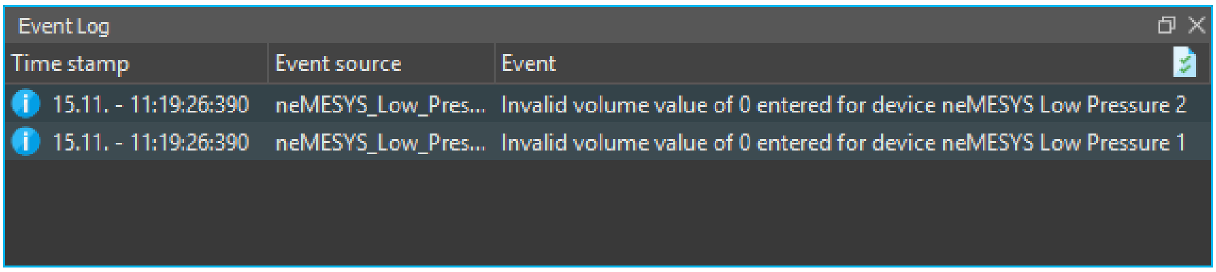


Figure 170: Sync Start errors in the event log

When you click the *Stop* button 2, all pumps are stopped immediately.

7.14 Continuous Flow

7.14.1 Introduction

In continuous flow mode you can pair two pumps together using the software, in order to create a continuous flow over a long period of time through alternating reagent take-up and discharge. In this case one of the pumps doses at a particular flow rate, while the other pump takes up reagent from the reservoir. Once the dosing pump has emptied the syringe, the software automatically switches to the second dosing unit with a full syringe and continues dosing with that pump. This creates a continuous, uninterrupted flow, which lets you dose at a constant flow rate for a nearly unlimited period of time.

7.14.2 Operating and Display Elements

You can recognize a dosing unit that was configured for continuous flow by the activated continuous flow icon in the direct control window (see image).

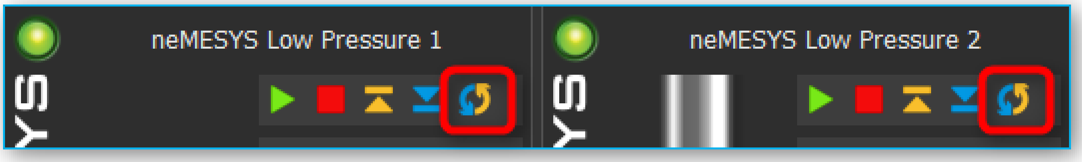


Figure 171: Continuous flow mode indicator

The continuous flow mode is terminated automatically, as soon as you change the syringe configuration of one of the connected dosing units. In this case you have to reconfigure all parameters. If continuous flow has been activated for one of the dosing units, the operating panel will indicate “continuous flow” in its status display (see image below).

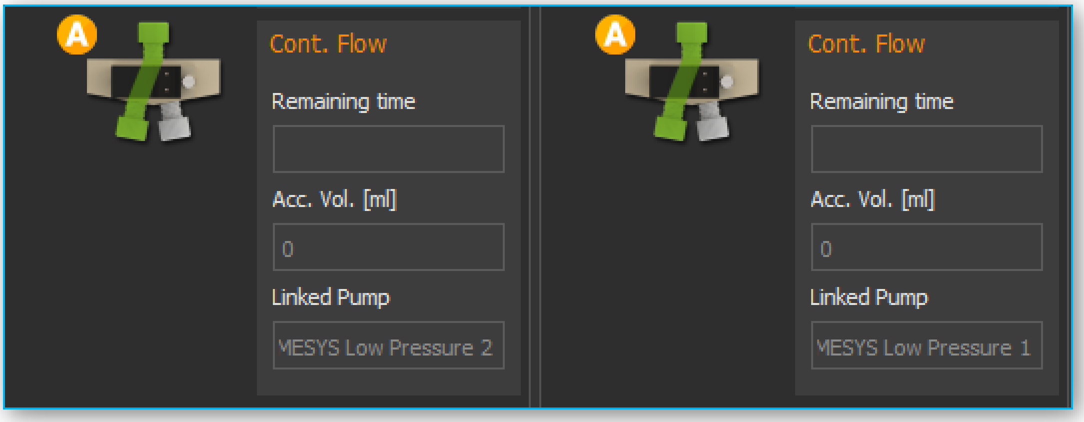


Figure 172: Continuous Flow Status Display

The following display elements will then be visible:

- *Remaining time* – shows the remaining time for continuous dosing. When this display reaches zero, continuous flow is stopped automatically.
- *Accumulated volume [ml]* – the accumulated discharged volume shows the total volume discharged by both pumps into the application.
- *Linked pump* – the linked pump is the second pump paired to this pump for the purpose of achieving continuous flow.

7.14.3 Configuring Continuous Flow

You will need at least two neMESYS dosing units to achieve continuous flow. Both units should have a valve. To configure continuous flow perform a right mouse click on the operating panel of the direct controller and select "configure continuous flow" from the context menu (see image below).

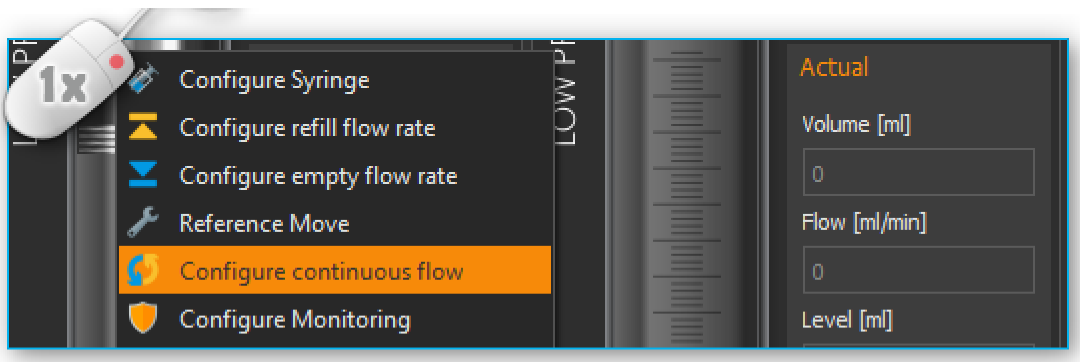
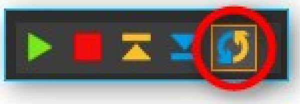


Figure 173: Continuous Flow Configuration

A configuration dialog will appear. The dosing unit you click on for configuring continuous flow is the first of the dosing units to be paired. If continuous flow has not been configured for this dosing unit, you can also open the configuration dialog by pressing the “continuous flow” button in the direct control window (see image below).



If you see the following configuration page after calling up the dialog (see image below), the previously set configuration was loaded successfully and all you have to do is click the Finish button to accept the configuration.

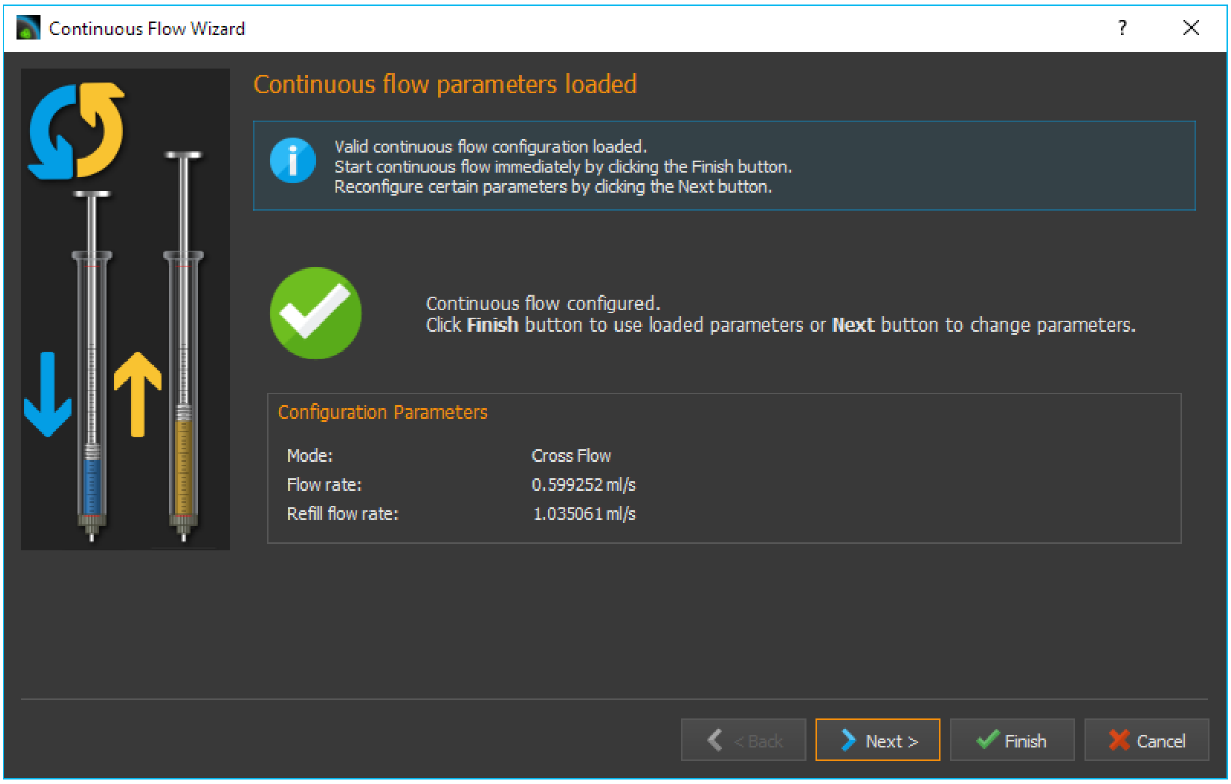


Figure 174: Continuous Flow Configuration Parameters Successfully Loaded

If you want to change certain parameters of the loaded configuration, click on the Next button to browse the individual pages of the configuration assistant.

7.14.4 Selecting the second Dosing Unit

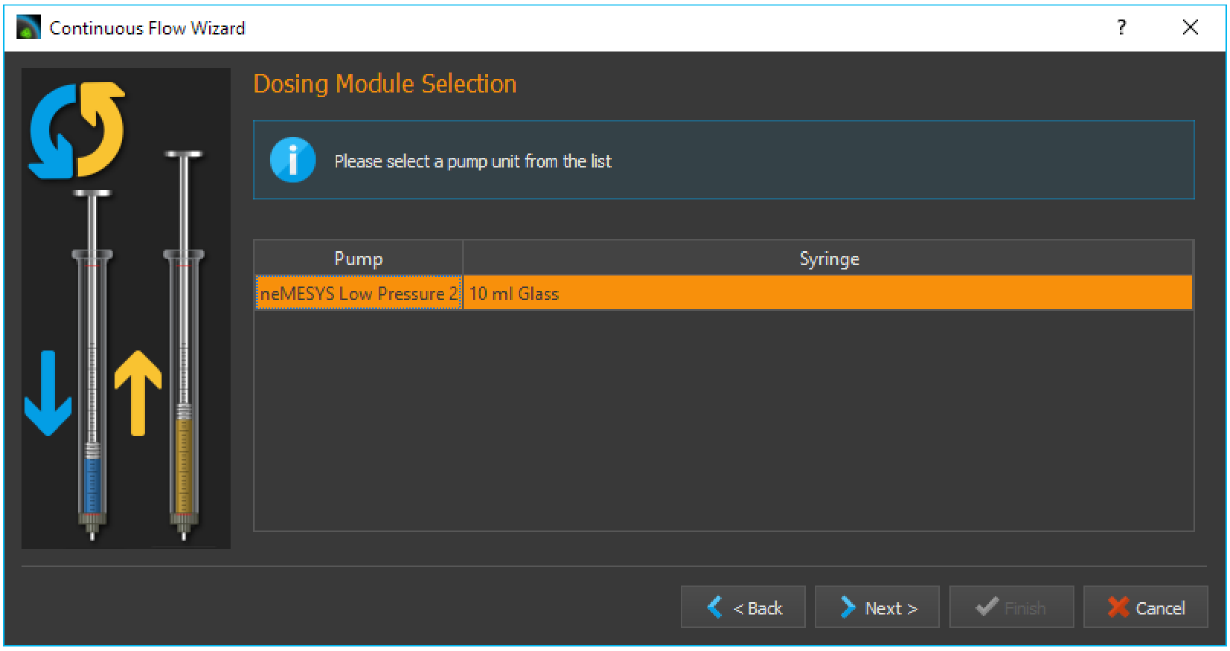


Figure 14: Selecting the Second Dosing Unit

On this page of the configuration dialog you can select the second continuous flow unit from a list of available dosing units. If no configuration has been loaded, this will be the first page of the configuration assistant. Select the dosing unit of your choice by clicking the relevant name in the list. Then click on Next to continue the configuration.

IMPORTANT. For continuous flow you should always select two dosing units that are immediately connected and displayed next to each other in the software.

7.14.5 Selecting a Continuous Flow Mode

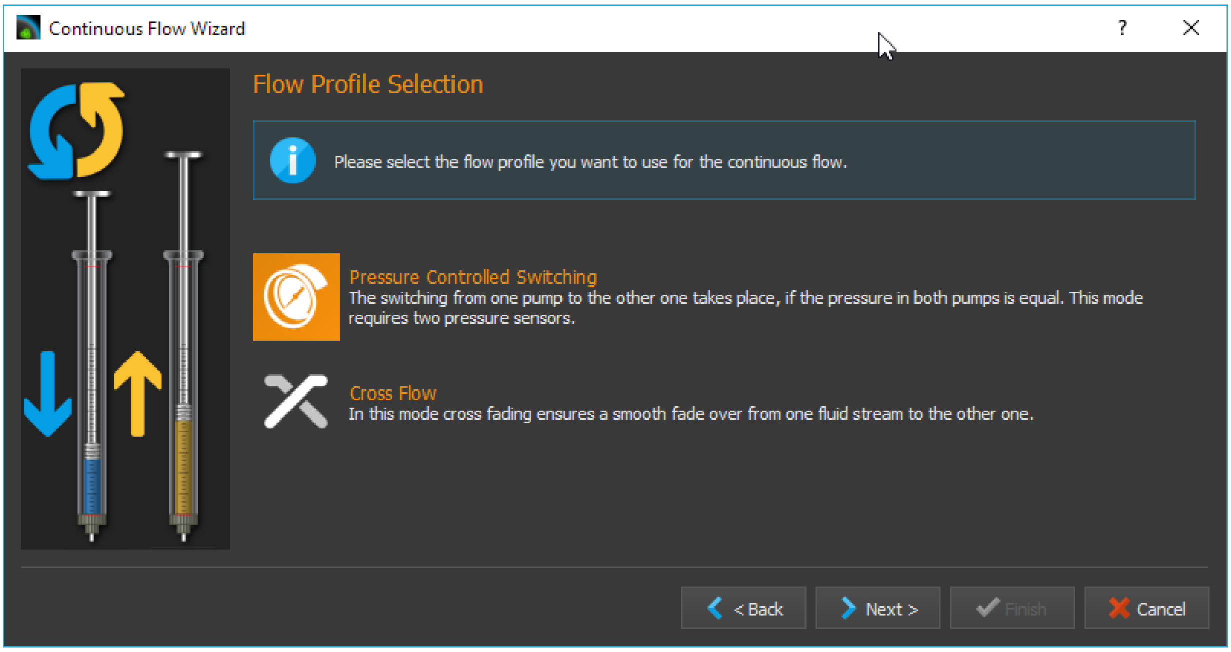


Figure 175: Selecting a Continuous Flow Mode

You can choose from the following continuous flow modes:

- *Pressure Controlled Switching* - yields the best results without changing pressure or flow rate when switching units. However, you will need a pressure sensor on each pump.
- *Cross-Flow* - provides a soft transition and a constant flow rate when switching from one dosing unit to the other. This mode is most suitable if your application uses a low system pressure and you don't have pressure sensors.

The individual modes will be explained in detail in the following subsections. Perform a left mouse click on the mode you wish to configure and then click on Next to continue.

7.14.6 Configuring 3x2-way Valves

During continuous flow operation the respective valves get switched for both pumps. In the *Configure 3x2-way Valves* window (see image below) you can configure the valves and valve positions for your 3x2-way valves.

HINT. If you have already assigned valves to your pumps, these valves are automatically selected here.

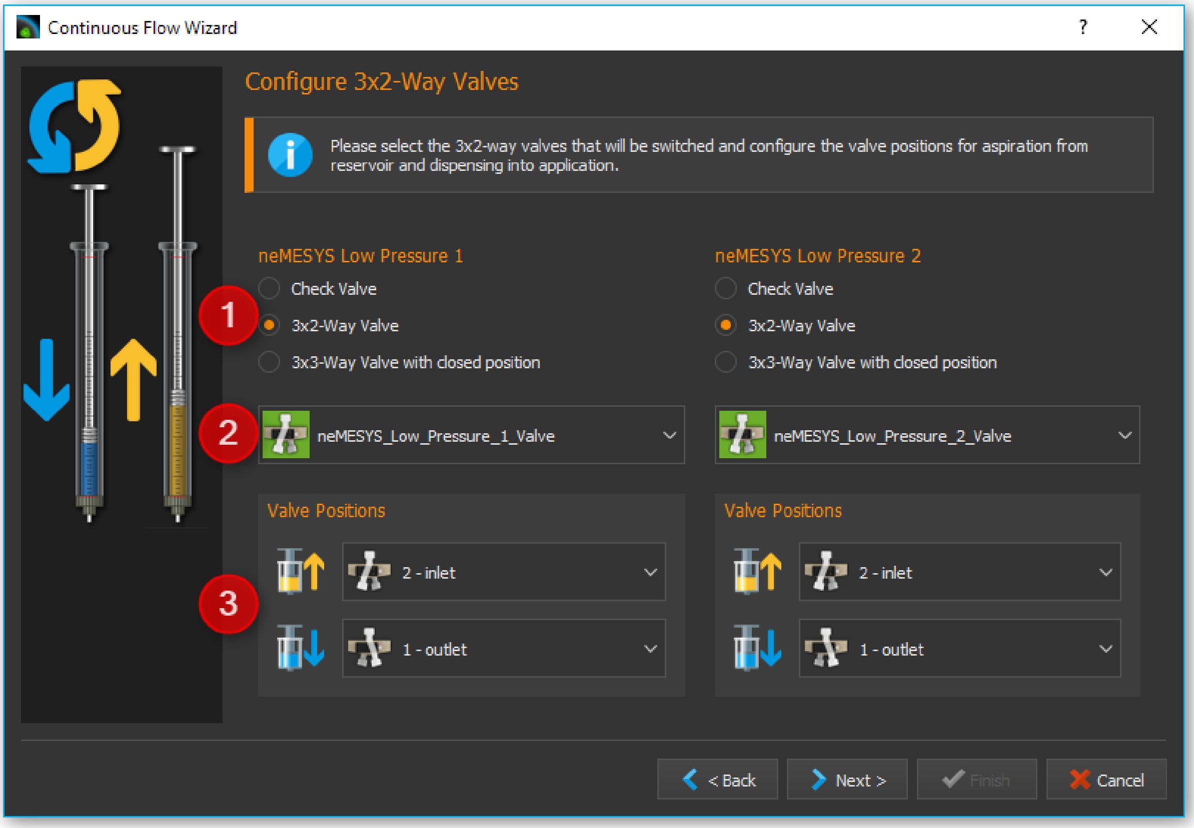


Figure 15: Valve Configuration for Continuous Flow Operation

First, select the valve type you wish to use 1. If you use 3x2-way valves and have selected *Pressure-Controlled Switching*, you will need additional shut-off valves. If you use 3x3-way valves with an additional closed position, you will not require additional shut-off valves for pressure-controlled switching.

Use the selection box 2 to select the valve you want to switch. Using the two selection boxes below 3, select the positions you want the valve to switch to when filling and dosing. If you have a 3x3-way valve with an additional closed position, you also need to configure the closed position.

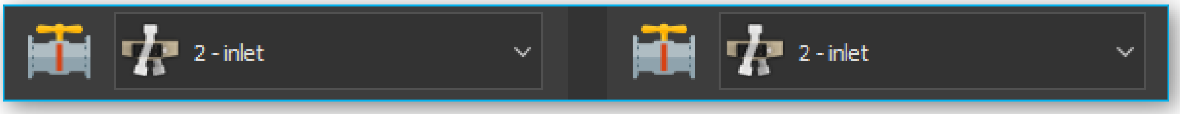


Figure 16: Configuring The Closed Valve Position

7.14.7 Configuring Shut-off Valves

If you have selected pressure-controlled switching and your 3x2-way valves do not have a closed position, you will need to configure your shut-off valves in this next step. As in the valve dialog above, please select the valves and configure the closed 1 and open 2 valve position.

Figure 17: Configuration of Shut-off Valves

7.14.8 Configuring Pressure Sensors

If you have selected pressure-controlled switching, you need to configure the pressure sensors for both pumps in this next step (see image below).

HINT. If you have already assigned pressure sensors to your pumps that are displayed on the front panel, these sensors are automatically selected here, and you can proceed immediately by clicking **Next >**.

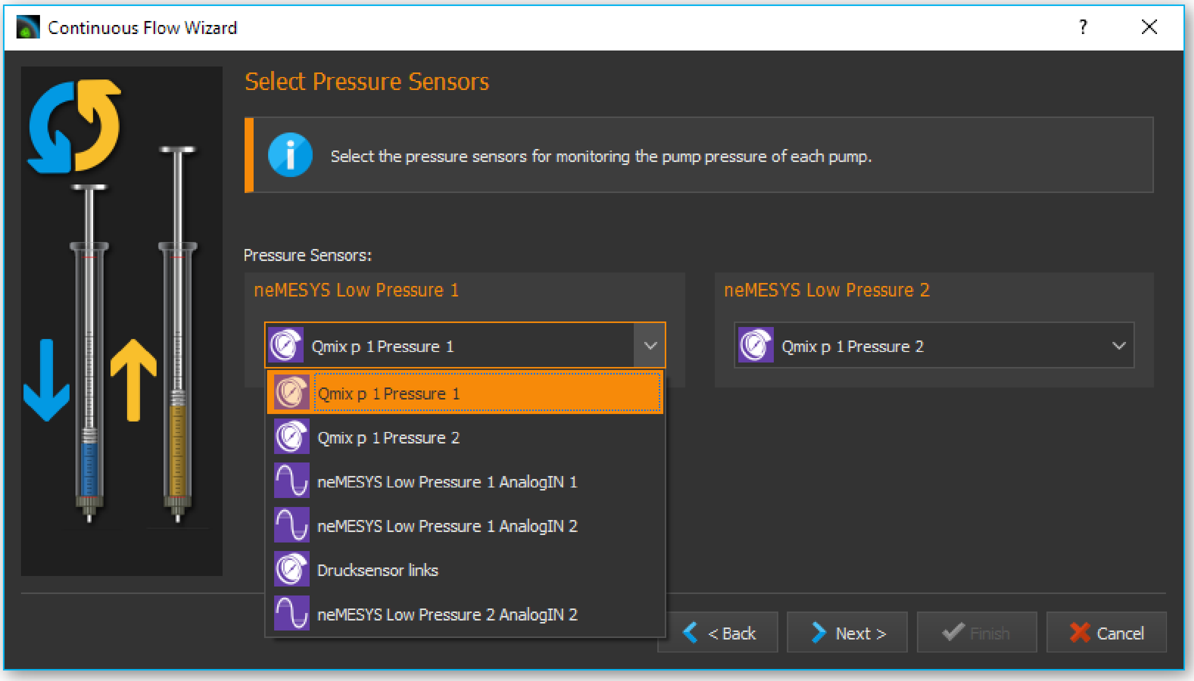


Figure 18: Configuration of Pressure Sensors for Pressure-controlled Switching

7.14.9 Pressure-controlled Switching

7.14.9.1 INTRODUCTION TO PRESSURE-CONTROLLED SWITCHING

In pressure-controlled switching the pressure of the filling pump is adapted to the pressure of the dosing pump prior to switching, through the use of valves and pressure sensors. In addition, during the switching process the flow of both pumps is cross-faded (cross-flow). This prevents the occurrence of a switching impulse, while pressure and flow rate stay nearly constant.

Since a certain period of time is needed to fill and cross-fade the pump, in order to build and reduce pressure and to switch the valves, the achievable flow rate in continuous mode is lower than the maximum flow rate of each of the individual pumps involved.

In the configuration window for pressure-controlled switching you can set all parameters and you can adjust the pressure controller of each pump for the current configuration and the desired application pressure.

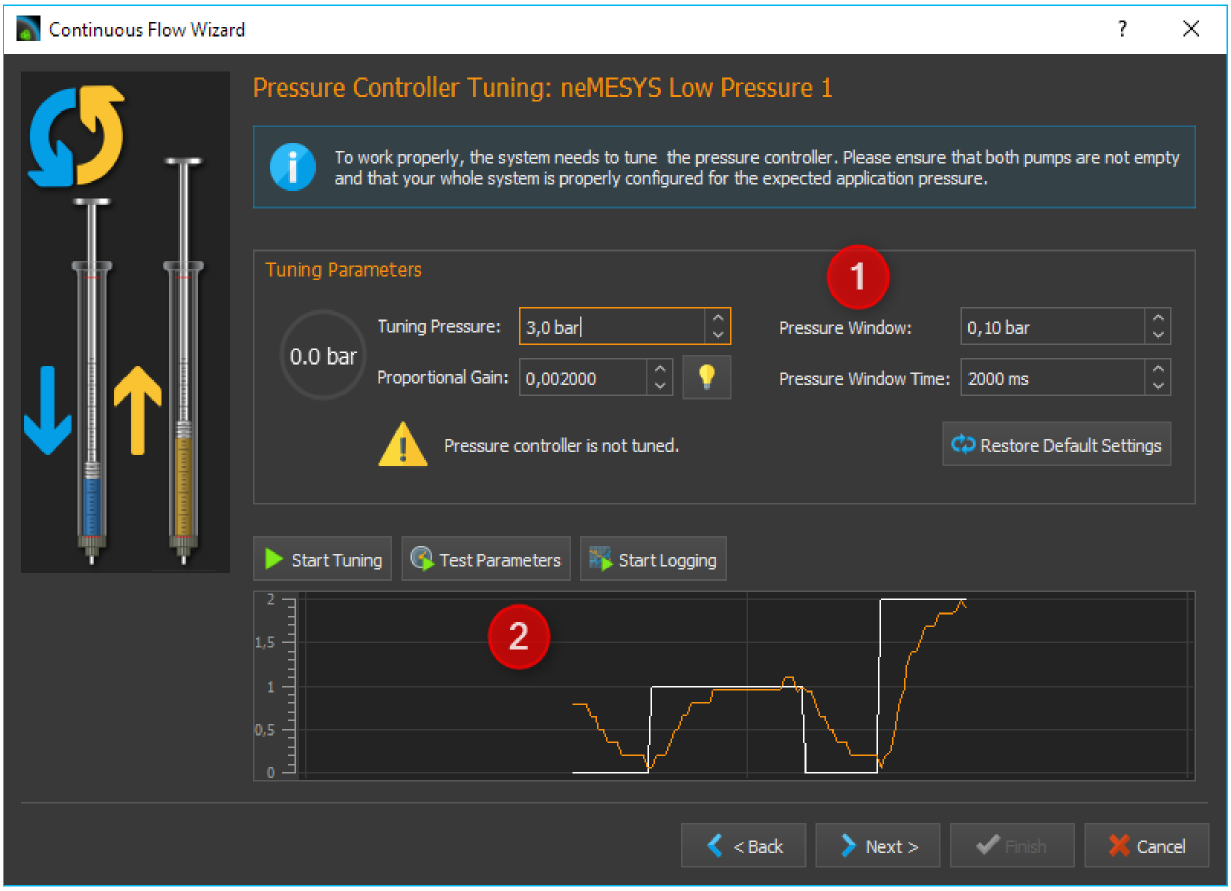


Figure 19: Configuration of Pressure-controlled Switching

In the upper part of the window ❶ you find the tuning parameters. In the lower part of the window ❷ you will find a diagram, which shows the controller's target value (white) and the measured pressure value (in color) during adjustment. Using this graphic representation you can get an idea of the pressure controller's operational status and detect issues like excessive oscillation.

CAUTION. Very high pressure may occur during tuning. If necessary, you should activate the pressure monitor, in order to stop the pump when certain pressure values are exceeded.

7.14.9.2 TUNING PARAMETER OVERVIEW

The continuous mode uses a dedicated pressure controller for each pump to regulate the target pressure in the filling pump before switching to the application. The proportional gain of the controller depends on many factors, such as pressure, tightness, tube length and the fluid being used. This value must be re-calibrated or retuned for each setup. The tuning section uses the following operating elements:

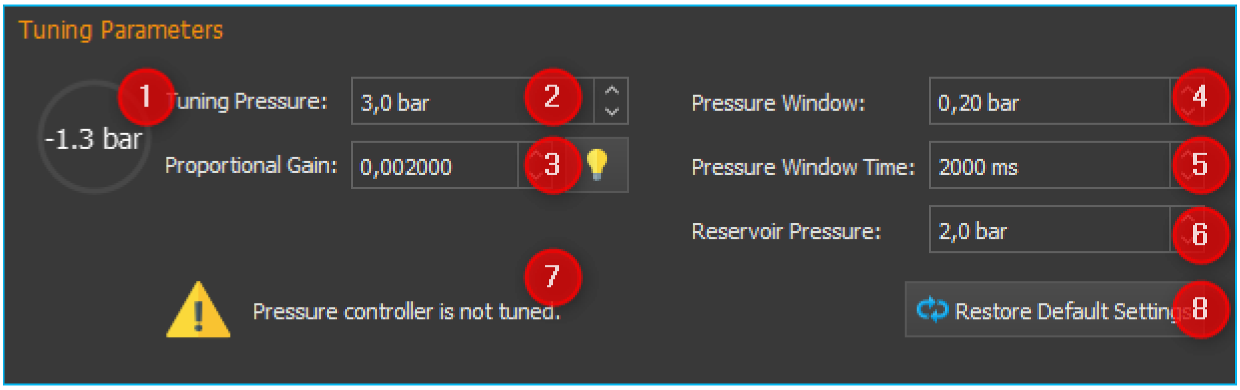


Figure 20: Tuning Parameters for Pressure-controlled Continuous Flow

1 **Pressure Display** – depending on the dosing direction (filling/discharge) the pressure display is colored in orange or blue.

2 **Tuning Pressure** – this is where you set the pressure to which the pressure controller will be tuned. Set the anticipated application pressure you will use for dosing into your application.

3 **Proportional Gain** – this field shows you the proportional gain of the pressure controller. If you click the button with the lamp symbol, the system will suggest a proportional gain for the target pressure you set. This value can be used as a starting point for tuning.

4 **Pressure Window** – Defines a symmetrical range of accepted pressure values relative to the target pressure. This means that only if the pressure is within this window for the duration of the *Pressure Window Time* it will be considered stable. A value that is too high may cause the controller to overshoot, a value that is too low may cause long times for the target pressure to settle, or may cause the target pressure never to be reached. The value that should be set depends on the target pressure and the quality of the pressure sensors used. The pressure window must always be greater than the fluctuations of the pressure sensor, otherwise the pressure will never be considered stable.

5 **Pressure Window Time** – The time the pressure must be within the *Pressure Window* for it to be considered stable by the software.

6 **Reservoir Pressure** – Set the pressure that exists in the reservoir from which the syringe is filled. The pressure is lowered to this pressure before the valve to the reservoir is opened to fill the syringe.

7 **Status Display** – the status display gives you information regarding the current status of the

pressure controller or the calibration. You should only go to the next step if you see a green checkmark here or if a previous tuning has been loaded.

8 Restore Default Settings – pushing this button restores the standard tuning parameters.

7.14.9.3 TUNING THE PRESSURE CONTROLLER

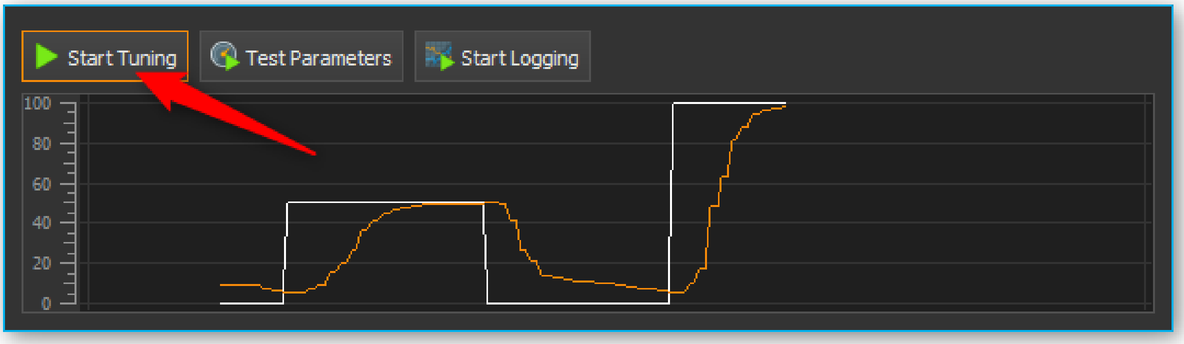
A pump's pressure controller should be tuned at the approximate piston position at which the switching to the other pump takes place. If you use two identical pumps, this usually happens when the syringe is almost completely filled. To make sure that the syringe can still be filled further for pressure building, the piston should not be in the rearmost position. As a rule of thumb the syringe should be filled to about 90%.

Please follow the steps below to tune the pressure controller.

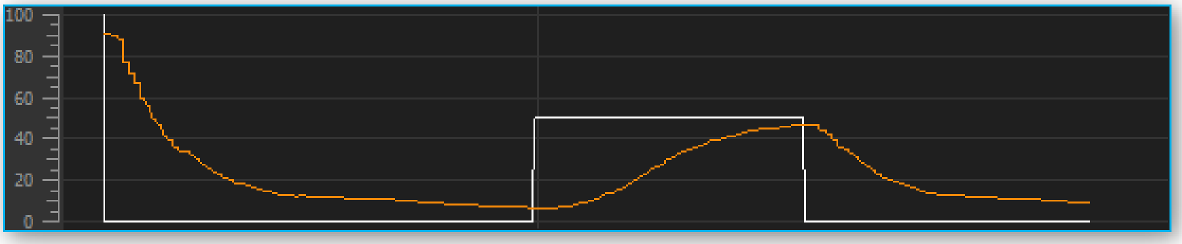
(1) First, enter the desired tuning pressure 2. This should be approximately equal to the application pressure at the desired flow rate. You can determine the tuning pressure ahead of time by dosing into your application at the desired flow rate.

(2) Now select the starting value for the proportional gain 4 of the controller. If you click on the button with the lamp symbol, the system will suggest a starting value for the selected pressure.

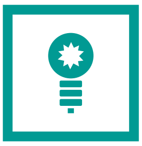
(3) Start the tuning process by clicking the *Start Tuning* button. The software will now start tuning the controller automatically. For safety reasons tuning will begin at 50% tuning pressure, to avoid damage due to pressure overshooting. The system tunes the controller in such way as to arrive at the target pressure quickly with a minimum of overshoot.



If achieving the target pressure takes a long time (10 seconds or more – see image below), you can interrupt the tuning process and increase the proportional gain manually (by doubling it, for example).

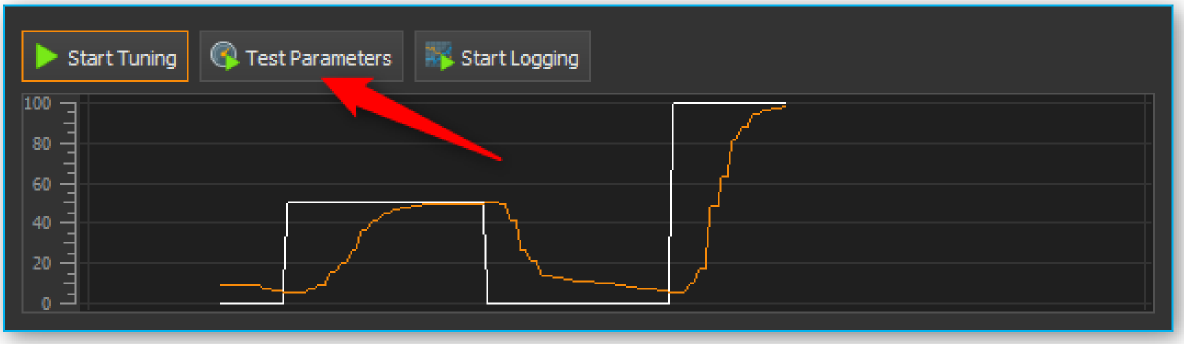


If the controller overshoots a lot, you can interrupt the tuning process and decrease the proportional gain manually (by halving it, for example). Restart the tuning process with the new values.



TIP. After a successful tuning, change the value for the target window (Pressure Window) and start another tuning run. You will get a feeling for how Pressure Window value affects controller tuning.

During tuning the controller's proportional gain is adjusted automatically. If you just want to manually test the current control parameters without the automatic adjustment, simply start a parameter test by clicking on *Test Parameters*.



During the parameter test the pump cycles between a non-pressurized state of 0 bar and the set tuning pressure. This lets you visually check and evaluate the current control parameters using the diagram.

Perform a right mouse-click on the diagram to open the context menu with additional diagram functions.

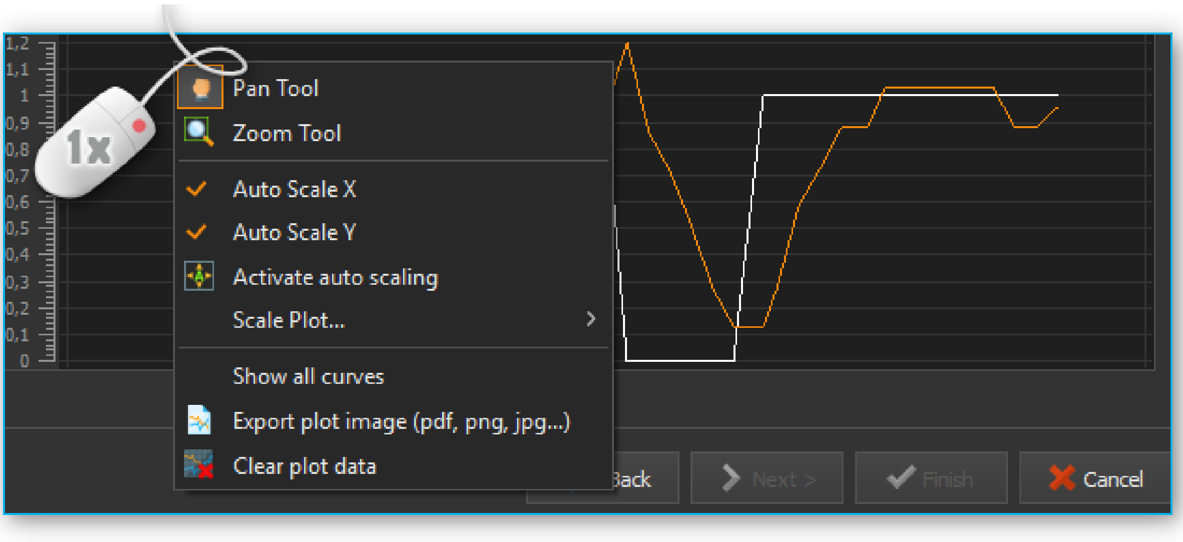


Figure 176: Context menu of Pressure Control Diagram

7.14.9.4 SETTING PARAMETERS

After tuning both pumps, go to the next page to set further parameters for pressure controlled switching.

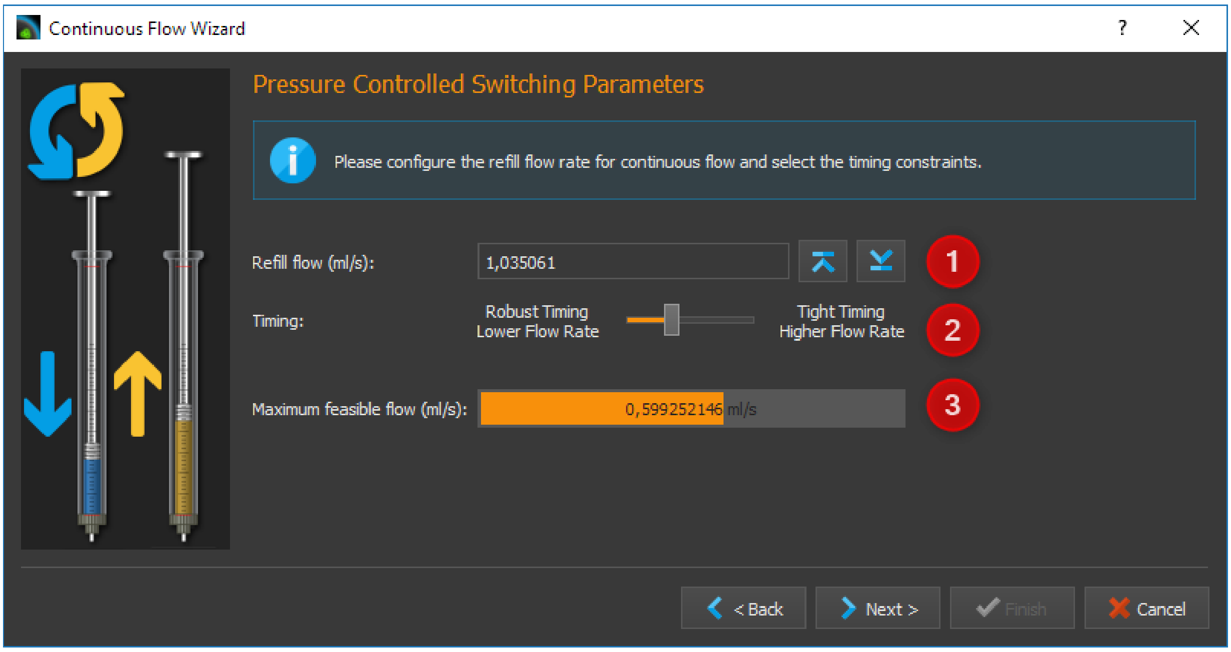
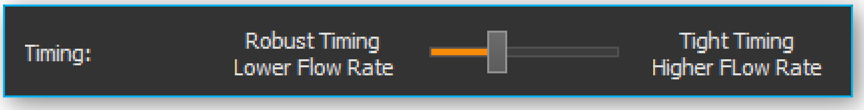


Figure 21: Continuous Flow Parameters for Pressure-controlled Switching

In the *Refill Flow* field 1, set the flow rate used to fill the syringe. The larger you select this value, the higher the maximum flow rate that can be realized for continuous dosing. This means that you should always select the filling flow rate as high as possible or as high as your application allows. If the filling flow rate is too high, air bubbles may form.

The *Timing* slider 2 lets you control the timing of continuous flow operation. During tuning, the volume and time periods needed for pressurizing the filling pump are determined. These time values are multiplied by a certain factor to create a safety margin. This buffer is used to compensate for delays during pressure building, which can be caused by issues or variations in the system. You can use the slider to select a higher margin (*Robust Timing*) or a lower margin (*Tight Timing*).



Using a larger buffer decreases the flow rates that can be achieved in comparison to a smaller buffer. In the same way, a smaller buffer increases the risk of disrupting the continuous flow timing in case of system issues or variations, possibly leading to flow instabilities. Double-clicking on the slider sets the standard value we recommend, unless you absolutely need a higher or lower flow rate.

IMPORTANT. A small safety buffer for the timing increases the risk that the timing of the continuous flow is disturbed and the continuous flow is interrupted if the application pressure fluctuates or if other disturbances occur.

In the field *Maximum feasible flow (ml/s)* 3 you can see the maximum flow rate that can be achieved with the set parameters. The coloured bar shows how far away this flow rate is from the maximum flow rate that you can achieve in normal pump operation mode. This allows you to quickly see how the refill flow rate and timing affect the realizable flow rate.

Once you have set all parameters, continue the configuration with section 7.14.11 Set flow rate and duration of continuous flow.

7.14.10 Configuring Cross-Flow

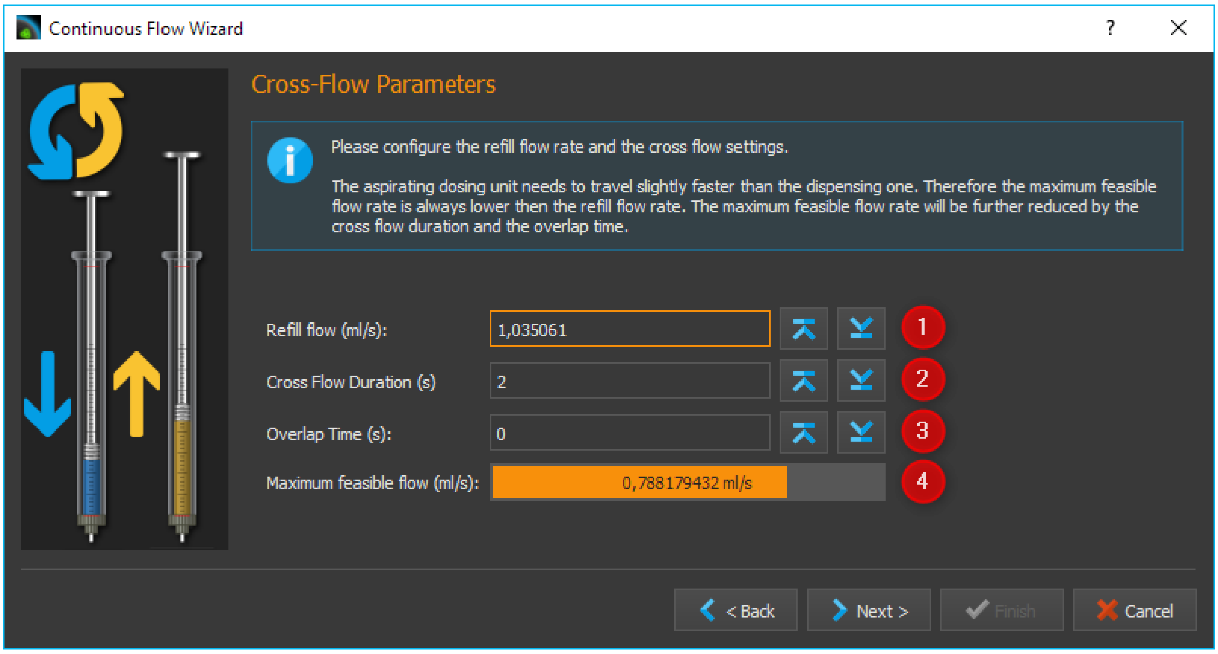


Figure 22: Continuous Flow Parameters Using the Example of Cross-Flow

In the *Refill Flow* field 1, set the flow rate used to fill the syringe. By pressing the *Min* and *Max* buttons. The larger you select this value, the higher the maximum flow rate that can be realized for continuous dosing. This means that you should always select the filling flow rate as high as possible or as high as your application allows. If the filling flow rate is too high, air bubbles may form. This flow rate is always slightly higher than the rate indicated in the *Maximum feasible flow* (ml/s) field 4, since the syringes must be filled more quickly in order to be ready for the next dosing run when the syringes are switched.

Cross-flow makes it possible to cross-fade the flows of the dosing modules. This is done by configuring the *Cross-flow Duration* 2.

In the field *Maximum feasible flow (ml/s)* 4 you can see the maximum flow rate that can be achieved with the set parameters. The coloured bar shows how far away this flow rate is from the maximum flow rate that you can achieve in normal pump operation mode. This allows you to quickly see how the refill flow rate and timing affect the realizable flow rate.

The following image shows the cross-fading of flows from two individual pumps.

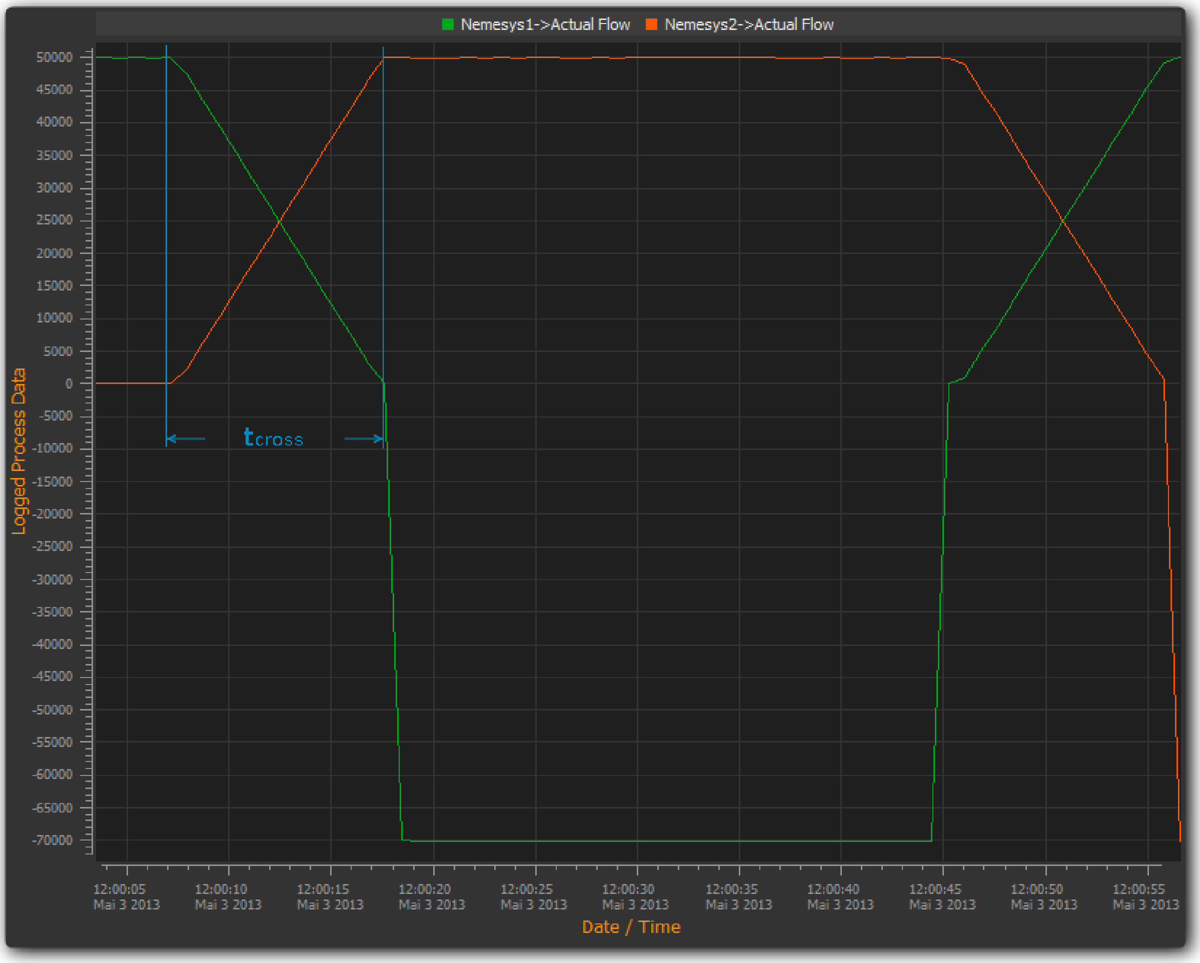


Figure 23: Cross-flow Profile

The *Cross-Flow-Duration* t_{cross} expresses the period of time needed for cross-fading one dosing module to the other. The smaller this value, the steeper the flow rate curve and the less time is required for the cross-flow operation to be completed.

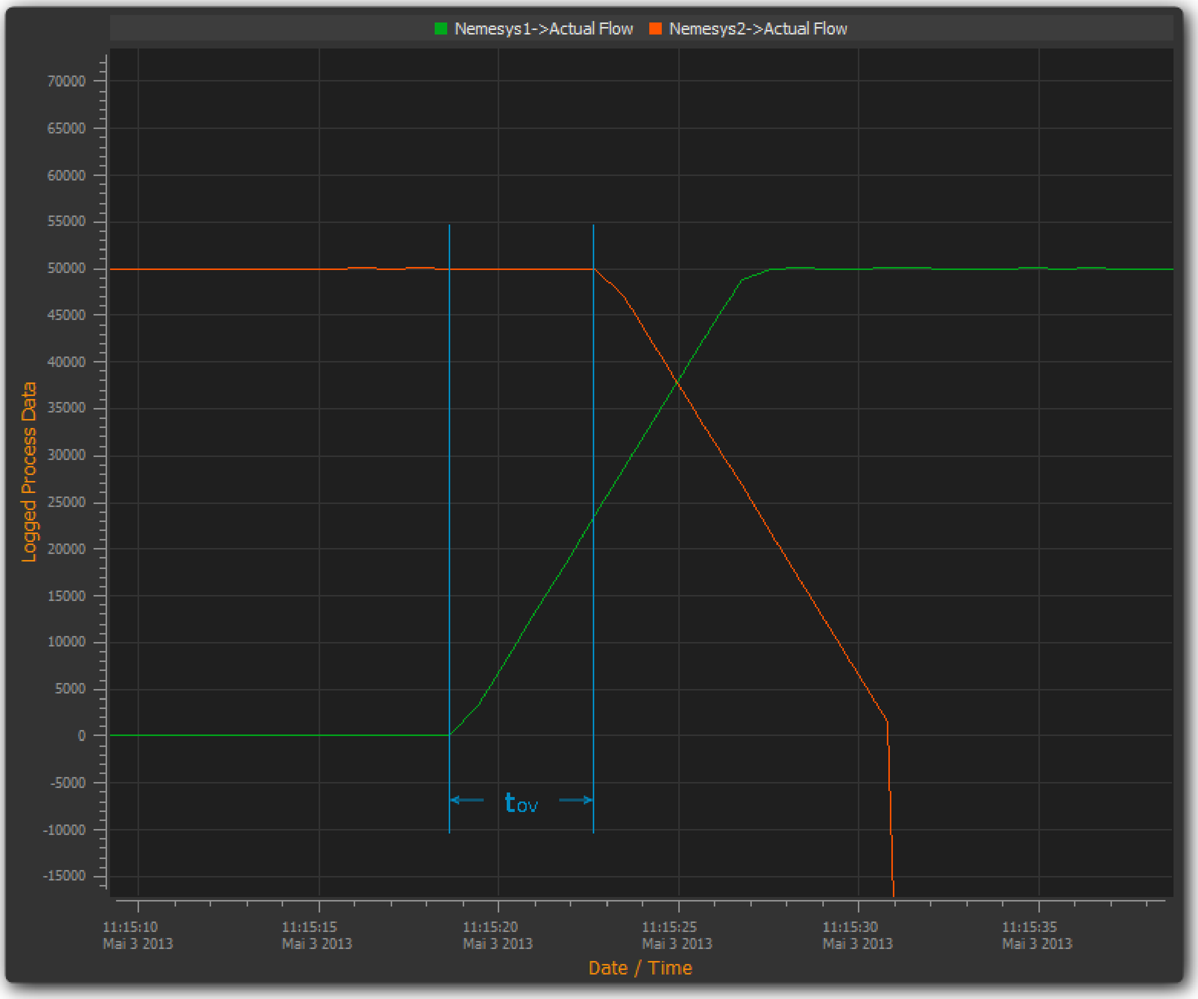


Figure 24: Overlap Time

Cross-flow offers you a simple way of compensating for pressure drops during switching. The *Overlap Time* t_{ov} ❸ governs the period of time by which the flow curves of both dosing modules overlap. The larger the t_{ov} value, the longer both dosing modules keep dosing simultaneously (see image above).

Once you have set all parameters, continue the configuration with section 7.14.11 Set flow rate and duration of continuous flow.

7.14.11 Set flow rate and duration of continuous flow

In this window (figure below) you set the flow rate at which the continuous flow is started ❶ and the duration of dosing ❷. If you don't want to limit the duration, simply leave the setting at *Dose unlimited*. With this setting continuous flow is maintained until you stop the dosing process manually.

The software offers you three ways of limiting the duration of continuous flow:

- *Dose volume* – continuous flow is stopped as soon as a defined dosing volume has been reached.
- *Dose for* – continuous flow ends after a predetermined duration of time.
- *Dose until* – continuous flow ends at a specific time in the future.

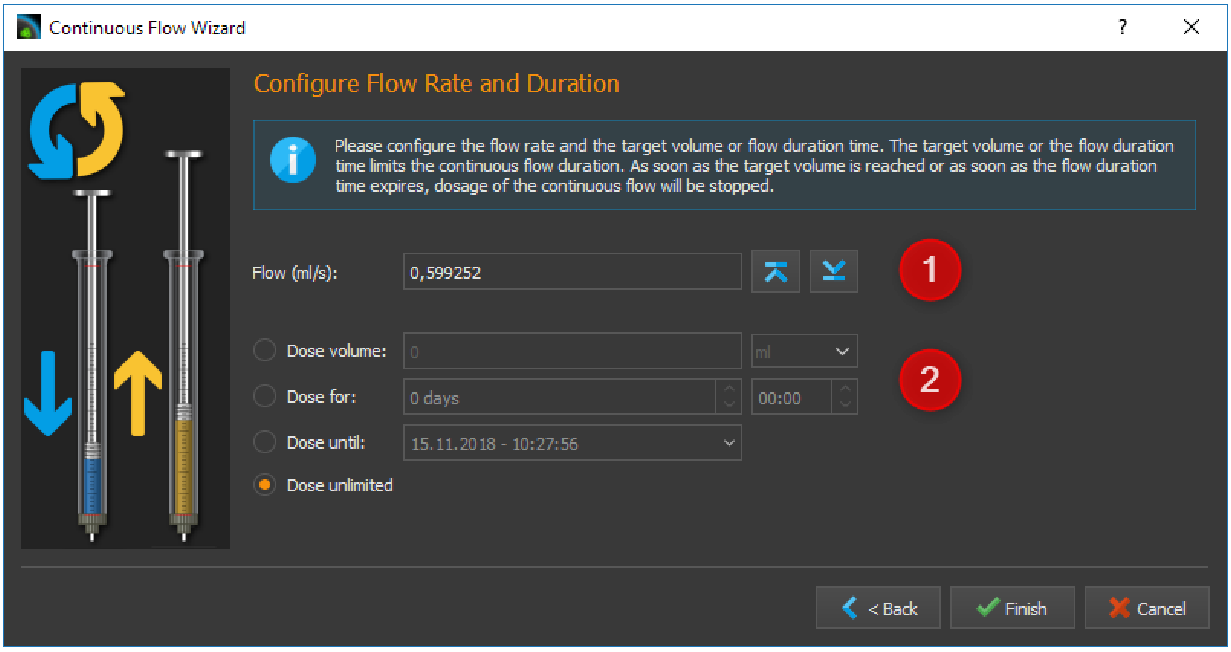
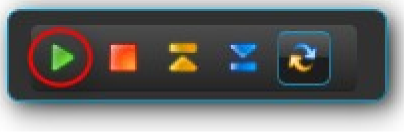
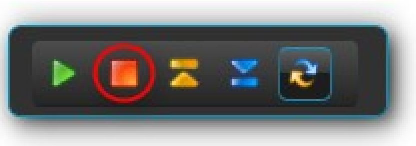


Abbildung 25: Konfiguration of continuous flow rate and duration

7.14.12 Starting/Stopping Continuous Flow



After the configuration you can start continuous flow operation by a left mouse-click on the Start Dosing button in the direct controller window.



You can interrupt continuous flow operation at any time by clicking the Stop Dosing button.

If you stop a dosing unit involved in continuous flow operation, you also stop the other dosing unit tied into the system. If you have configured a particular duration for continuous flow operation, stopping a dosing unit will also interrupt the preset duration, meaning that when you restart the dosing unit, continuous flow operation will resume for the time remaining at the point of interruption. If you configured a target volume for continuous flow operation, the system will resume dosing the remaining volume at the time of interruption, upon restart.

7.14.13 Cancel/Restart Continuous Flow

Once the preset time for continuous flow operation has elapsed or the target volume has been reached, continuous flow operation is terminated and both dosing units are stopped. Performing a left click on the *Start* 1 button lets you restart continuous flow operation. In this case the duration or target volume will be reset to their starting values. Left clicking on the *Continuous Flow* 2 button terminates continuous flow mode and stops the continuous flow operation in progress. In this case re-starting will not be possible (see image below).

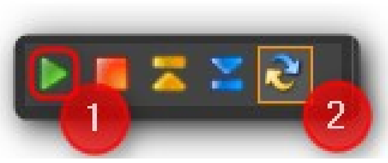


Figure 26: Canceling or Restarting Continuous Flow

7.14.14 Change flow rate in continuous flow

You can change the flow rate during a dosing run in all operating modes. To achieve this enter the desired flow rate into the *Flow* field and confirm by clicking on *Start* with the left mouse button (see image below).

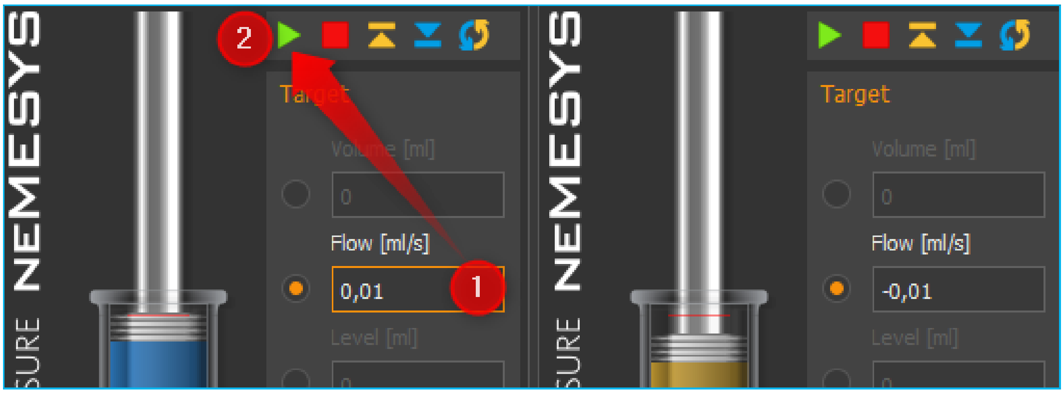


Figure 27: Changing the Flow Rate

The dosing module accepts the change in flow rate, while retaining all other dosing parameters, such as volume or continuous flow parameters. When you configure continuous flow operation you will be shown the maximum dosing flow rate achievable with the currently set refill rate and the other continuous flow parameters, for guidance (see image below).

7.15 Nemesys Script Functions

7.15.1 Introduction

The Nemesys plugin offers a number of script functions which can be used to program automatic sequences. The following script functions are available:

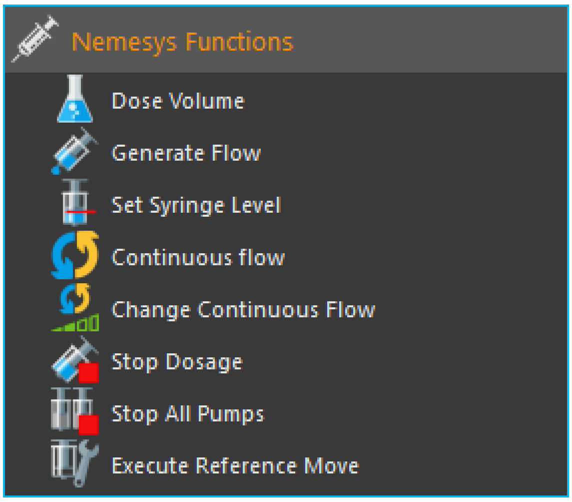


Figure 177: Nemesys script functions

7.15.2 Automatically stop pumps at script stop

If you want all pumps to be stopped immediately when the user stops the running script program, activate the corresponding button in the main toolbar.

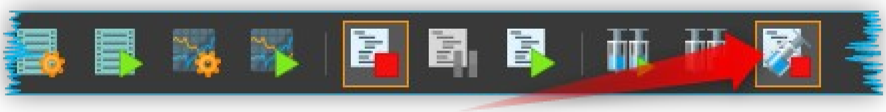
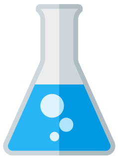


Figure 178: Automatically stop pumps at script stop

7.15.3 Dose Volume



With this function, you can dose a specific volume at a precisely defined flow rate. In the selection field Dosing Module 1 you select the dosing module you want to use for dosing. Alternatively, you can also enter the name of a script variable in the field, which contains a device reference of a pump.

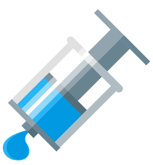
All other parameters, such as the volume to be dosed and the flow rate, can be set in the Target Values 2 area.

Figure 179: Dose Volume Script Parameter

You can also activate or deactivate the *Run to completion* 3 parameter in the configuration area. When *Run to completion* is activated, the script execution is not continued until the complete volume has been dosed and the dosing process has ended. If this parameter is not active, the dosing is started, and then the next script function is executed immediately. This enables you, for example, to start a number of dosing modules almost simultaneously.

HINT. All the dosing functions support the use of variables. That means, in all input fields marked with a coloured V in the script configuration panel (e.g. flow rate and volume) you can enter variables.

7.15.4 Generate Flow



This function is used to generate a constant flow rate. In the configuration area, you can select the dosing module and set the flow rate. If the *Run to completion* parameter is active, the next script function is not executed until the module has stopped or reached one of the limit positions. The configuration of the parameters corresponds to the [Dose Volume](#) function.

HINT. You can use script variables with device references in the pump drop-down box.

7.15.5 Set Syringe Level

You can use this function in a script if you want to reach a specific syringe fill level. The dosing module then doses until the target fill level is reached. You can set the dosing module, the fill level and the flow rate in the configuration area of this function. The configuration of the parameters corresponds to the [Dose Volume](#) function.

7.15.6 Continuous Flow

This function allows you to start a continuous flow of two pumps from a script. To do this, select the first dosing unit ❶ in the script configuration area of this function and then start the *Continuous Flow Wizard* ❷ to configure all parameters. For a detailed description of the configuration, refer to the [Configuring Continuous Flow](#) section.

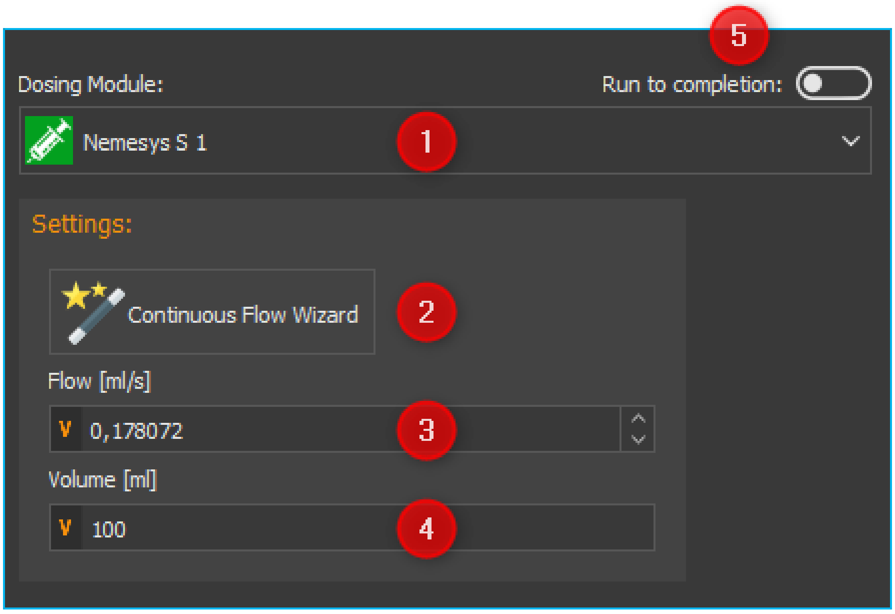


Figure 180: Configuration Continuous Flow Script Function

After configuration with the *Continuous Flow Wizard*, you will see the configured flow rate in the *Flow* field ❸. If you have configured a certain volume for the continuous flow in the wizard, the *Volume* field

4 is also displayed. You can use the *Flow* and *Volume* fields to set the flow rate and volume using script variables.

With the checkbox *Run to completion* 5 you define when the next script function will be executed. If no check mark is set here, the continuous flow is started and the next script function is executed immediately. If *Run to completion* is active, the script is only continued when the abort condition configured in the wizard occurs - i.e. when the configured volume has been dosed or the set time has elapsed.

7.15.7 Change Continuous Flow

With this function you can start a previously configured continuous flow or change an already running continuous flow. To execute this function without errors, you must either have previously configured a continuous flow for two pumps or have executed the Continuous Flow script function.

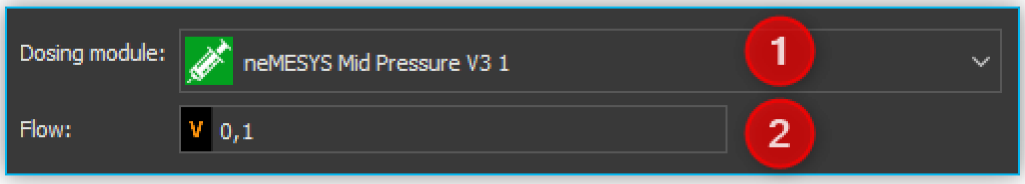
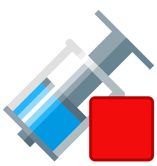


Figure 181: Configuration Change Continuous Flow Script Function

In the configuration area, select the pump 1 whose continuous flow you want to change. In the Flow field 2 enter the flow rate. The unit corresponds to the unit configured in the Continuous Flow Wizard when configuring the continuous flow.

HINT. You can use script variables to set the flow rate.

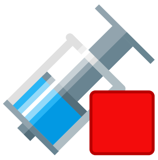
7.15.8 Stop Dosage



You can stop an active dosing process of a module with this function.

HINT. You can use script variables with device references in the pump drop-down box.

7.15.9 Stop All Pumps



Stops the dosing of all pumps simultaneously.

7.15.10 Execute Reference Move

This function allows you to start a reference move from the script. With the parameter *Run To Completion* you can specify whether the function is terminated after starting the reference move or after completion of reference move.

8 Qmix I/O Plugin

8.1 Introduction

The Qmix I/O plugin is used to integrate various Qmix I/O modules into the QmixElements software and for displaying I/O channels of other devices like neMAXYS positioning systems or neMESYS syringe pumps.

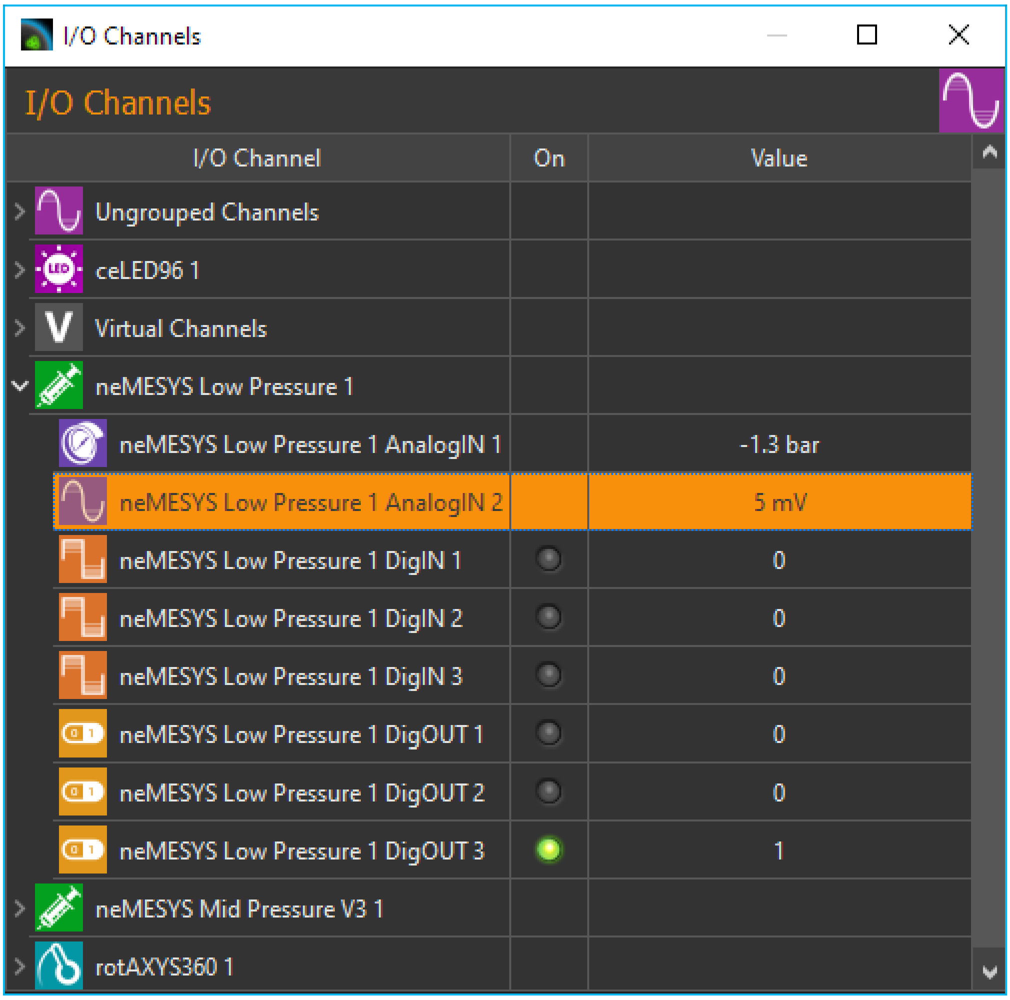
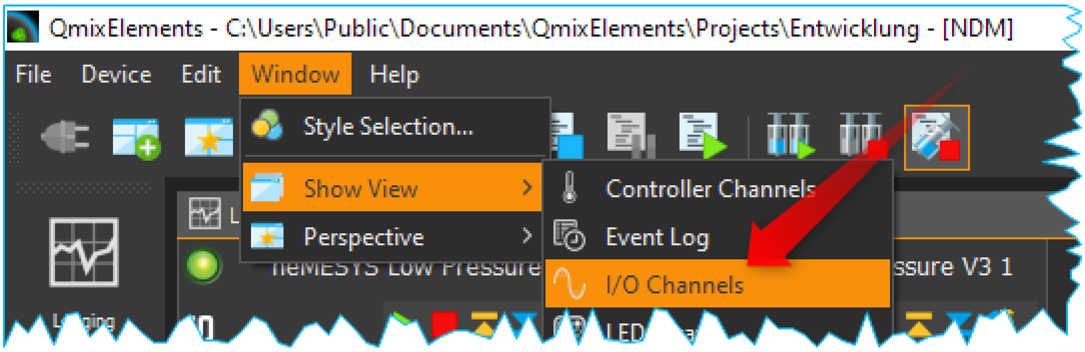


Figure 182: List of I/O channels

8.2 List of I/O channels

All available Qmix I/O channels are shown in the list of I/O channels. You can distinguish the different types of I/Os (AI: analogue input, AO: analogue output, DI: digital input, DO: digital output, Pressure: pressure sensor) by their symbols and names (see figure above). If the I/O list is not visible, you can display the window via the main menu of the application:



The following columns are present:

- *I/O Channel* – contains the name of the I/O channel and displays a symbol for the channel type.
- *On* – a lit green LED indicates that a channel is switched on and that a digital channel is 1 (instead of 0), respectively.
- *Actual Value* - Shows the current value of the channel - in the case of output channels, this is the value that is output, and in the case of input channels the value read from the device.

8.2.1 Channel types

The following types of channel are currently supported:

analog inputs (voltage and current)

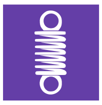
analog outputs

digital inputs

digital outputs

analog pressure sensor inputs

analog temperature sensor inputs



analog force sensor

analog flow sensor

virtual channels

8.2.2 Grouped display

By default, the I/O channels are displayed in groups. This means that the channels of a certain device are grouped under the device name, so that a tree-like structure is created. I.e., you can show or hide the display of channels for certain devices, e.g. for the neMESYS pumps. In the following figure, for example, only the channels of the first neMESYS pump are displayed.

IMPORTANT. If you are still working with an older device configuration, you may have to save the device configuration in the Device Configurator again so that the I/O channels are displayed in groups.

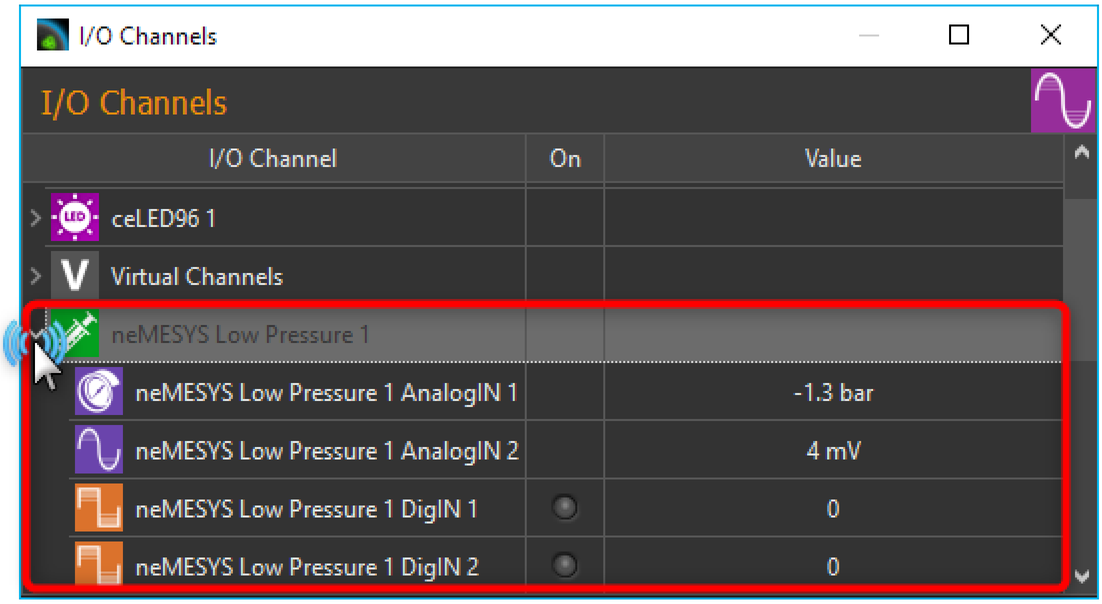


Figure 183: Grouping of I/O channels

All I/O channels that do not belong to a specific device or channels of devices that do not yet support grouping are grouped together in the *Ungrouped Channels* group.

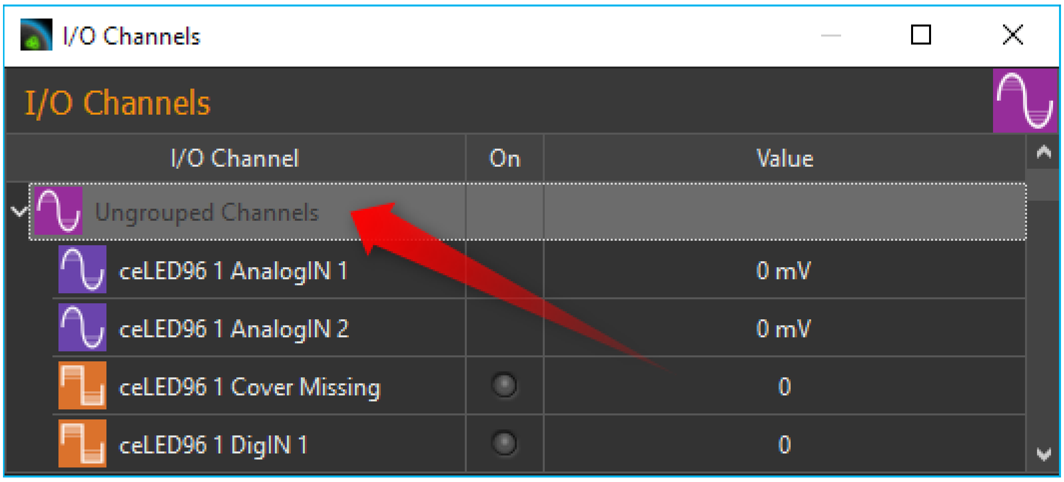
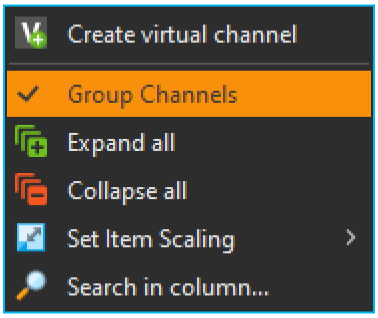


Figure 184: Ungrouped Channels

You can activate and deactivate the grouping of channels at any time. Simply right-click in the list of I/O channels and select *Group Channels* from the context menu (figure below).



If the channel grouping is deactivated, you get a flat display of the I/O channels in list form.

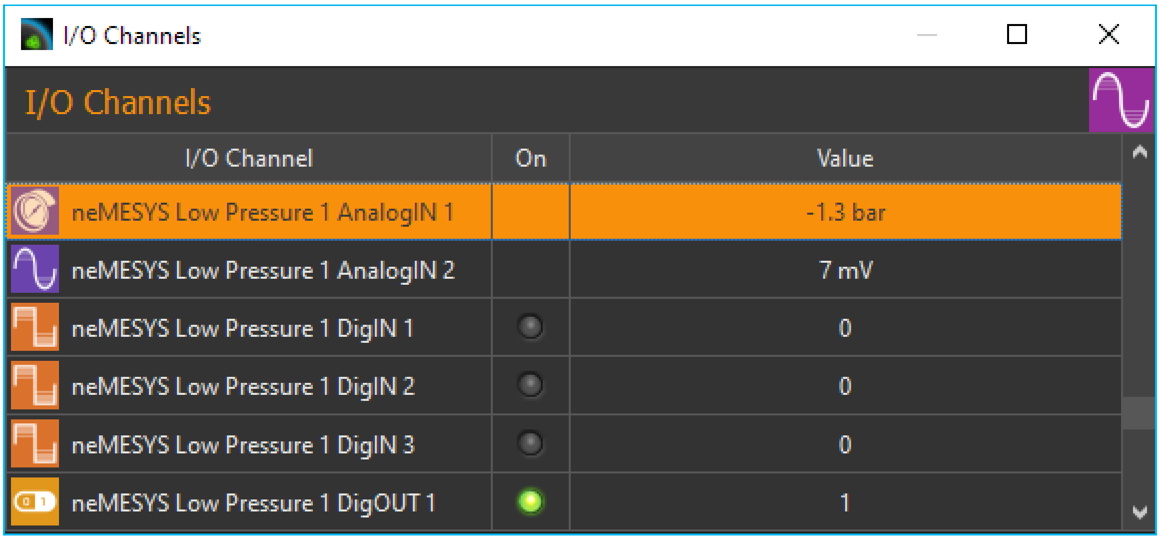
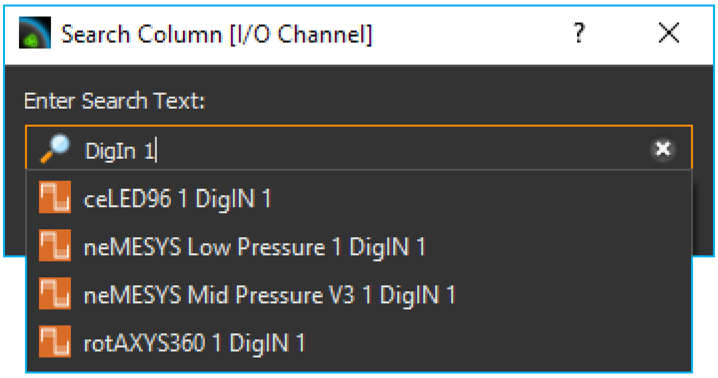


Figure 185: Display of I/O channels without grouping.

8.2.3 Search I/O channels

If you want to quickly search for a specific channel in the list, right-click in the first column (*I/O Channel*) and select *Search in column* from the context menu.



A search dialog is displayed in which you can enter the search term. A list of possible hits is displayed as you enter the search term. If you select an entry, the corresponding channel is displayed in the I/O list and is highlighted in color.

8.2.4 Change Item Scaling

In order to increase clarity or improve readability, the display of the list entries can be switched between three sizes. To do this, choose *Set Item Scaling* from the context menu and then select the required size:

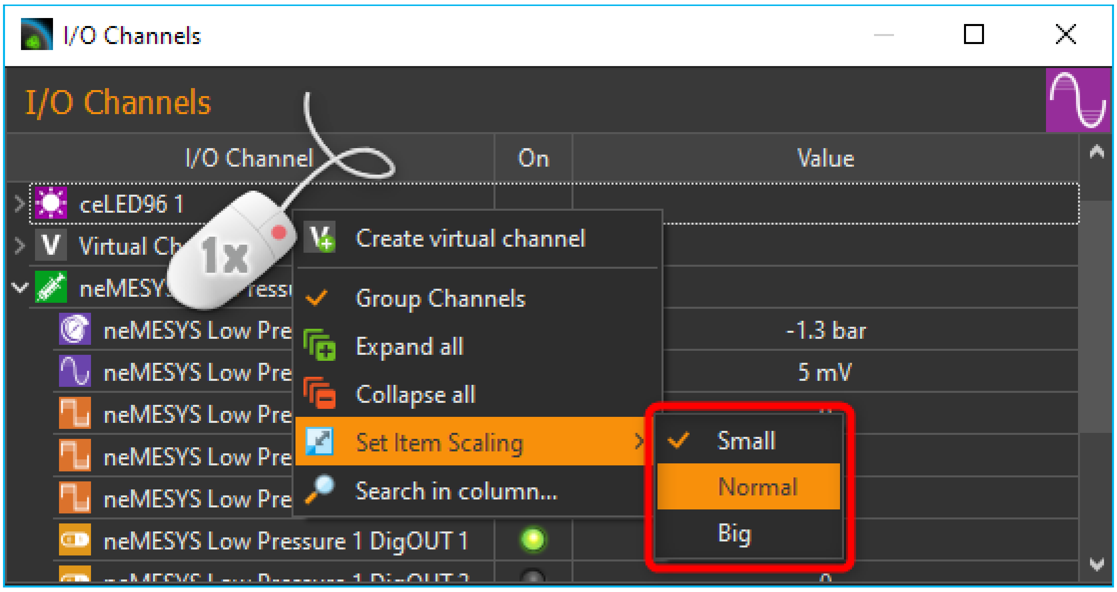


Figure 186: Changing the scaling of list items

8.3 Set outputs

The values of output channels can be changed by the user. Output channels include digital and analog output channels as well as virtual channels. Digital output channels can be switched on and off by clicking on the LED in the *On column* of the channel.

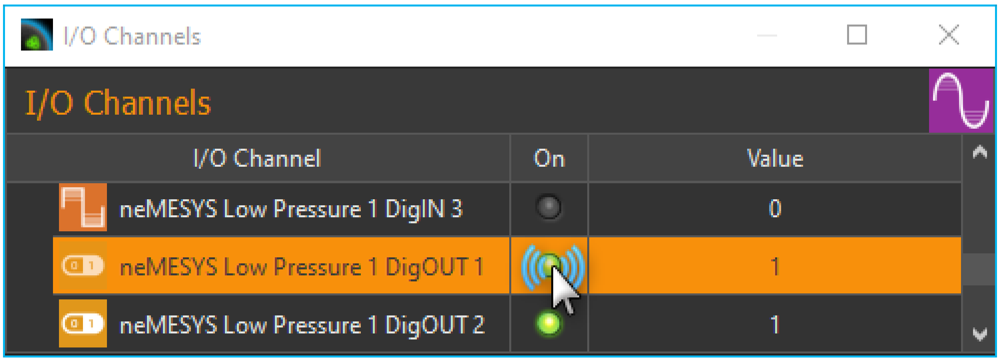


Figure 187: Switching digital output channels on and off

Analog channels can also be switched on and off by clicking on the LED. If an analog channel is switched off, the value 0 is output. If an analog channel is switched on, the value from the *Value column* is output. If you want to change the analog value, simply double-click with the left mouse button in the value column of a channel, or select the value column of the channel and then start typing on the keyboard.

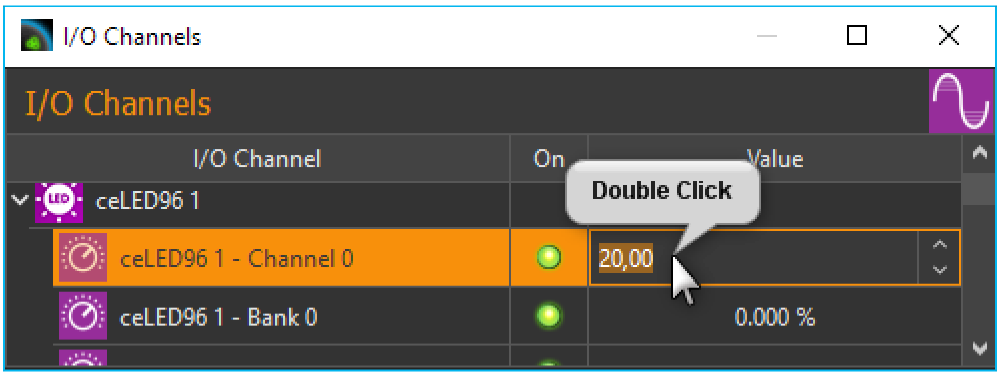


Figure 188: Changing values of analog output channels

This also allows you to change the values of virtual channels.

8.4 I/O Channel Configuration

8.4.1 Changing channel names

You can change the name of each channel at any time, for example, to assign a memorable name suitable for your particular application. You may change a name by the following steps:

1. **Double-click** the name you want to change.
2. The name is now highlighted in yellow: Enter the new name (figure below).
3. Complete your entry by pressing the *Return* key.

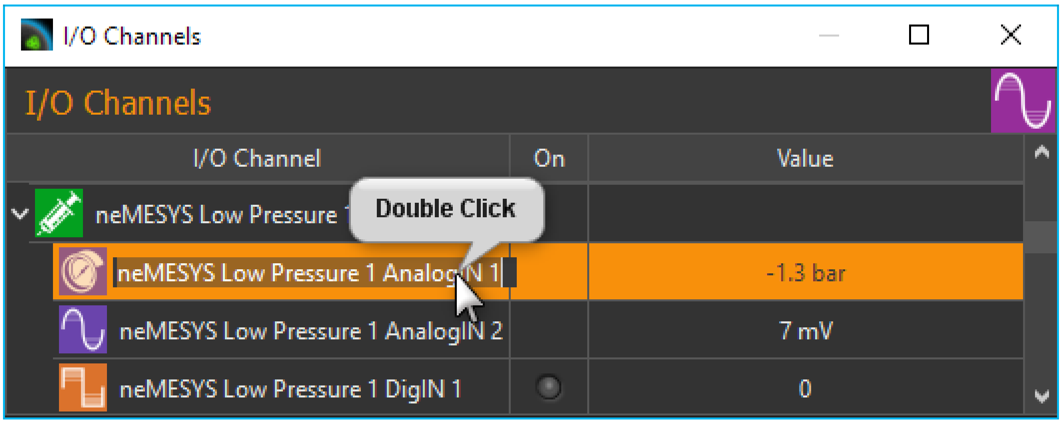


Figure 189: Changing a channel name

8.4.2 Activating the Configuration Dialog

In addition to customizing the name, for most I/O channels there are additional parameters that may be configured, such as the scaling of analog in- and outputs. You will find these configuration parameters in the respective configuration dialog of each channel.

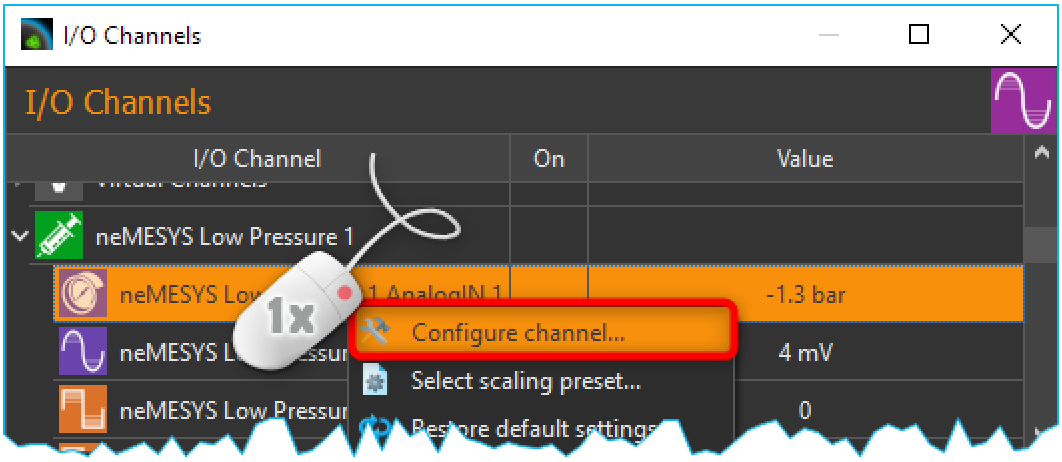


Figure 190: Opening the channel configuration panel

To open the configuration panel, right-click on the respective channel name from the I/O channel list and select the context menu item *Configure channel* (figure above).

8.4.3 Configuration Dialog

In the configuration dialog you can configure all accessible parameters of an analog in- and output channel:

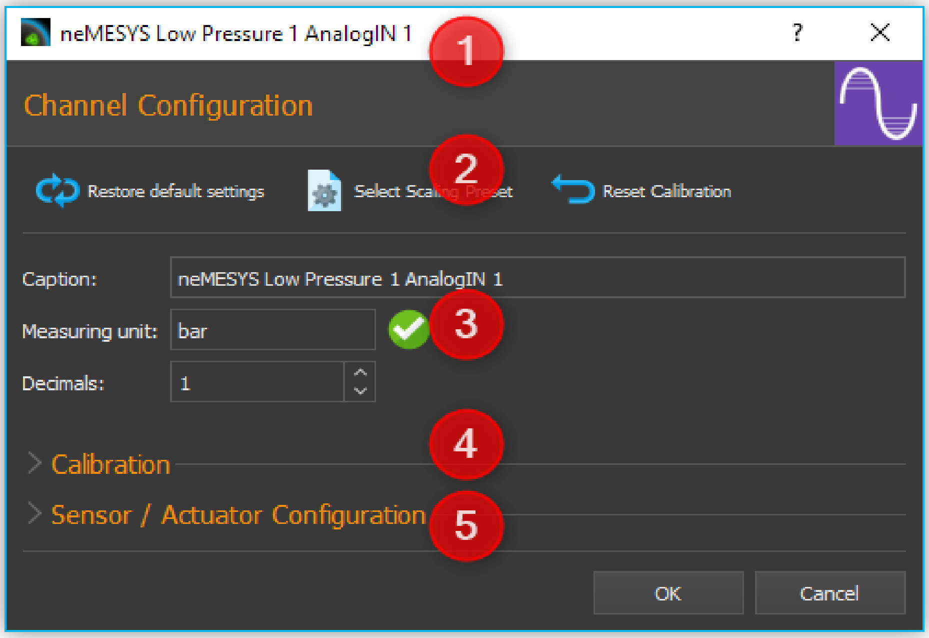


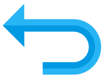
Figure 191: Configuration dialog for I/O channels

1 The title bar shows the name of the channel you are configuring.

2 In the toolbar you can find the actions you can perform:

Restore default settings – By clicking this button, all parameters (channel name, decimal points, scaling, *etc.*) are reset to their original values.

Select Scaling Preset - opens a selection dialog with predefined scaling settings.



Reset Calibration – resets the calibration of the channel to the default settings (offset = 0, factor = 1).

3 Here you can configure the general settings of the channel, such as name, decimal places or unit of measurement:

- *Caption* – Here you can set a new name for that channel, which is then displayed in the channel list.
- *Measuring Unit* – Enter the measurement unit in which the measured values are displayed. The unit must be compatible with the basic unit of the channel or sensor. For example, you can use the units **bar**, **psi**, or **Pa** for a pressure measurement channel, but not the units **mV**, **mA** or **kg**. The software automatically converts the measured values of the sensor into the selected unit.
- *Decimals* – This field is to set the number of decimal points of the scaled unit of measurement.

4 In the *Calibration* area, you can compensate measurement value deviations of channels by means of calibration.

5 For channels that support the connection of sensors (for example, voltage or current measuring inputs), you can display the operating elements for configuring the scaling of the sensor. To do this, click *Sensor / Actuator Configuration*.

HINT. All changes will be activated only after being accepted by pressing OK. To cancel all changes use Cancel.

IMPORTANT. Clicking Restore default settings will instantly reset all parameters to their original setting – clicking OK is not required.

8.5 Calibration

With the two-point calibration you can correct measurement deviations of a channel. To do this, you record the measured value of the channel at two points 1 and 2 (*Value*) and enter the actual corrected value of the channel for these points (*Calibrated value*). The software calculates the slope (factor) and the offset of the scaling.

If you click on the *Capture current channel value* 3 button, the current value of the channel is automatically entered into the *Value* field.

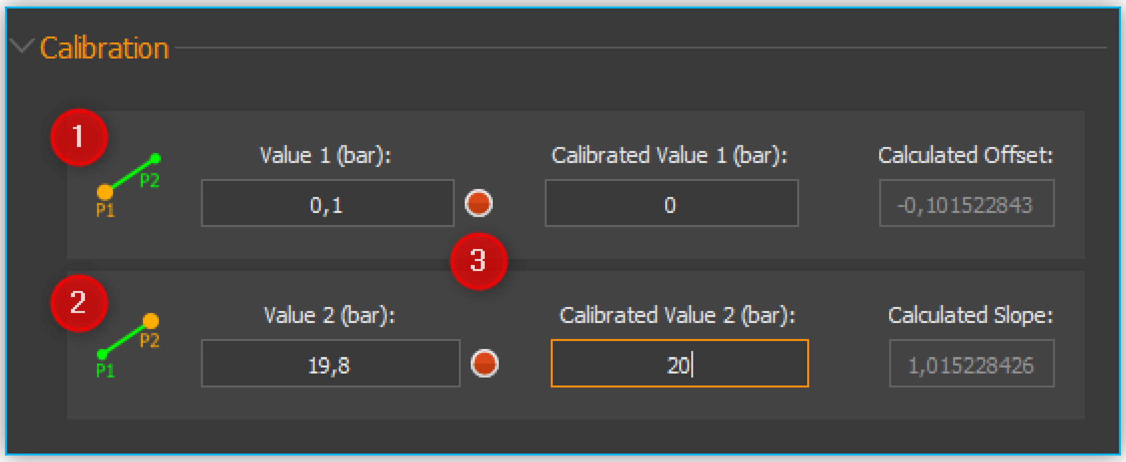


Figure 192: I/O Channel calibration

IMPORTANT. All calibration values are entered in the configured unit of measurement. If you change the unit of measurement or the scaling of the channel, you must check the calibration and adjust it if necessary.

8.6 Configure Sensor Scaling

8.6.1 Introduction

For analogue channels that support the connection of sensors (such as voltage or current measurement inputs), you can configure sensor-specific scaling. This allows you to connect sensors to the analog inputs and then scale the measured values to the value range of the analog sensors and display them in the correct unit. To do this, click on *Scaling Configuration* in the configuration dialog.

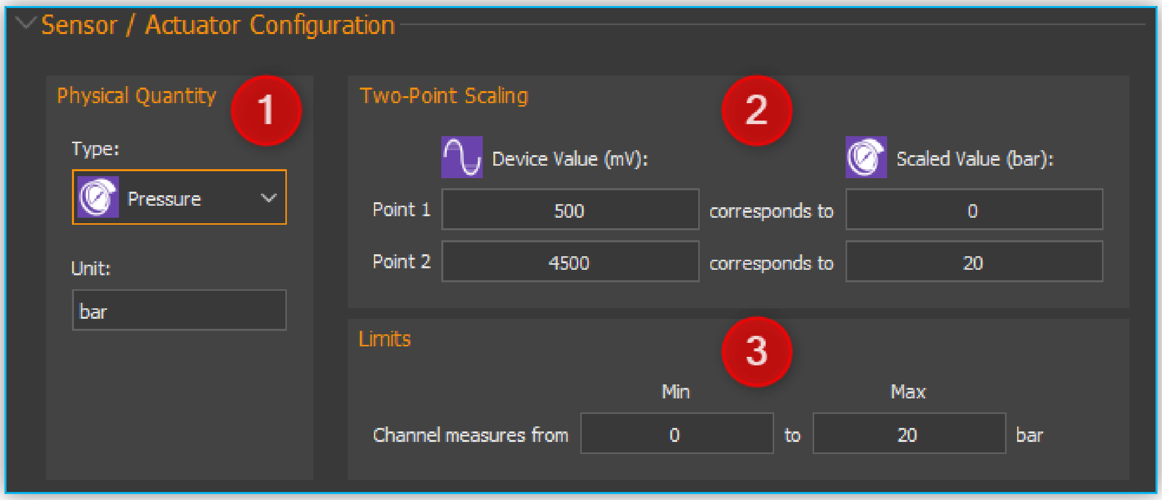


Figure 193: Configuration scaling parameters

8.6.2 Select Physical Quantity

In the *Physical Quantity* pane 1, select the physical quantity to be measured with the sensor. If the desired sensor type is not supported in the *Type* selection box, simply select the *Custom* type for a user-specific scaling. For a pressure sensor you simply select the *Pressure* type.

In the *Unit* input field, enter the basic unit of the sensor. When selecting a sensor type, a corresponding unit is already suggested to you. The unit must be compatible with the selected physical quantity (e.g., bar, psi, Pa or atm for pressure sensors).

8.6.3 Two-Point Scaling

The two-point scaling 2 is used to configure the conversion between the analog measured values of the I/O channel (for example in mV or mA) and the sensor readings of the sensor connected thereto (for example, bar or ° C). Normally you will find these values in the data sheet of the sensor.

IMPORTANT. Currently, the software only supports linear scaling of measurement values.

For example, if you have a pressure sensor with an analog input range from 0.5 V to 4.5 V and a measurement range from 0 bar to 20 bar, you can enter the following parameters. Enter the I/O channel range 500 mV and 4500 mV in the first column (*Device value*) and the sensor range 0 and 20 bar in the second column (*Scaled value*).

IMPORTANT. Make sure that the units used correspond to the units in the data sheet of the sensor.

8.6.4 Limits

In the Limits 3 pane you define the measuring range of the sensor. In many cases, this range matches the range that you entered in two-point scaling pane. Therefore, the values for the two-point scaling are automatically transferred to the range limits. Here, you can further restrict or change the value range.

IMPORTANT. Clicking **Restore default settings** will instantly reset all parameters to their original setting – clicking **OK** is not required.

8.6.5 User-specific Scaling

If you are using a sensor which is not yet supported in the *Type* selection box, simply select the sensor type *Custom* 1. In this case, the input field measuring unit 2 is grayed out, since automatic unit conversion is no longer possible.

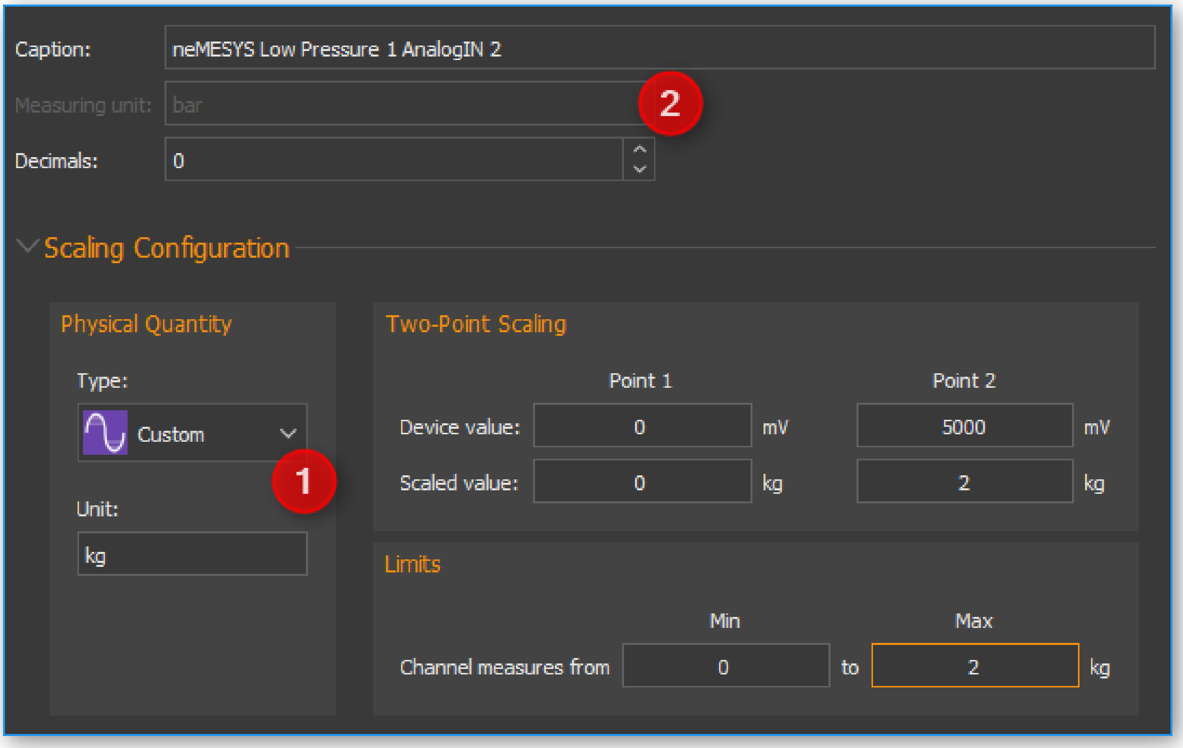


Figure 194: User-specific scaling

8.7 Scaling presets

Some analog input and output channels offer a choice of predefined configurations. These include pressure sensor configurations for the analog inputs of neMESYS syringe pump devices.

IMPORTANT. Predefined configurations are not offered by all analog channels. For those channels the menu item **Select scaling preset** is not displayed in the context menu.

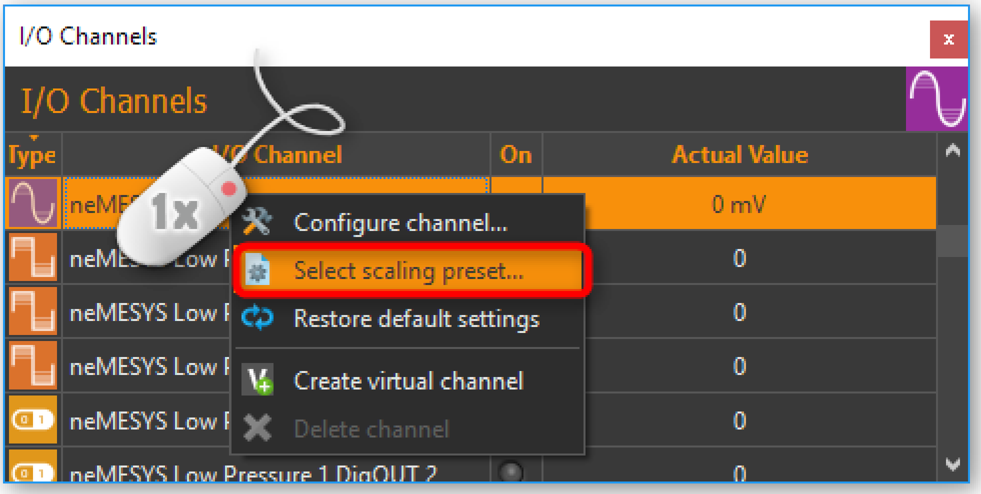


Figure 195: Opening predefined configurations dialog

To select a predefined configuration proceed as follows. Right-click the respective analog channel in order to show the context menu (see figure below). Choose *Select scaling preset*. A dialog containing a selection of scaling presets appears (figure below). Select the preset you are going to use and confirm your choice by left-clicking *Ok*.

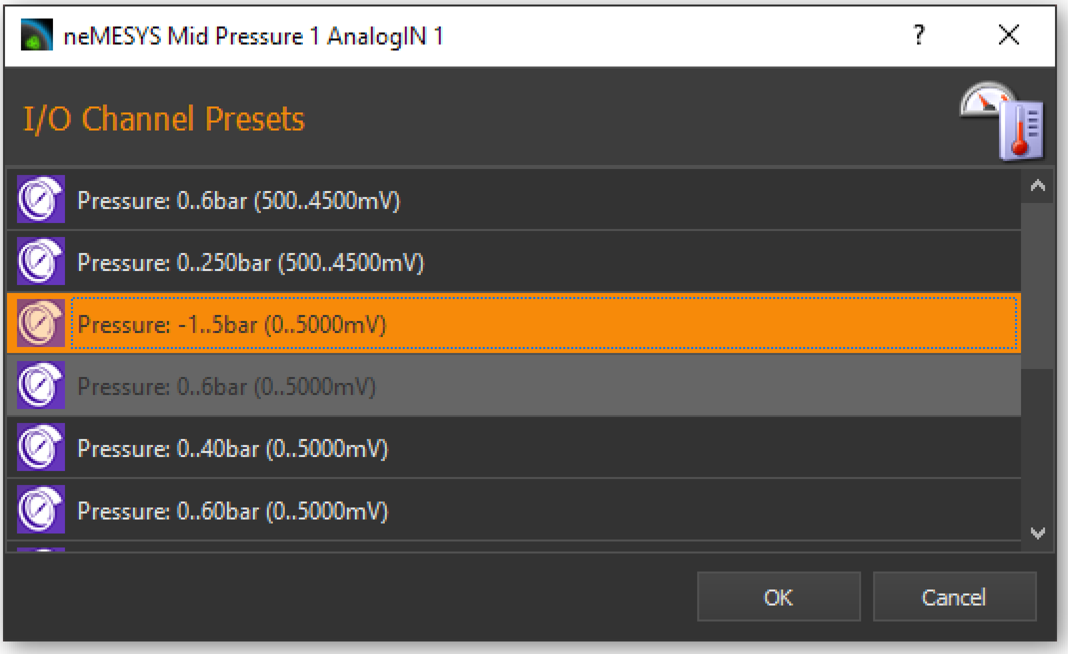


Figure 196: Selecting predefined configurations

To restore the default channel settings, simply click the menu item *Restore default settings* in the channel context menu (see figure below).

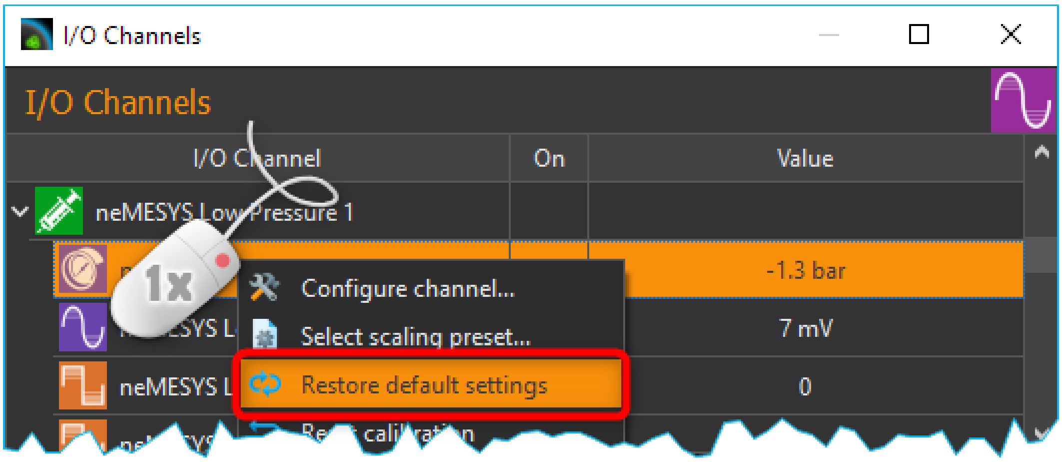


Figure 197: Reset channel scaling and settings

8.8 Virtual Channels

8.8.1 Creating virtual channels

The software allows the creation of virtual I/O channels. These channels are not assigned to a physical I/O device, but they are a kind of memory. You can write values into virtual channels and read them out later - just like a memory. Using these channels, you can, for example, show calculated values from a QmixElements script in the graphical logger. You only have to create the channel, add the channel to the logger, and then you can write values from the script into the channel and you will see them in the graphical logger.

To create a virtual channel, click with the right mouse button in the I/O channel list and select the context menu item *Create virtual channel*.

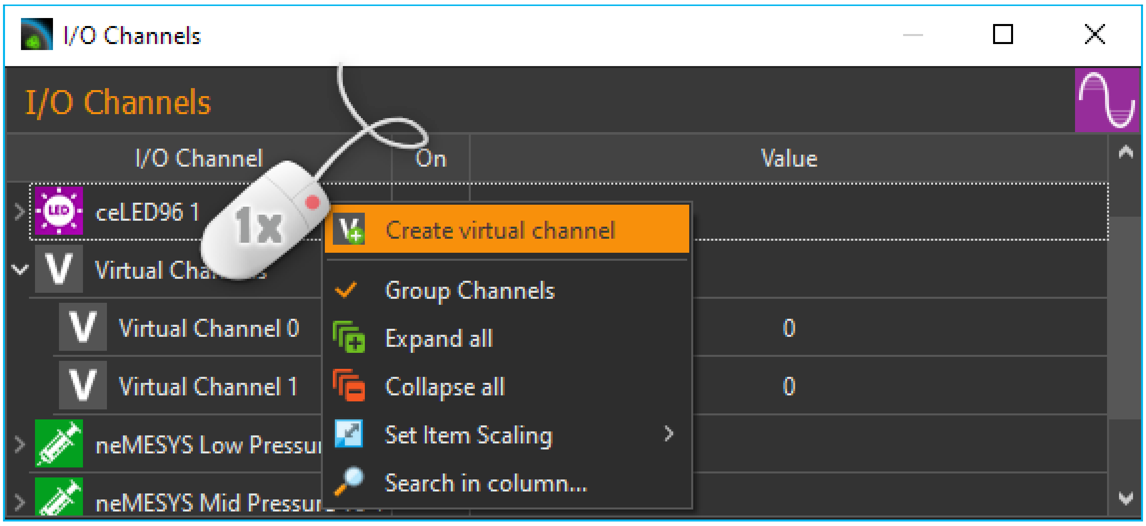


Figure 198: Creating virtual channels

A virtual channel will then be added and you can configure the channel further - e.g. change the channel name. If channel grouping is active, all virtual channels are grouped together in the *Virtual Channels* group and inserted there when they are created.

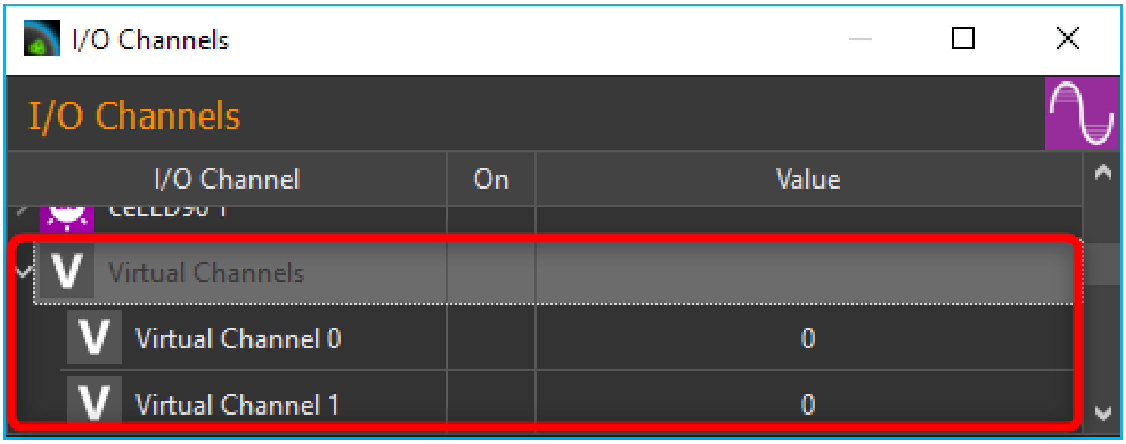


Figure 199: Virtual channels in the list of I/O channels

If channel grouping is disabled, the newly created virtual channel is inserted at the end of the list.

8.8.2 Access to virtual channels out of script programs

To access virtual channels out of QmixElements script programs (read and write access), you can use the script functions from the category Device Functions (figure below).

Figure 200: Read and write access to virtual channels

8.8.3 Deleting virtual channels

To delete a virtual channel, click with the right mouse button on the channel in the I/O channel list, and then select from the context menu the item *Delete virtual channel* (figure below).

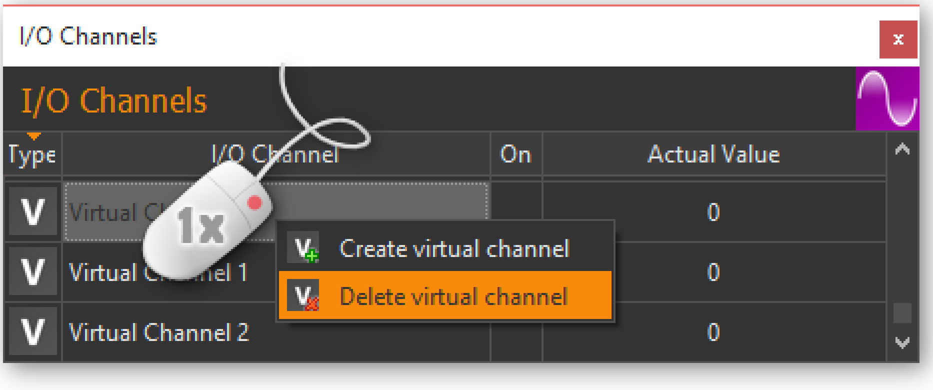


Figure 201: Deleting virtual channels

8.9 I/O Script Functions

8.9.1 Introduction

The Qmix I/O plugin contains script functions for switching digital outputs and setting the initial values of the analogue outputs.

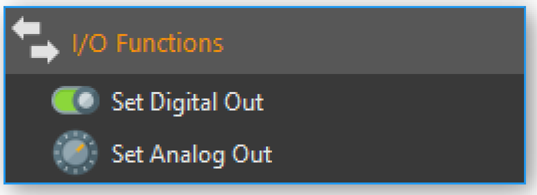
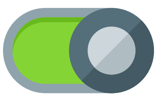


Figure 202: I/O script functions

8.9.2 Set Digital Out



You use this function to set/delete a digital output from a script. Select the digital channel in the configuration area of the function and then set the desired initial value.

8.9.3 Set Analog Out

With this function, you can write a value from a script to an analogue output channel. Select the analogue channel in the configuration area, and then configure the analogue initial value that is to be set during the subsequent execution of the function.

This function supports the use of variables. This means that, instead of a numeric value, you can insert a name of a variable into the field *Value*. This variable will then be set to the analog output value when the script is run (figure below). This variable may then subsequently be used for calculations or to carry out value-specific functions.

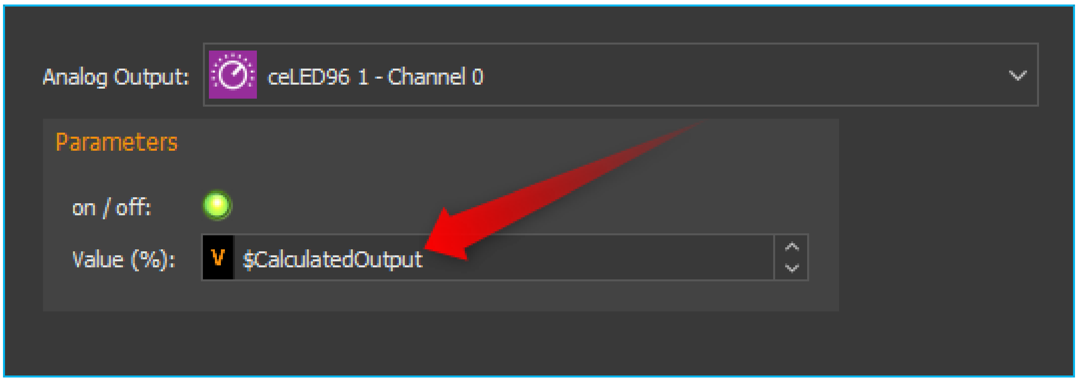


Figure 203: Analog output variable configuration

9 Qmix Controller Plugin

9.1 Introduction

The Qmix Controller plugin is used to integrate the Qmix controller modules into the QmixElements software. The following modules are supported:

- *Qmix Q-* - thermoelectric cooling module
- *Qmix Q+* - two-channel heating module
- *Qmix TC* – tow channel controller module for external heating- / cooling-systems (i.e. for connecting syringe heating)

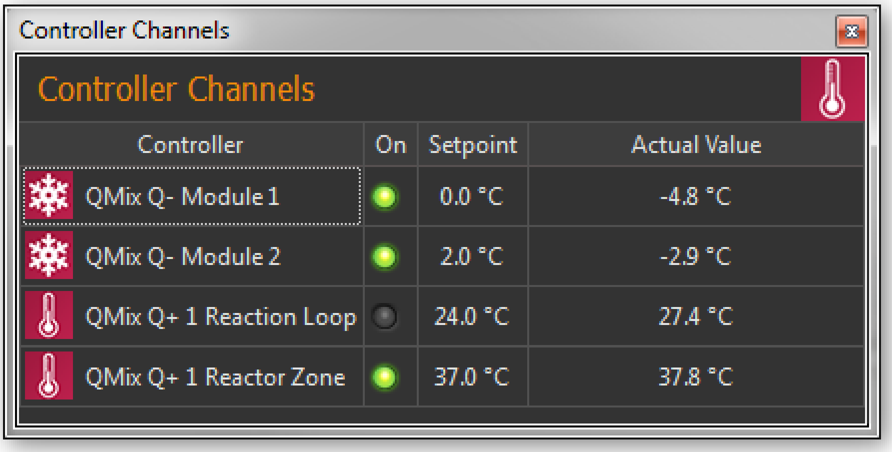


Figure 204: Controller Channel list

9.2 Controller Channel List

All Qmix controller channels are shown in the Controller Channels list. You can identify the different types of channels (cooling module Q-, heating module Q+...) by the different signs in front of the module names (see Figure above). The list of controller channels is in a tool window, that you can move freely at any time to another position in the graphic interface by dragging and dropping the title bar, or you can move it out of the interface to become a separate window.

The channel list shows you all the available Qmix controller channels in tabular form. The following columns are present:

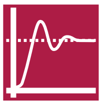
- *Controller* - shows the name of the controller module and indicates its type by a sign.
- *On* - indicates by a green LED whether the controller is switched on or off. Click the LED to switch the control loop on / off.
- *Setpoint* -contains the set setpoint of the controller channel
- *Actual Value* - shows the actual value

9.2.1 Channel types

The following types of channels are currently supported:

Snowflake icon: Qmix Q- cooling modules

Thermometer icon: Qmix Q+ heating modules



Control curve icon: dynamically generated, custom channels

9.2.2 Changing channel names

You can change the name of a channel at any time and, for example, assign a name suitable for your particular application.

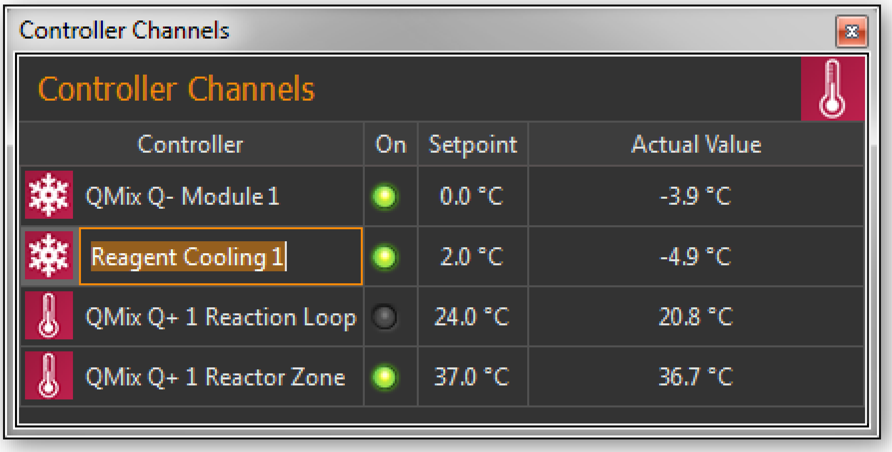


Figure 205: Changing channel names

You change a name by the following steps:

1. Double-click the table cell containing the name you want to change.
2. Enter the new name in the Editing window which now appears (Figure above).
3. Complete your entry by pressing the *Return* key.

9.2.3 Switching control devices on / off

To switch the controller on or off, simply click the LED of the channel you want to switch.

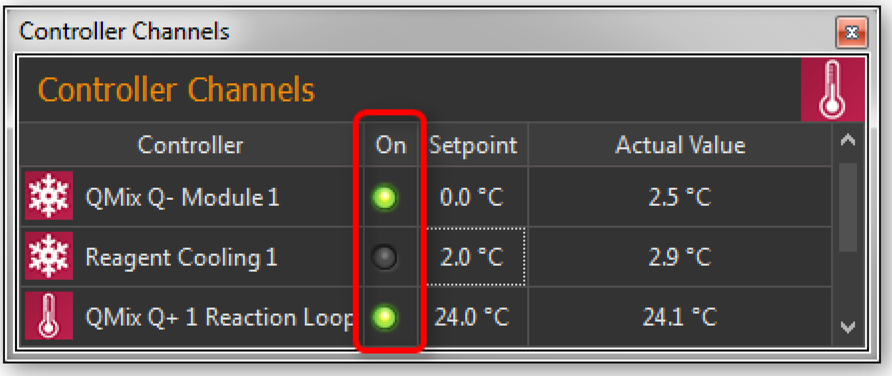


Figure 206: Switching control devices on/off

9.2.4 Entering setpoint

To input a setpoint, double-click in the Setpoint column of the channel with the setpoint you want to change. Now enter the setpoint in the Editing window that opens (Figure below) or use the arrow buttons to raise or lower the setpoint incrementally.

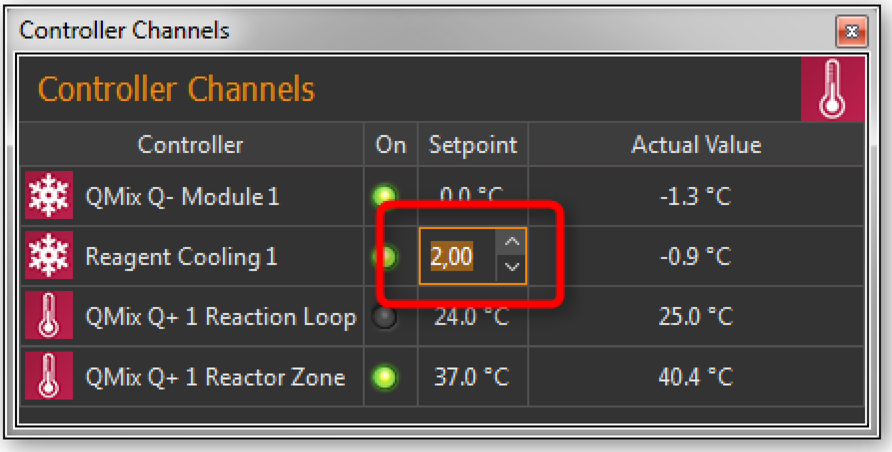


Figure 207: Changing setpoint

9.3 Context menu for control channels

By right-klick into the control channel list you can display a context menu with additional functions.

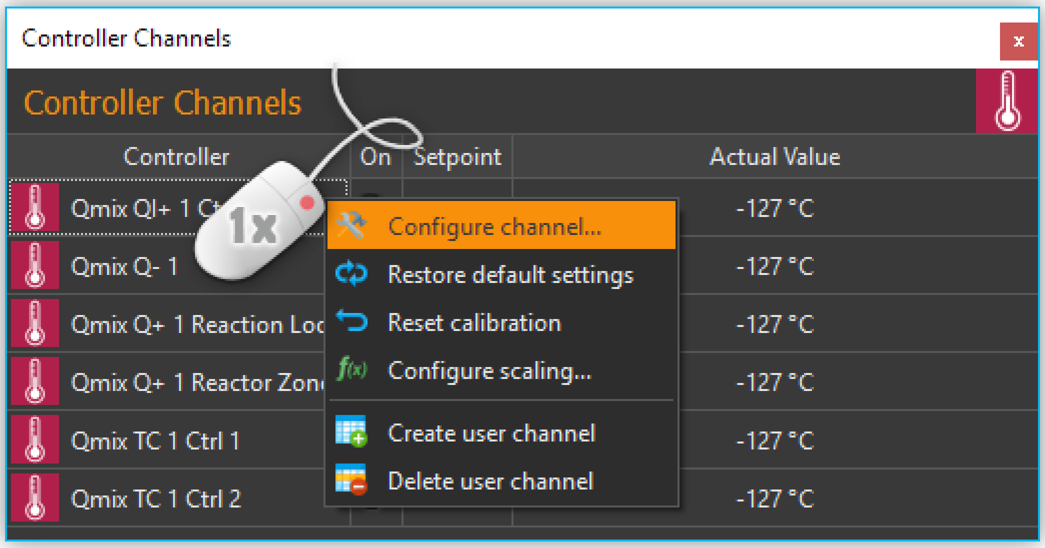


Figure 208: Opening the context menu

The context menu contains the following menu items:

Configure Channel...

Opens the configuration dialog of the channel for configuring all channel parameters.

Restore Default Settings

Restores the default settings of the channel.

Reset Calibration

Resets the two-point scaling to scale factor 1 and offset 0

Configure scaling...

Opens the configuration dialog displaying the page for configuration of the controller scaling.

Select PID parameters...

Opens the configuration dialog displaying the page for configuration of the PID control parameters.

Delete user channel

If the selected channel is a user-specified channel, it is deleted by selecting this menu item. For other channels this menu item is disabled.

Create user channel

Opens up the wizard for creating user-specified control channels.

9.4 Configure Channel Settings

For opening the scaling configuration dialog select *Configure channel* in the context menu of the control channel. (see section 9.3)

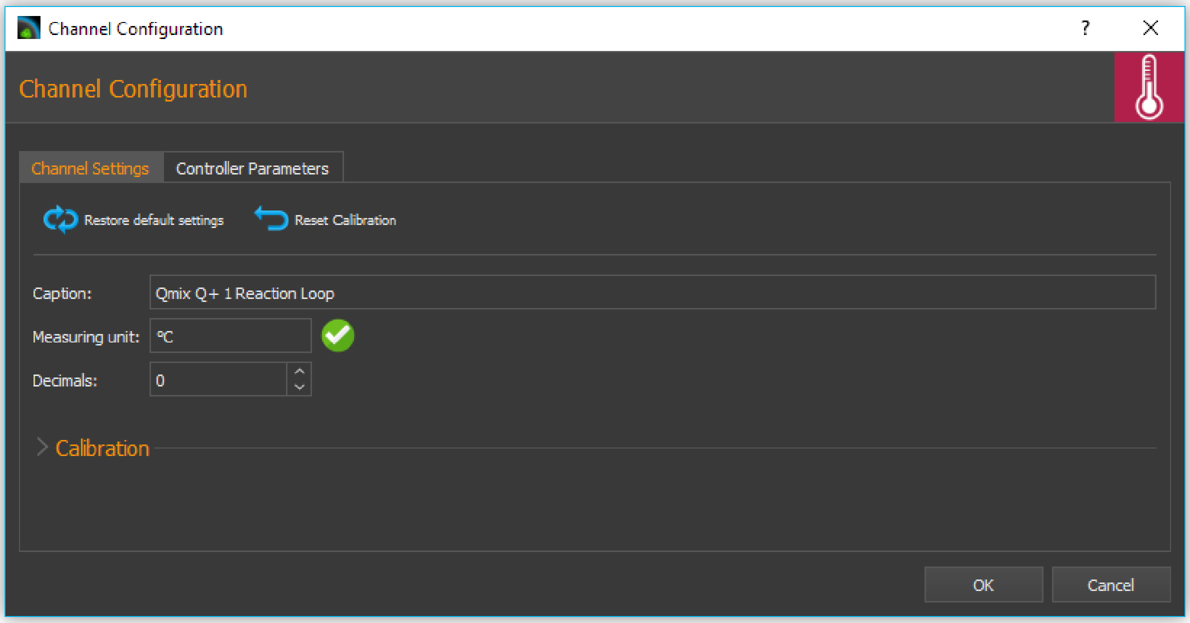


Figure 209: Configure scaling

The configuration of the general channel settings and the calibration of the controller channels are identical to the configuration of the I/O channels of the Qmix I/O Plugin. For a detailed description read the section I/O Channel Configuration.

9.5 Selection and configuration of controller parameters

9.5.1 Overview

To set the optimal control behavior, you can adjust the controller parameters of each single channel. For this you can either choose from a list of predefined PID parameter sets or create new parameter sets.

To access the controller parameters selection, choose *Select PID parameters* from the controller channel context menu.

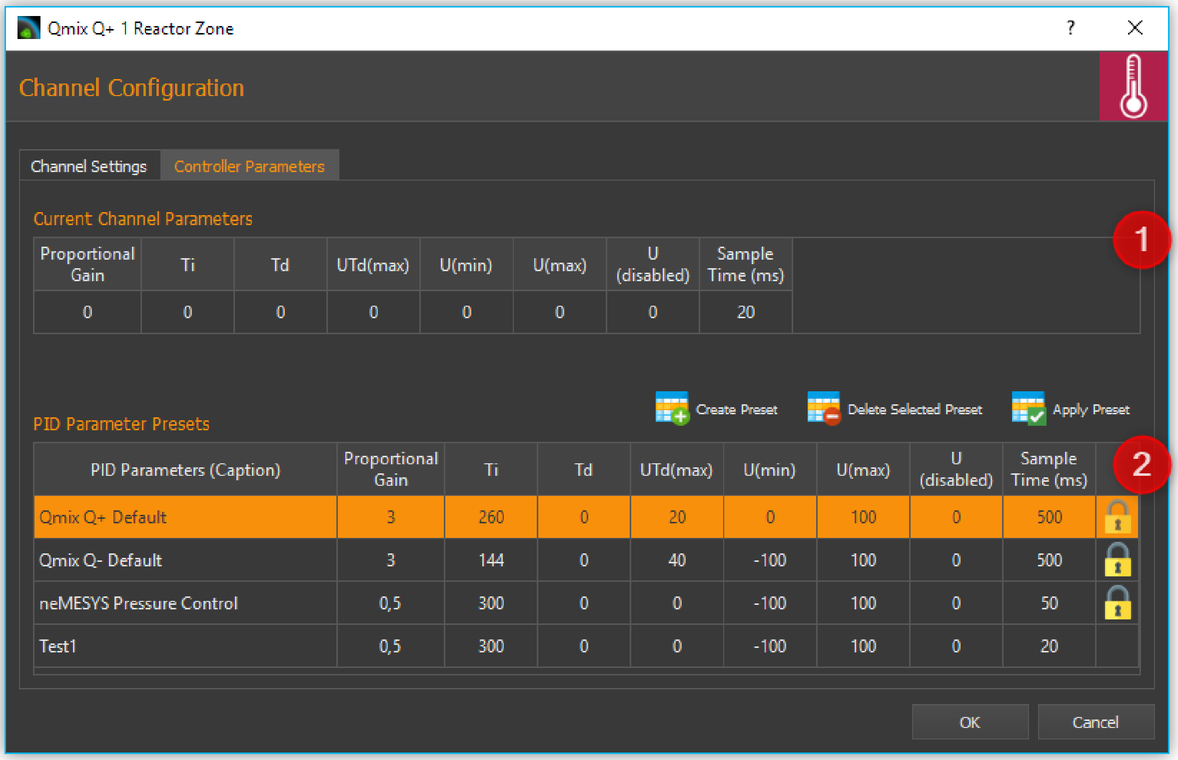


Figure 210: Selecting a control parameter set

The upper area shows the current parameters of the device 1 (*Current Channel Parameters*). In the lower area you will find a list of *PID Parameter Presets* 2.

9.5.2 Changing controller parameters

To edit the current controller parameters, double-click with the mouse in the *Current Channel Parameters* area in the field you want to change and enter the new value:

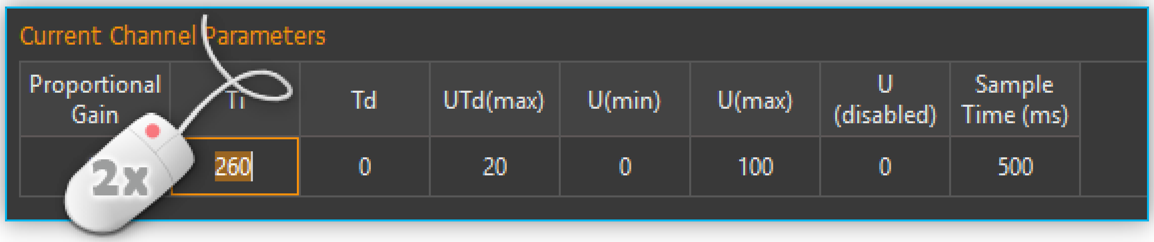


Figure 211: Changing controller parameters

9.5.3 Selecting a PID Parameter Preset

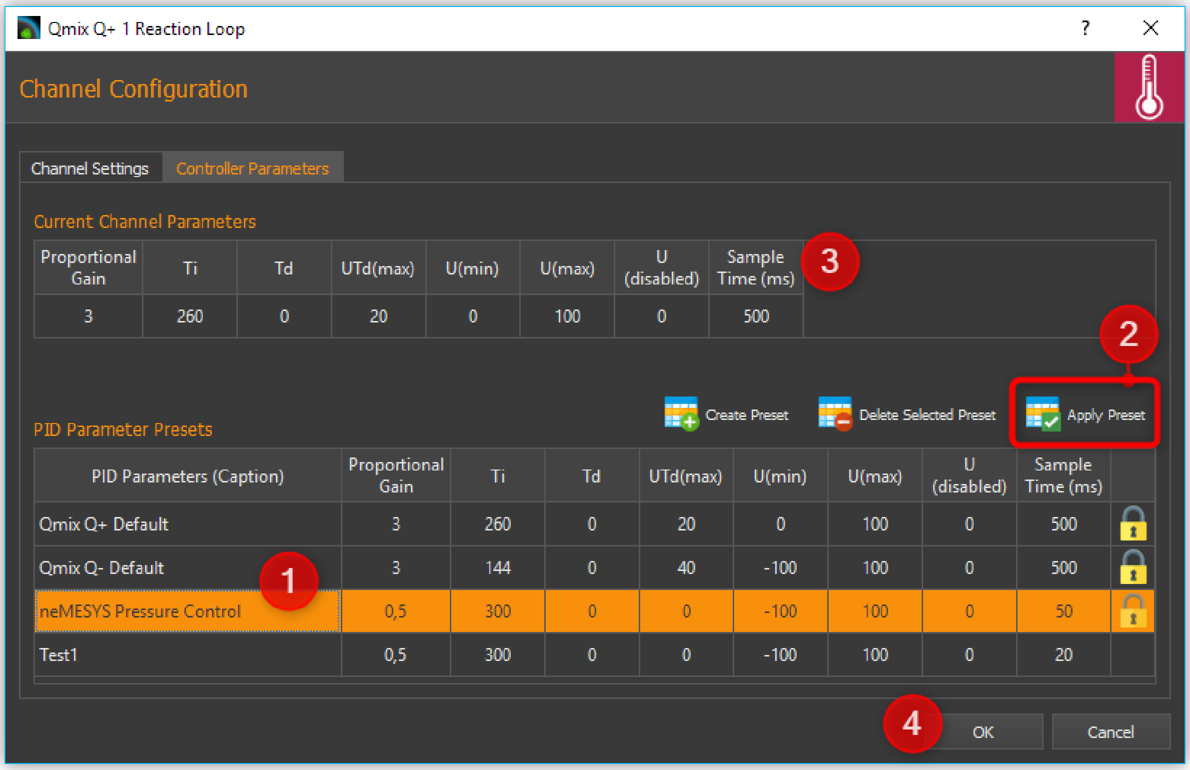


Figure 212: Selecting a PID Parameter Preset

Select a PID Preset from the table of available presets by left-clicking on it 1 and then click on the *Apply Preset* 2 button to apply the control parameters. The values in the *Current Channel Parameters* 3 area are updated with the new values from the preset. Complete the configuration by clicking the *Ok* 4 button.

The PID parameter presets already contain default controller parameter sets for different Qmix devices like Qmix Q+, Qmix Q- or Qmix TC or certain accessories, i.e. syringe heating or tube heating..

HINT. For optimum adaptation to the controlled systems in your application, you can create your own parameter sets with controller parameters.

9.5.4 Creating a PID Parameter Preset

By selecting the *Create Preset* button 1 you create a new set of controller parameters 2. (see figure below) You can then edit the individual values of the parameter set. Then you can edit the individual values of the parameter set by double-clicking in a field 3 and entering a new value.

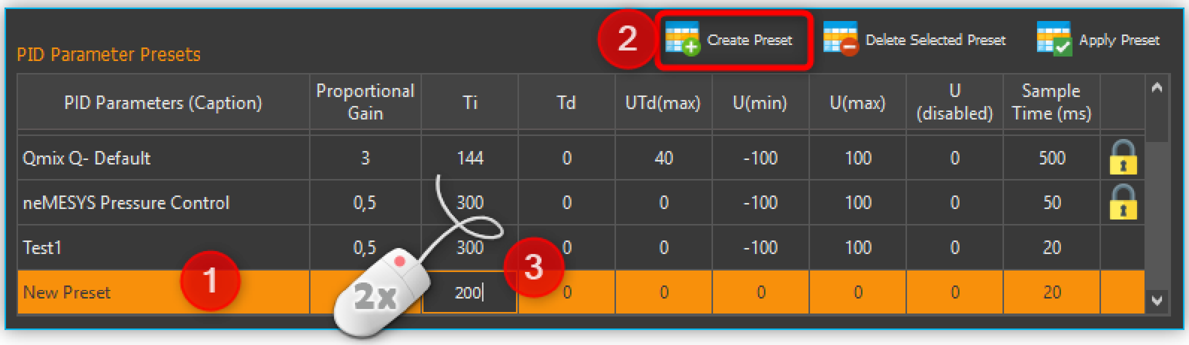
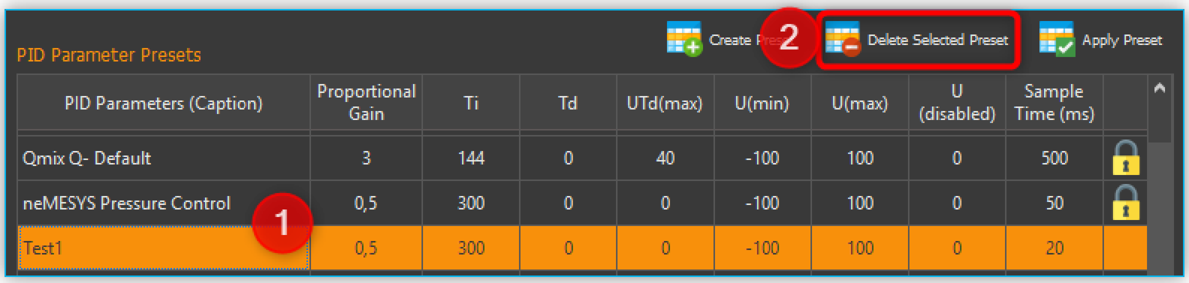


Figure 213: Creating a PID Parameter Preset

Enter a meaningful, unique name for each parameter set and adapt the controller parameters to the controlled system in your application. For finding adequate controller parameters proceed as described in section Procedure for setting controller parameters.

9.5.5 Deleting PID Parameter Presets

Select a set of controller parameters from the table 1 and left-click the *Delete Selected Preset* button 2 for deleting it. (see figure below).



PID Parameters (Caption)	Proportional Gain	Ti	Td	UTd(max)	U(min)	U(max)	U (disabled)	Sample Time (ms)
Qmix Q- Default	3	144	0	40	-100	100	0	500
neMESYS Pressure Control	0,5	300	0	0	-100	100	0	50
Test1	0,5	300	0	0	-100	100	0	20

Figure 214: Deleting a PID parameter set

IMPORTANT. Predefined controller parameters are locked and can not be deleted. Locked parameters can be recognized in the table by the lock symbol.

9.6 User defined control channels

9.6.1 Introduction

You can create customized control channels using arbitrary device properties for control loop input and output. Thus using i.e. a pressure measurement channel of a Qmix p device and a neMESYS syringe pump you can build up a pressure control.

9.6.2 Creating control channels

By right-clicking the control channel list and selecting the button *Create user channel* you open the dialog for creating control channels.

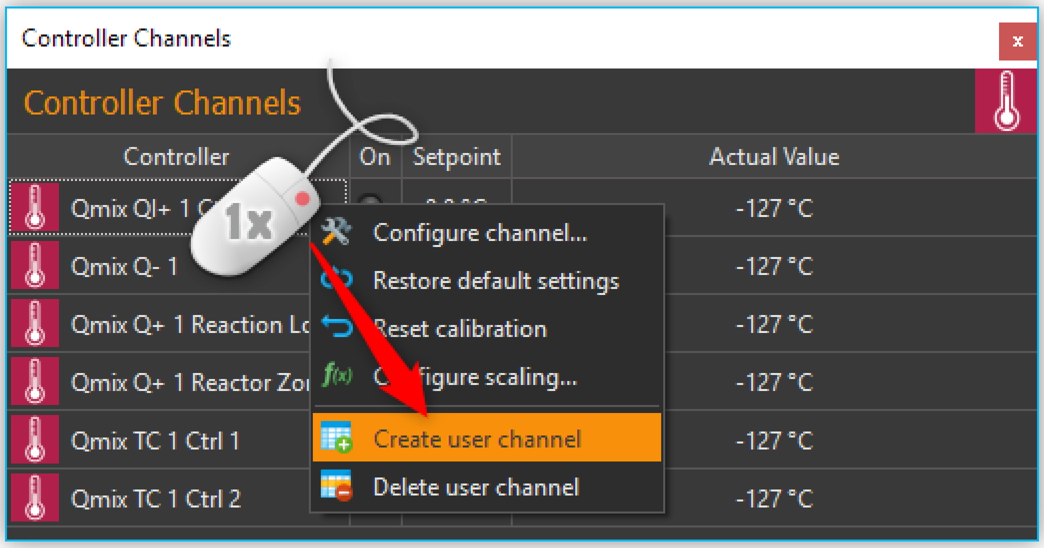


Figure 215: Open controller channel creation dialog

In the configuration wizard that is displayed now, proceed as follows:

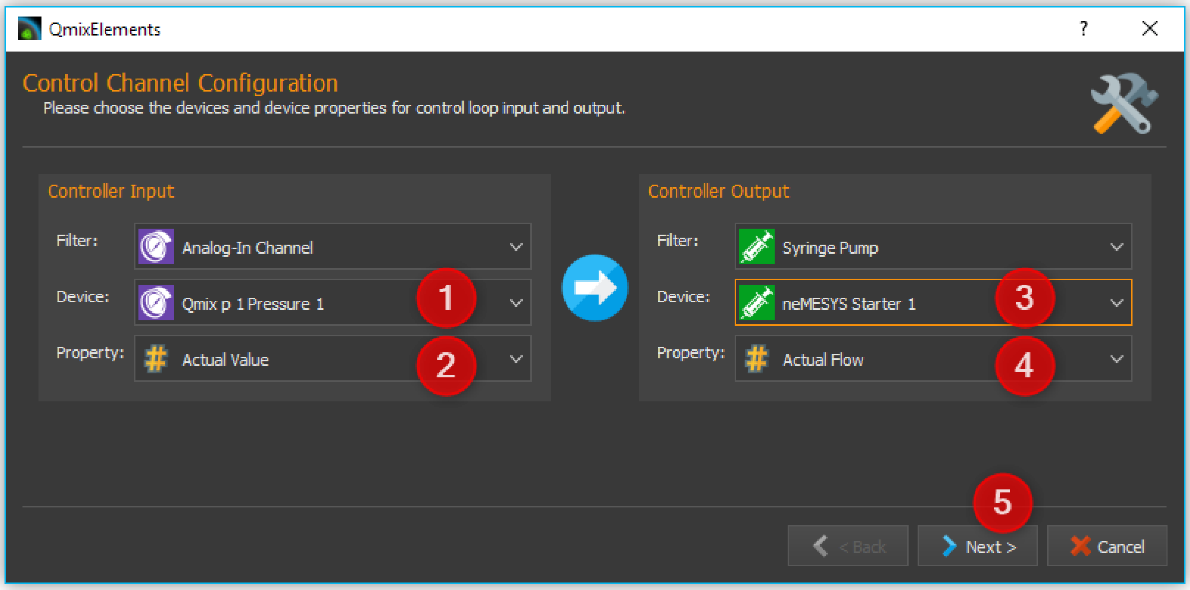


Figure 216: Selecting input and output values of the controller channel

1 Select the device that provides the measurement value (controller input) of the controller.

2 Select the device property that is used as measurement value.

3 Select the device that provides the control value (controller output) of the controller.

4 Select the device property that is used as control value.

5 Click *Next* in order to proceed.

The final page of the wizard allows configuration of controller channel parameters as described in section Selection and configuration of controller parameters. You complete the control channel creation by clicking the *Finish* button.

HINT. You can change the controller parameters at any later time and adapt them perfectly to your controlled system.

9.6.3 Changing the output value scaling or unit

The output value is determined by the PID control algorithm and written to the output device without any scaling information. This means that you need to adjust the controller parameters, in particular the control value limits, each time you change the scaling or the unit of the appropriate device. With a neMESYS dosing module this is also true if you change the syringe size.

IMPORTANT. If you change the scaling or unit of a device beeing part of a control loop you must check the control parameters and adjust them if necessary. With a neMESYS dosing module this also applies if you change the syringe.

9.7 Controller Script Functions

9.7.1 Introduction

The Qmix controller plugin contains a script function for changing the controller parameters from a script. This makes it possible to achieve, for example, time controlled temperature curves.

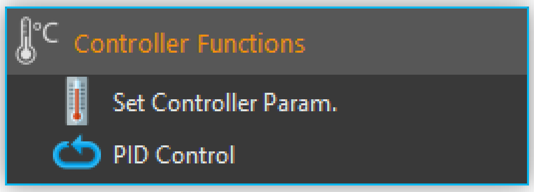
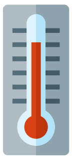


Figure 217: Qmix controller script functions

9.7.2 Set Controller Param



With this function, you can transfer a new setpoint to the controller channel or switch the control loop on or off. To switch it on and off, simply click the *Control loop on / off* LED in the configuration area (see figure below).

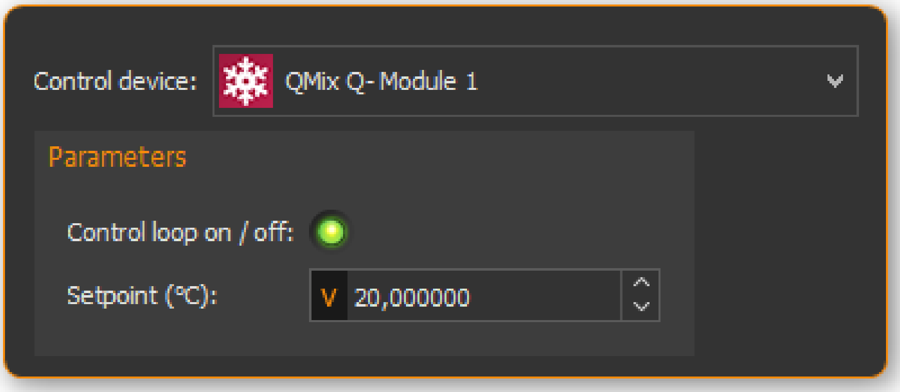
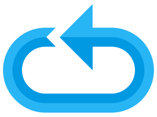


Figure 218: Configuration of the controller script function

9.7.3 PID Control Function

9.7.3.1 INTRODUCTION



This function implements a PID controller using a PID algorithm for applications that require an efficient algorithm. The PID algorithm features control output range limiting with integrator anti-windup.

Currently, the Proportional-Integral-Derivative (PID) algorithm is the most common control algorithm used in industry. Often, people use PID to control processes that include heating and cooling systems, fluid level monitoring, flow control, and pressure control. In PID control, you must specify a process variable and a setpoint. The process variable is the system parameter you want to control, such as temperature, pressure, or flow rate, and the setpoint is the desired value for the parameter you are controlling. A PID controller determines a controller output value, such as the heater power or valve position. The controller applies the controller output value to the system, which in turn drives the process variable toward the setpoint value.

9.7.3.2 CONFIGURATION

In the configuration area of this function you can configure all parameters required for proper PID control.

You need to configure the following parameters:

1 *Actual Value Input* – this parameter specifies the measured value of the process variable being controlled. This value is equal to the feedback value of the feedback control loop. Enter a variable name here, or enter a *Device Property Identifier* in order to directly read specific process data of a certain device.

2 *Setpoint* – specifies the setpoint value, or desired value, of the process variable being controlled. You can enter a fixed value (like 50°C) or you can pass the value in a variable.

3 *Control Loop Parameters* – specify the proportional gain (K), integral time (T_I), and derivative time (T_D) parameters of the controller and thus directly affect the control characteristic of the controller.

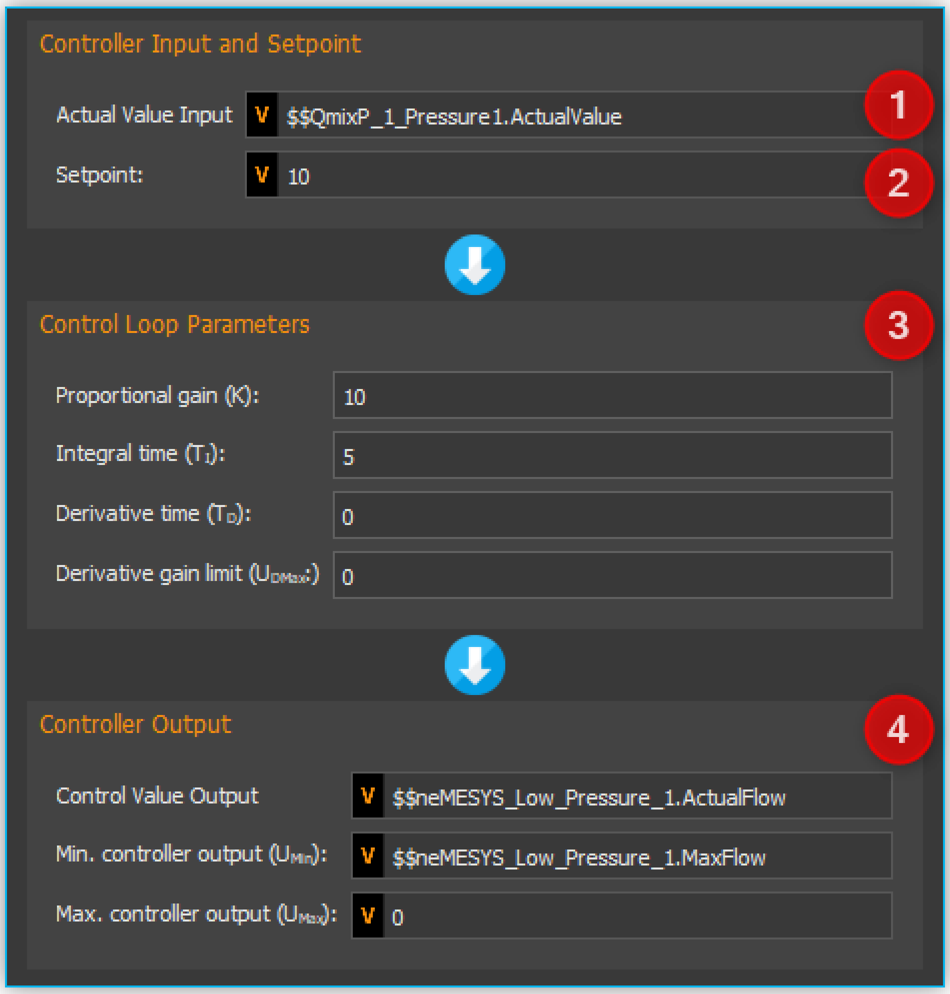


Figure 219: PID Control function configuration

4 *Controller Output* – This section groups all settings for the controller output. *Control Value Output* returns the control output of the PID algorithm that is applied to the controlled process. I.e. if you implemented a temperature control loop, the *Control Value Output* would be the heating power that must be generated from the heater. Enter a variable name of a variable that can store the output value or use a Device Property Identifier to write the value directly into a device property of a certain device.

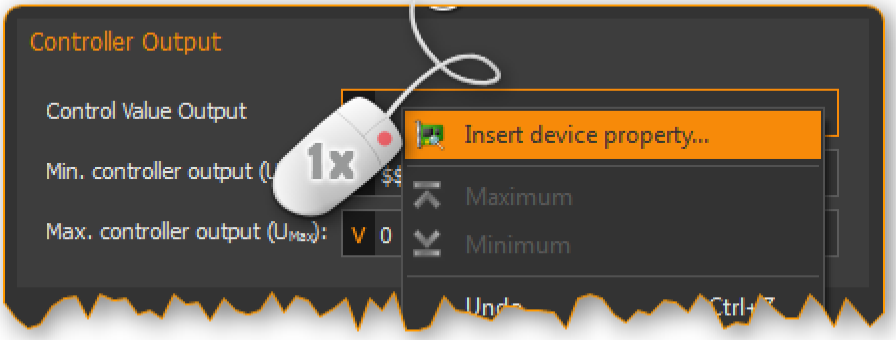


Figure 220: Enter process data identifier via context menu

With the parameters U_{max} and U_{min} you limit the range of the controller output. If you e.g. control the heating power via a 0 – 5 V analog output, then enter 0 for U_{min} and 5 for U_{max}. If the control algorithm generates values that are outside of this range, they will be limited to the range automatically.

HINT. You can use the PID control function to implement a P, PI or PD controller by simply setting the unneeded PID gains to 0.

9.7.3.3 PID CONTROL PARAMETERS

A set of PID control parameters contains a proportional, an integral and a derivative part.

PROPORTIONAL PART

The proportional part computes the control deviation from the setpoint (W) and the actual value (X)

$$e=(X-W)$$

and multiplied with the proportional gain, gives it as control value (Y) to the controlled system. The following equation shows the proportional part.

$$Y=K\cdot e$$

INTEGRAL PART

Mathematics forms the integral part the area enclosed by control deviation and timeline. If there is a contant control error, the integral part grows ramp shape.
For a constant control deviation the equation is:

$$Y = \frac{K}{T_i} \cdot e \cdot t + Y_{t0}$$

Y_{t0}: Control value at the beginning of the observation

T_i: Integration time

If the actual value equals the setpoint the control value does not change. The control value built up by the integral part remains and will not decrease until the actual value exceeds the setpoint value. With controlled systems containing a delay line the integral part eliminates the steady-state error which a proportional controller is not capable of. In general the following equation holds for the integral part.

$$Y = \frac{K}{T_i} \cdot \int_{t_0}^{t} e\,dt + Y_{t0}$$

With the integration time the speed of the controller can be changed. The smaller T_I the faster the integral part builts up a control value. The given equation states that the proportional gain K, too, impacts the integral part. In QmixElements the integral part can only be configured in conjunction with the proportional part. Thus the following equation applies:

$$Y = K \cdot e + \frac{K}{T_i} \cdot \int_{t_0}^{t} e\,dt + Y_{t0}$$

IMPORTANT. The integral part compensates the steady-state control error.

DERIVATIVE PART

The derivative part counteracts changes in the actual value. Two scenarios can be considered with respect to the effect of derivative part.

- After the actual value has reached a stable final value, it is decreasing because of a disturbance suddenly occurred. The derivative part gives an additional control value that helps increasing the actual value.
- If the setpoint value is increased, the actual value increases too. The derivative part recognizes the increasing actual value and slows down by an additional negative control value starting up to the target value.

In practical use the derivative part occurs only in combination with a proportional part. The controller equation is:

$$Y = K \cdot e - K \cdot T_d \frac{dx}{dt}$$

The bigger the proportional gain K and the derivative time T_D the bigger the effect of the derivative part and the stronger the change of the actual value is counteracted.

SUMMARY

The following table summarizes the effect of the different control parameters.

PID PARAMETER	REGULATING ON A DISTURBANCE OF THE CONTROLLED SYSTEM	START-UP OF THE SETPOINT
K higher	stronger repsonse (reduced damping)	faster start-up
K smaller	weaker response (increased damping)	slower start-up
TI higher	weaker repsonse, particularly on short-term disturbances, the controller has only a weak response	Slower start-up and compensation of the steady-state error
TI smaller	stronger repsonse, particularly on short-term disturbances, the controller has only a weak response	faster start-up and compensation of the steady-state error (Overshooting if T_I is too small)
TD higher	stronger repsonse	slower start-up (stronger reaction against changes in actual value)
TD smaller	weaker response	faster start-up (weaker reaction against changes in actual value)

9.7.3.4 PROGRAMMING THE CONTROL LOOP

This section shows you how to realize a complete PID control loop in a script program by using the PID control function.

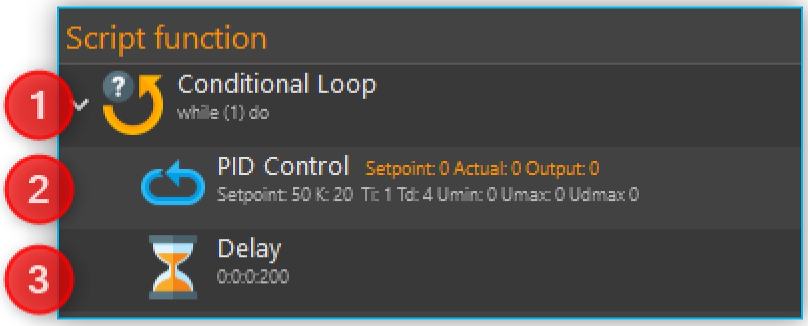


Figure 221: PID controller example scripts

To implement a control loop, proceed as follows:

1 The controller must be called cyclically in a fixed time interval. For this, you should use a loop. In this case, use a conditional loop (*Conditional Loop*) with the loop condition 1. The condition 1 is always true and the loop runs forever and never stops, except the user stops program execution manually.

2 Now create a *PID Control* function within the loop and configure all parameters.

3 Inside the loop you need to place a *Delay Function* to insert a defined delay time 200 milliseconds. This time determines the frequency, with which the control algorithm is called and hence the *dt* that is used in the algorithm for the calculation of the control parameters.

Now you have built a control loop that reads the current temperature via an analog input, computes an output value in the PID control function, which is then used with an analog output for adjusting the heating power.

HINT. According to control theory, a control system must sample a physical process at a rate about 10 times faster than the fastest time constant in the physical process. For example, a time constant of 60 s is typical for a temperature control loop in a small system. In this case, a cycle time of about 6 s is sufficient. Faster cycling offers no improvement in performance

9.8 How to set controller parameters

9.8.1 Closed control loop and PID controller equation

The controller (e.g. PID controller) and the controlled system (e.g. temperature-controlled system) together make up a feedback system, the closed control loop. (see figure below)

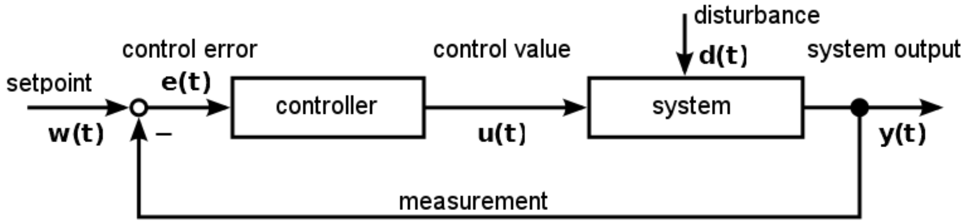


Figure 222: Closed control loop

A PID-controller determines the control value *u* at the time t_1 using the following equation:

$$u(t_1) = K_p \cdot e + \frac{K_p \cdot \Delta t}{T_i} \cdot \sum_{t=t_0}^{t_1 - \Delta t} e + K_p T_d \frac{\Delta y}{\Delta t}$$

The control value contains 3 shares.

$K_p \cdot e$	The proportional share (P) forms by means of the factor K_p the direct effect of the control error on the control value.
$\frac{K_p}{T_i} \cdot \sum_{t=t_0}^{t_1 - \Delta t} e$	The integral share (I) computes the sum of the error over the time and by means of K_p and the time constant T_i maps it to the control value. The bigger K_p and the smaller T_i the bigger the integral share of the control value.
$K_p T_d \frac{\Delta y}{\Delta t}$	The differential share depends on the temporal change of the actual value that is mapped to the control value by K_p and T_d.

9.8.2 Preparations for setting controller parameters in QmixElements

Initially make the plot of the datalogger plugin show the measurement value and the control value to you. (refer to section Process Data Graph) 1 It is recommended to do the same with the setpoint value. It is not necessary but facilitates orientation within the plot. (see figure below)

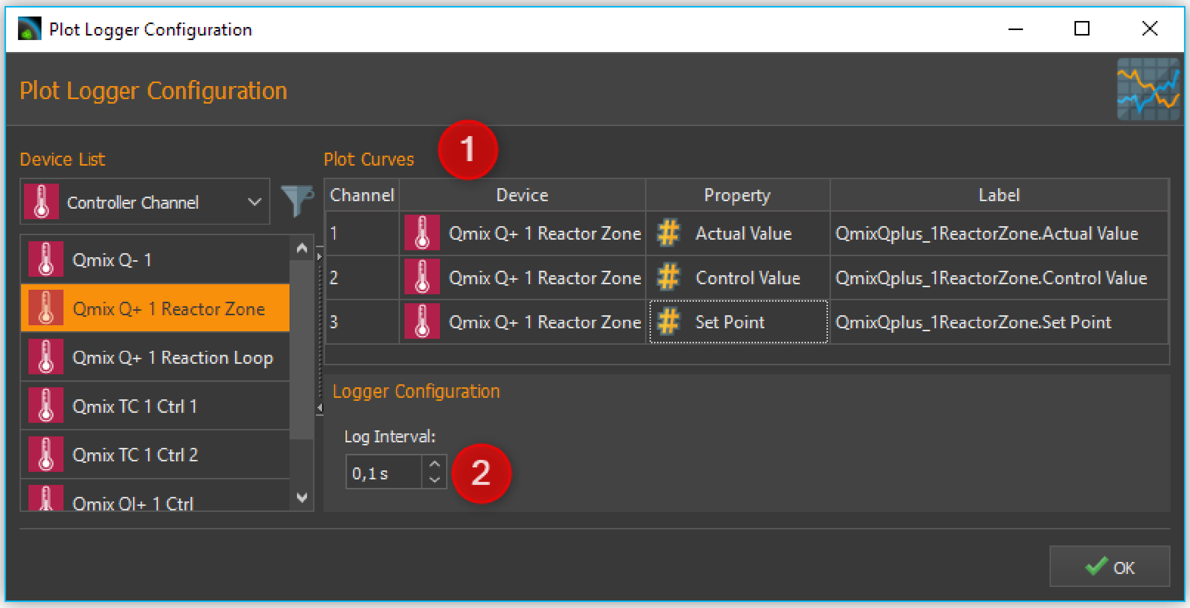


Figure 223: Configuration of the graphical logger for setting controller parameters

The setting for *Log Interval* 2 depends on the change frequency of the actual value. You should get a useful graph if you set *Log Interval* to the sample time used by your control channel. (see section 9.8.3)

9.8.3 Choosing the sample time

The time between 2 computations of the control value is defined as sample time. The smaller the sample time the more often the control value is calculated. As a rule of thumb you can keep in mind that the sample time should not be higher than one tenth of the smallest time constant in the closed control loop. Experiences have shown that the following values achieved useful results (stable control) with the appropriate devices.

APPLICATION	SAMPLE TIME (MS)
Qmix Q+	500
Pressure control using neMESYS and Qmix p	50

HINT. For the sample time choose values that are less than or equal to 1/10 of the smallest time constant occurring in the control loop.

9.8.4 Setting the control value limits

You can limit the minimum (U_{min}) and the maximum (U_{max}) control value of the QmixElements controllers. The control value should have a sufficient stroke in order to reach the desired setpoint values. But you should also take care not to damage the controlled system by choosing to large control value limits. (e.g. too high flow rate of a neMESYS dosing module in a pressure controlled system leads to damage of the fluidic system) You should test the control value limits by temporarily acting with them on your controlled system. (e.g. dose with a neMESYS dosing module using a flowrate equal to the control value limit) Futhermore you have to choose a value assumed by the controller if the control channel is disabled ($U_{disabled}$) which is usually zero.

ATTENTION. Insufficient limitation of the control value can lead to damage of the controlled system.

9.8.5 Determination of PI control parameters using the example of a temperature control system

A temperature control system usually makes up a delay line containing one or more delay time constants. It can often be approximated by a first-order delay line. The appropriate step response function is:

$$F(s)=\frac{K_s}{1+sT_1}$$

Setting the controller parameters targets the compensation of the delay time constant T_1 and adapting the controller gain K_p in order to reach an efficient control behaviour. In practice T_1 is often unknown, but you can gradually approach to a useful result using the following procedure.

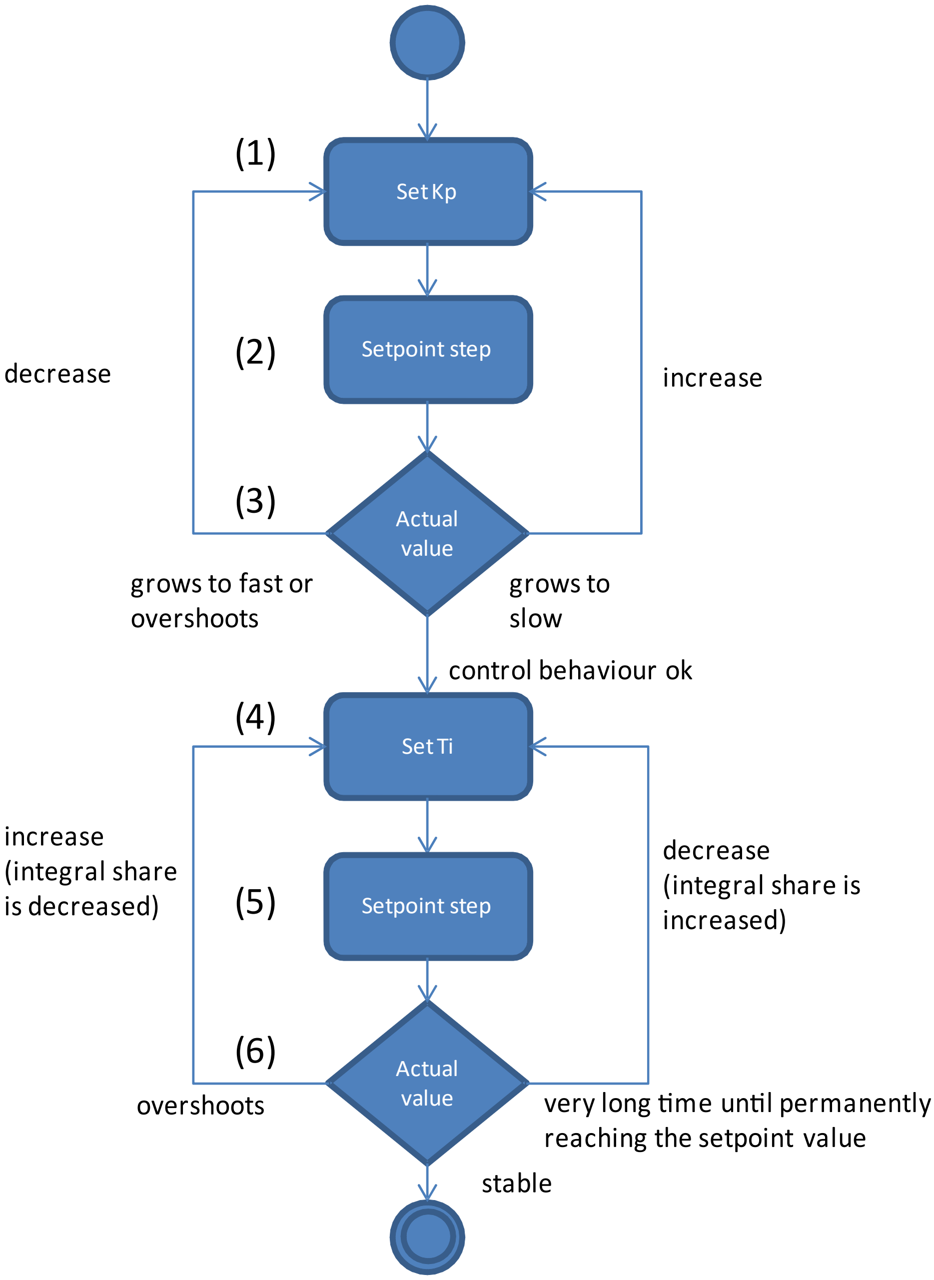


Figure 224: PI-controller setting for a first-order delay line

(4) Please choose the values for Sample Time, U_{min}, U_{max} und $U_{disabled}$ according to the recommendations of the sections 9.8.3 and 9.8.4. Disable the differential and integral share by setting the controller time constants to 0. This leads to a simplified controller equation. $U = K_p \cdot e$

Set a moderate value for the proportional gain. Keep in mind that by means of K_p the control error immediately impacts the control value. Choosing a too big value for K_p can drive the controller into saturation.

(5) Give a setpoint step to your control loop, i.e. by changing the setpoint temperature from room temperature to 50°C and activating the control channel. (see section 9.2.4 and 9.2.3)

(6) Providing a first-order delay line the actual value will behave according to the following figure.

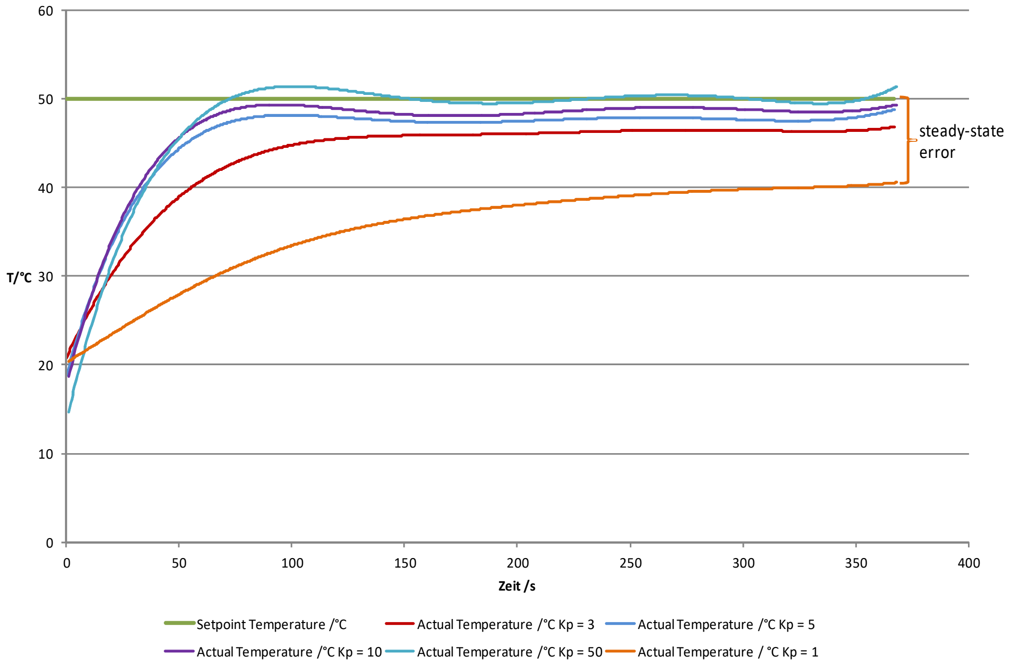


Figure 225: Temperature control using P-controller with different K_p values

Depending on chosen controller proportional gain K_p the actual value will vary quickly settle near the setpoint value. Due to the fact that a proportional controller is not able to fully compensate a delay line, a steady-state error establishes. If K_p is set too low, the actual value approaches only very slowly the target value. (see curve for K_p = 1 in figure above) If K_p is set to high, the actual value overshoots, optionally oscillates about the setpoint. (see curve for Kp = 50 in figure above) In the provided example the actual value reaches a steady-state quickly without overshooting using K_p=3. This is why we use that value for further optimizations.

(7) In the next step set T_i in a way that the steady-state error is compensated. You should begin using a large time constant T_i which means a small integral share.

(8) Give a setpoint step to your control loop, i.e. by changing the setpoint temperature from room temperature to 50°C and activating the control channel. (see section 9.2.4 and 9.2.3)

(9) Lower the time constant T_i if you want to reduce time for permanently reaching the setpoint value. Please notify that a time constant T_i set to small (large integral share) can lead to the control loop oscillating. In the provided figure you can see that T_i=260s leads to a good result. The actual value matches the setpoint value and the system does not oscillate. Using T_i=1000s the setpoint is not reached within the illustrated time range and T_i=20s leads to the system overshooting heavily. (see figure below)

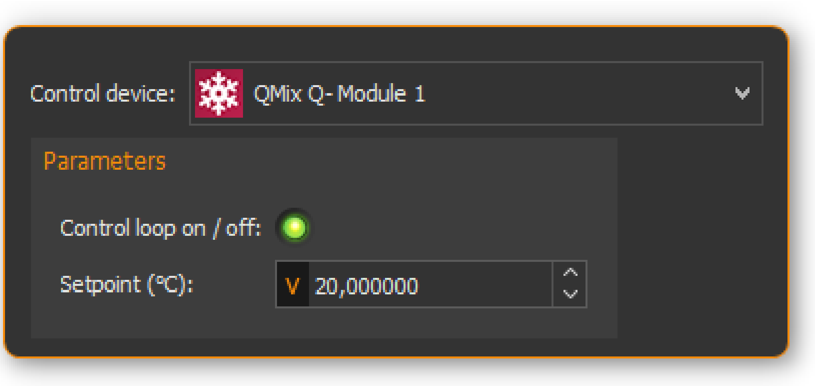


Figure 226: Configuration Set Controller Param function

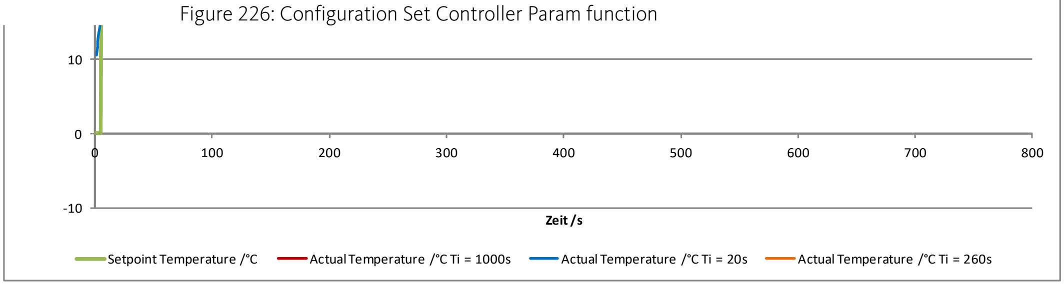


Figure 227: PI temperature-controller using Kp=3 and variable Ti values

(10) In many cases (i.e. temperature control) a PI controller is sufficient. There is no steady-state error and the dynamic behaviour is satisfactory. If the controller shall be robust with respect to sudden disturbances, it might be useful to include a differential component. A detailed consideration of control stability, control behaviour with respect to setpoint changes and disturbances is beyond the scope of this practical introduction. Reference is therefore made at this point on the control engineering literature.

(11) Now create a PID parameter preset with the values you have determined and assign a unique name.

10 QmixV Plugin

10.1 Introduction

The QmixV plugin is used to control the QmixV valve modules or to control valve devices that are part of other devices (i.e. valves mounted on neMESYS syringe pumps).

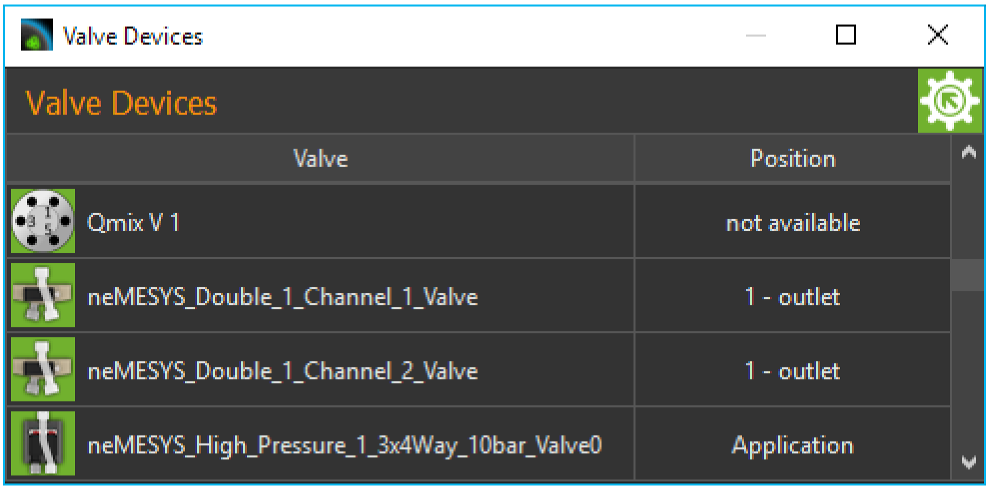


Figure 228: QmixV valve device list

10.2 Valve Device List

The plugin mainly consists of the valve device list (see Figure above), which is displayed as a tool window in the graphical user interface. You can move the window to another position in the graphic interface at any time by dragging and dropping the title bar, or you can move it out of the interface to become a separate window.

The valve device list shows you all the available Qmix valve devices in tabular form. The name of the module and the actual valve position status icon is shown in the left-hand table column, and the right-hand column shows the name of the current position of the valve.

10.2.1 Editing valve names

You can change the name of a valve at any time and, for example, assign a name suitable for your particular application. You change a name by the following steps:

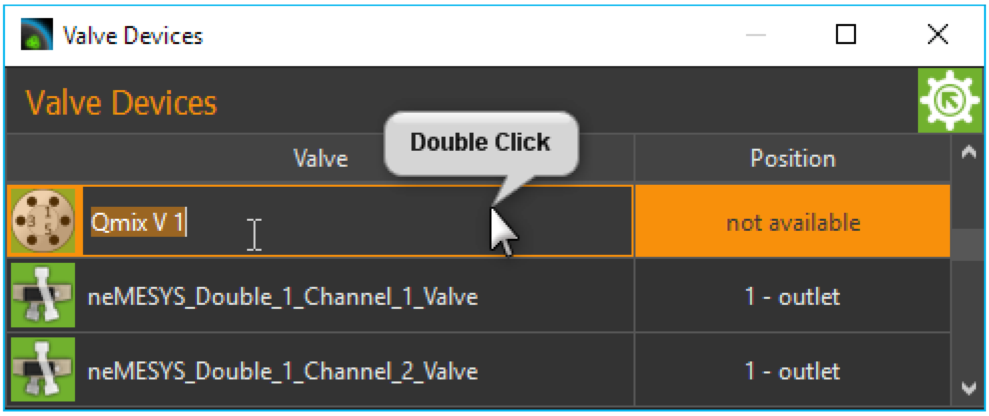


Figure 229: change valve name

1. Double-click the table cell containing the name you want to change.
2. Enter the new name in the Editing window which now appears (see Figure above)
3. Complete your changes by pressing the *Return* key.

10.2.2 Switching valve position

You can also switch the valve position in the module list. For this purpose proceed as follows:

1. Double-click in the table cell showing the valve position.
2. Select the desired valve position in the selection box that now appears (see Figure below).
3. Confirm the selection by pressing *Return* key or click on another table cell to close the selection box.

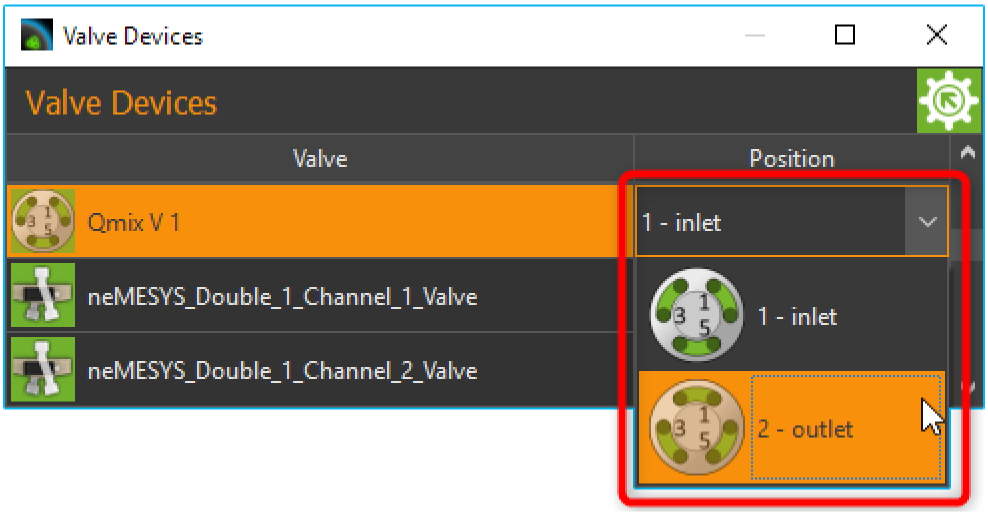


Figure 230: Switch valve position

The valve is now switched into the new valve position.

10.2.3 Naming valve positions

You can give each valve position an individual name in the software, and so configure the names of the valve positions to suit your application. Perform the following steps to change the names of the valve positions.

(1) Make a right mouse click in the line of the table of the valve for which you want to rename the positions.

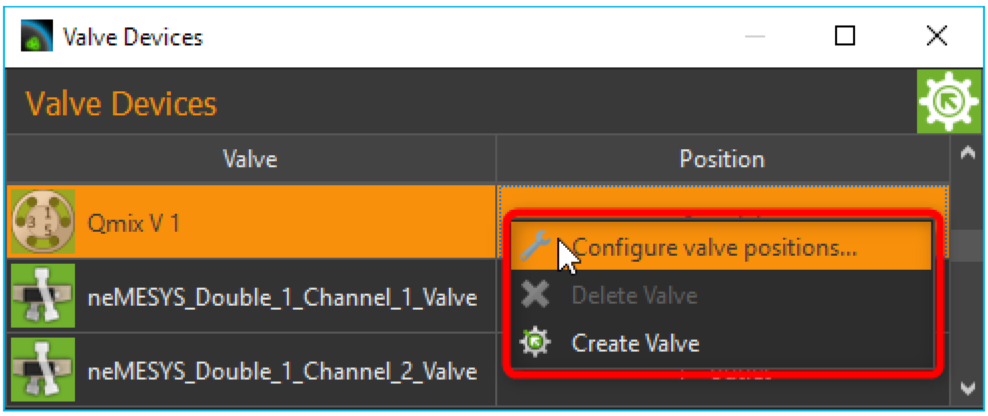


Figure 231: Open valve configuration

(2) Select the *Configure valve positions...* menu item in the context menu, that is being displayed.

(3) A configuration dialog opens (Figure below), which consists of a valve selection box 1 and the list of valve positions 2.

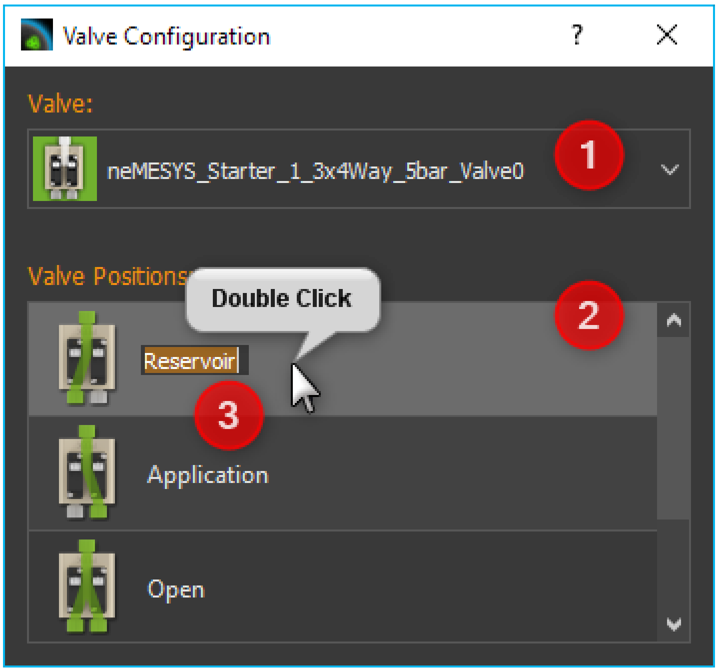


Figure 232: Open valve configuration

(4) Double click the line in the table containing the valve position you want to name.

(5) An Editing window 3 opens in the table cell, in which you can enter the new name of the valve position.

(6) Confirm your entry by pressing the *Return* key.

(7) You can close the configuration dialog when you have named all the valve positions.

From now on, the valve positions will always be shown with the new names in all areas of the software.

10.3 Creating Valves

You can control CETONI valves via one or more digital outputs. If you have connected your valve to the corresponding digital outputs, you must configure it in the software.

The first step is to open the I/O channels window and manually switch the digital outputs to check that the valve is correctly connected and switching.

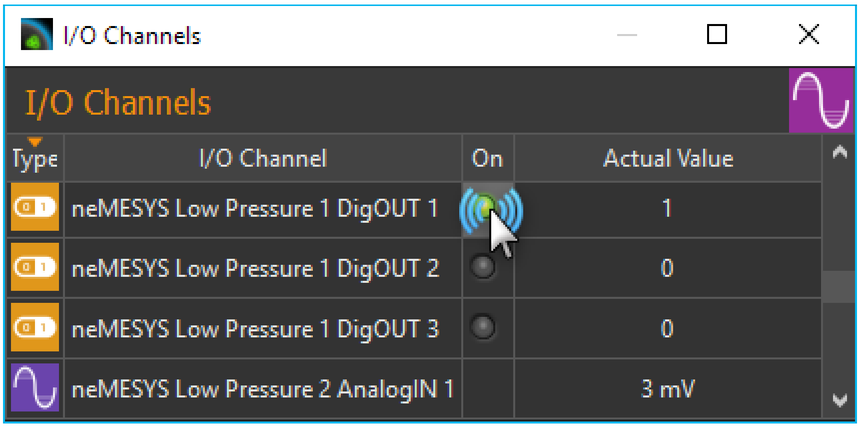


Figure 233: List of I/O channels - Test by manual switching of the digital outputs

Click on the LED in the column "*On*" at the corresponding channel to switch the digital channels off and on again. Check whether the valve is switched when switching the digital output. If the valve is not switched, it is either connected incorrectly or you are switching the wrong output. If the valve switches, you can continue creating the valve in the software.

To do this, right-click in the list of valves and then select *Create Valve* from the context menu.

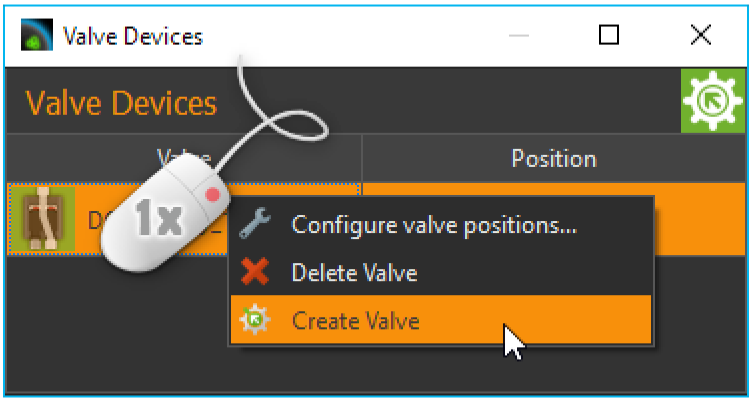


Figure 234: Open dialog for valve creation

The dialog for creating and configuring a valve is now displayed.

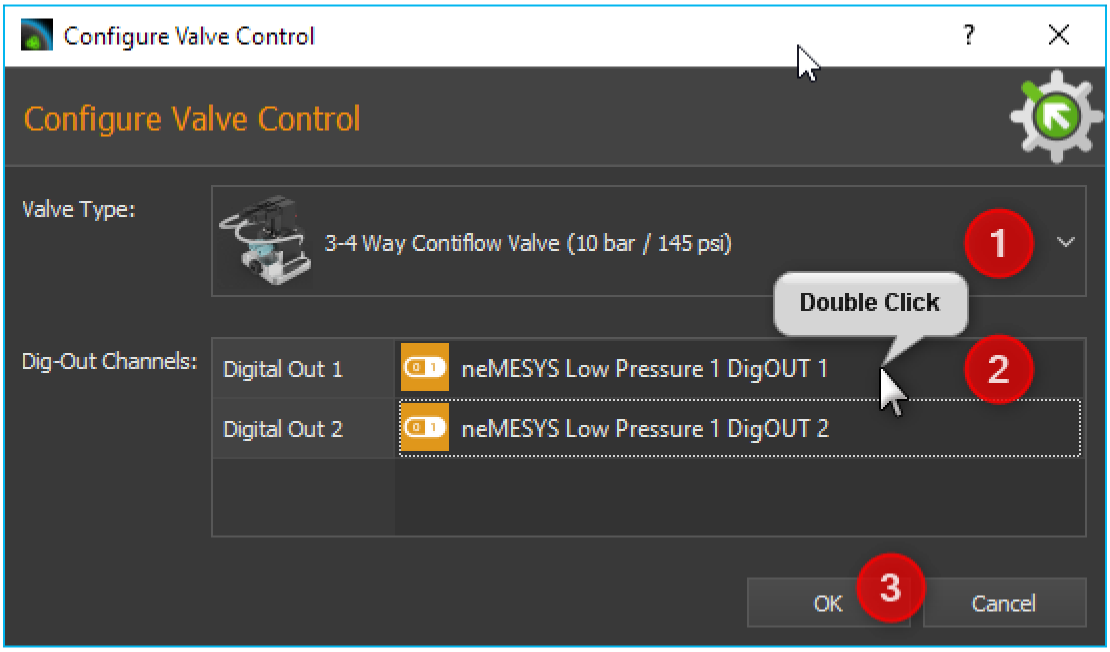


Figure 235: Dialog for valve creation and configuration

In the window, first select the type of valve you have connected in the field *Valve Type* ❶. Then you must configure the digital outputs for the valve in the *Dig-Out Channels* table. Depending on the valve type, these can be one or more outputs. To configure an output, double-click in the table cell ❷ for this channel and then select the corresponding digital output from the list. The *OK* button ❸ is only activated when the corresponding digital output channel has been configured for each switching input of the valve. You can then complete the configuration by clicking *OK*.

The newly created valve now appears in the list of valves. Valves created with the *Create Valve* function get the prefix DO_ to indicate that they are valves controlled by digital outputs.

HINT. Valves created with the *Create Valve* function can be identified by the prefix DO_ in the valve list.

10.4 Deleting Valves

Valves that you have created dynamically in the software, e.g. with the *Create Valve* function (see previous section), can be deleted from the valve list via the context menu. To do this, right-click on the line of the valve you want to delete and then select the *Delete Valve* menu item.

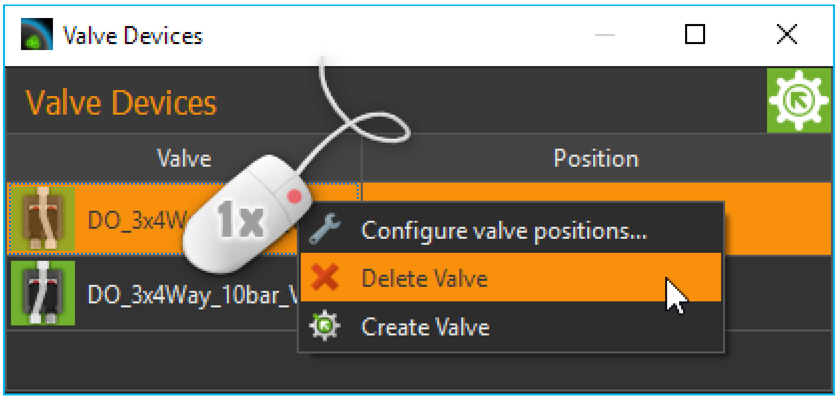


Figure 236: Deleting valves

IMPORTANT. Valves created and configured with the *Device Configurator*, or that are part of other devices, cannot be deleted with *Delete Valve* menu item.

10.5 Valve Script Functions

10.5.1 Introduction

The Qmix valve plugin contains a script function for switching the Qmix valves from a script.

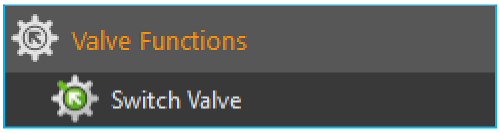


Figure 237: Qmix valve script functions

10.5.2 Switch Valve

You use this function to switch the valve position. In the configuration area of this function, you can select the valve module, and the target position at which the valve is to be switched.

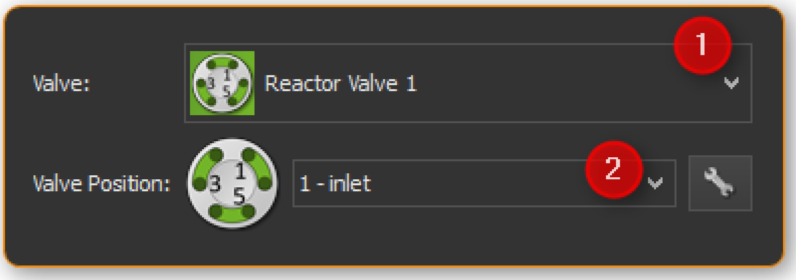


Figure 238: Switch Valve function configuration

11 rotAXYS / neMAXYS Plugins

11.1 Introduction

The QmixElements software supports the control and visualization of various cetoni positioning systems. The description in the following sections is valid for all rotAXYS and neMAXYS devices.

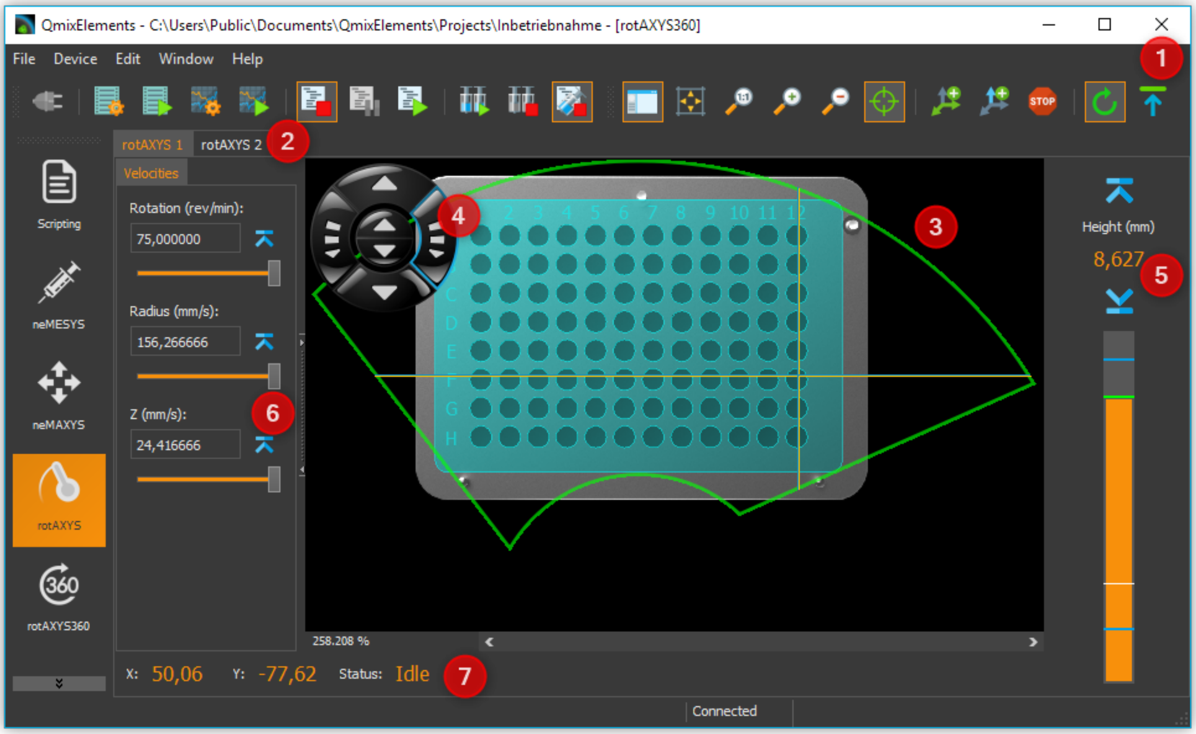


Figure 239: Work area for positioning systems

The working area of the this plugin mainly consists of the following components:

4 Toolbar

5 Device selection tabs

6 Positioning map for XY positioning

7 Control pad for manual control

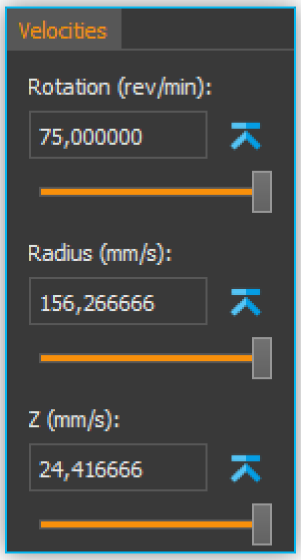
8 Z panel for positioning of Z axis (vertical axis)

9 Sidebar

10 Status Bar

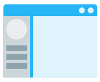
If you work with multiple devices simultaneously, you can use the device selection tabs 2 to show the control panel for the positioning device you want to work with.

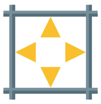
11.2 Sidebar



In the sidebar, which can be turned on and turned off, you can find additional controls to configure and control your positioning systems. In the sidebar there is the *Velocities* panel where you can configure the velocities of all axes. All positionings in the *Positioning Map* or in the *Z-Panel* are then carried out with the set speeds.

11.3 Toolbar


Hides or shows the sidebar with additional configuration and control elements


Adjust the magnification of the positioning map so that the complete display fits into the working area.

Changes the magnification of the positioning map to the actual pixel size


Increases the magnification factor


Decreases the magnification factor


Switches between free positioning and well positioning (see section 11.5 - Positioning Map)


Adds the script function *Move XY* with the current XY position as target position parameter to the script editor


Adds the script function *Move Z* with the current position of the Z axis as target position parameter to the script editor

Immediately stops the movement of all axes of the positioning system


Activates / deactivates the lift axis safety stroke. If it is active, then the Z-axis moves to a safe height before each movement in the XY plane.


Sets the current Z-axis position as safe height. If the safety lift feature is active, then the positioning system moves to this height before each movement in the XY plane.


Configure zero angle - the zero angle is never crossed by the axis system and it is the start and end point of the 360 ° rotation range

11.4 Control Pad

11.4.1 rotAXYS

With the 6-way control pad, you can manually position all the axes of the rotAXYS positioning system.

Rotate the rotary axis counter-clockwise

Rotate the rotary axis clockwise

Move the radius axis away from the centre of rotation (increase radius)

Move the radius axis toward the centre of rotation (decrease radius)

Move the vertical axis up and down

11.4.2 neMAXYS

With the 6-way control pad, you can manually position all the axes of the neMAXYS positioning system.

	Moves X-axis to the left
	Moves X-axis to the right
	Moves the Y-axis toward the back of the device
	Moves the Y-axis toward the front of the device
	Move the vertical Z-axis up and down

11.4.3 Changing the speed

When a direction is selected on the control cross, the corresponding axle is positioned at a fixed velocity. This velocity can be changed by simultaneously pressing a key on the keyboard. First press the key on the keyboard, and then click a direction button in the control cross.

The following keys can be used:

Shift key

Increases the velocity by a fixed factor

Control key

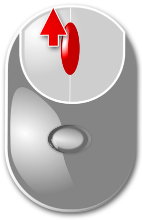
Decreases the velocity be a fixed factor for precise, fine positioning

ATTENTION. Danger of damage through collision with the sample holder! For example, if there is a needle in a holder, ensure that moving the rotary or linear axis will not cause collisions when the tool tip is lowered.

11.5 Positioning Map

11.5.1 Introduction

With the positioning map you can position the positioning system in the XY plane by clicking points on the positioning map. The magnification of the positioning map can be changed via the context menu in fixed steps or continuously adjusted with the mouse wheel.

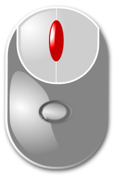


Increase magnification (zoom in)

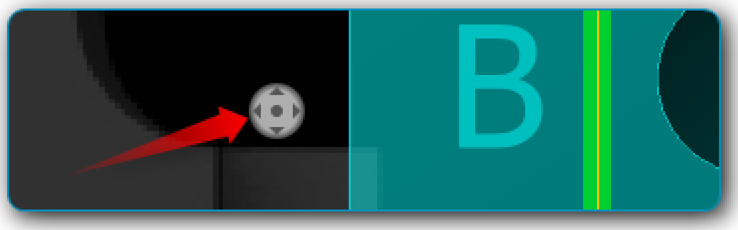
Decrease magnification (zoom out)

For navigation within the positioning map you can also use the mouse. You can move the currently visible area with the two scroll bars on the right and bottom of the map Alternatively, you can also enable free mouse navigation mode. To do this, follow these steps:

(8) Click the middle mouse button within the positioning map



(9) It is now displayed a navigation cross and the free navigation mode is activated



(10) You can now move the mouse to scroll the visible area of the positioning map. The further you move the mouse away from the on-screen navigation cross, the faster the positioning map is scrolled.

11.5.2 Positioning modes

The positioning map can be used in two different modes:

- Free positioning
- Well positioning

Switch between the two modes with the *Positioning Map* button in the toolbar or with the context menu of the positioning map.

The current position of the rotAXYS device is indicated by a green cross on the positioning map (Figure below).

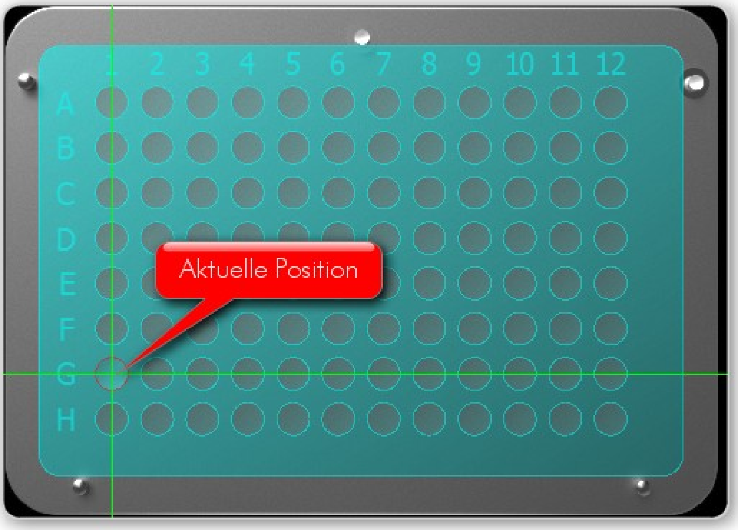


Figure 240: Actual position indicator

IMPORTANT. In order to avoid collisions, the vertical axis always moves automatically into its topmost position before travelling to a position.

11.5.3 Free positioning

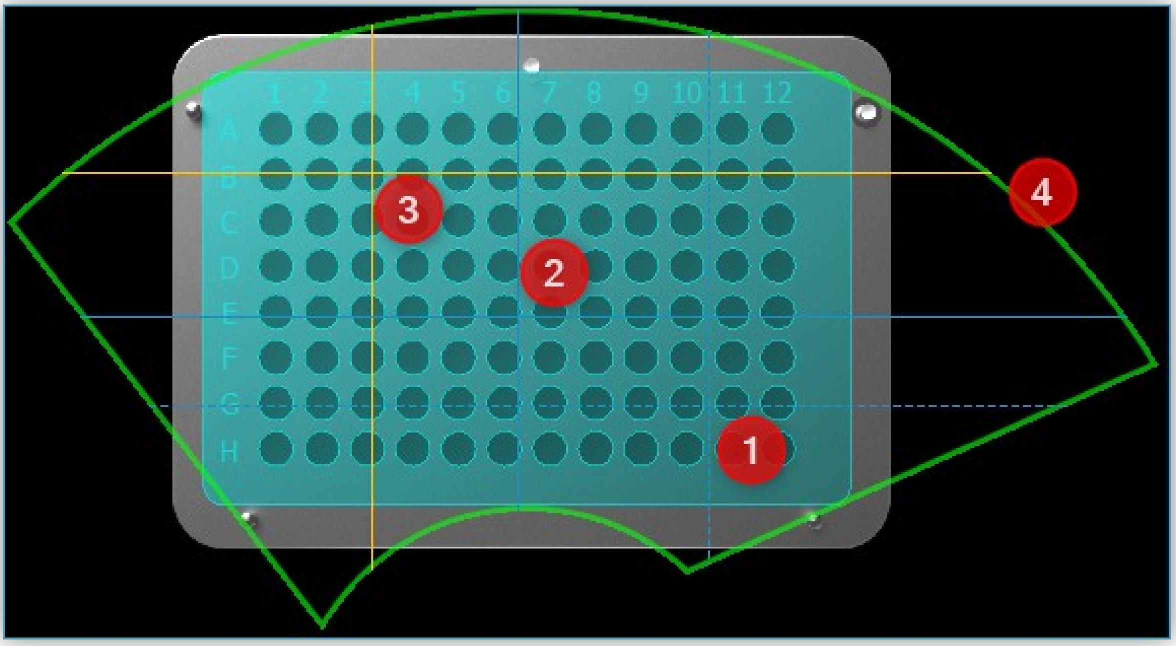


Figure 241: Free positioning mode

With free positioning, you can travel to all positions within a permissible area 4. The permissible area is indicated by a green border (see Figure above). When you move the mouse, the potential target position 1 is indicated by a target cross consisting of two dashed lines. When you click the left mouse button on the target position, the positioning system travels to the selected position. The indicator of the current position 2 then moves to the target position 3.

11.5.4 Well positioning

In this mode, simply click in one of the existing wells to position. As soon as you have clicked the target well, the system automatically moves to the centre of the selected target well.

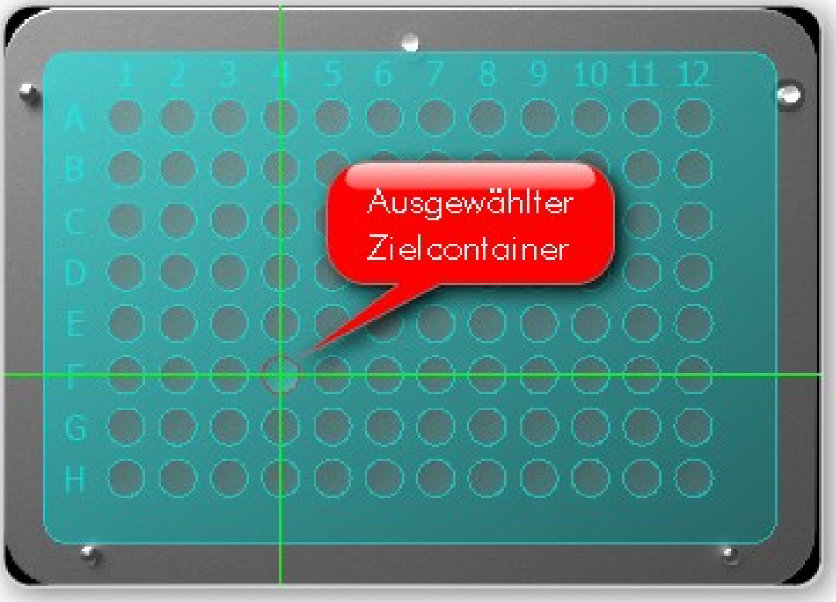


Figure 242: Well positioning

The selected target container is marked with a coloured marker.

11.5.5 Open context menu

Right-click on the positioning map to open the context menu.

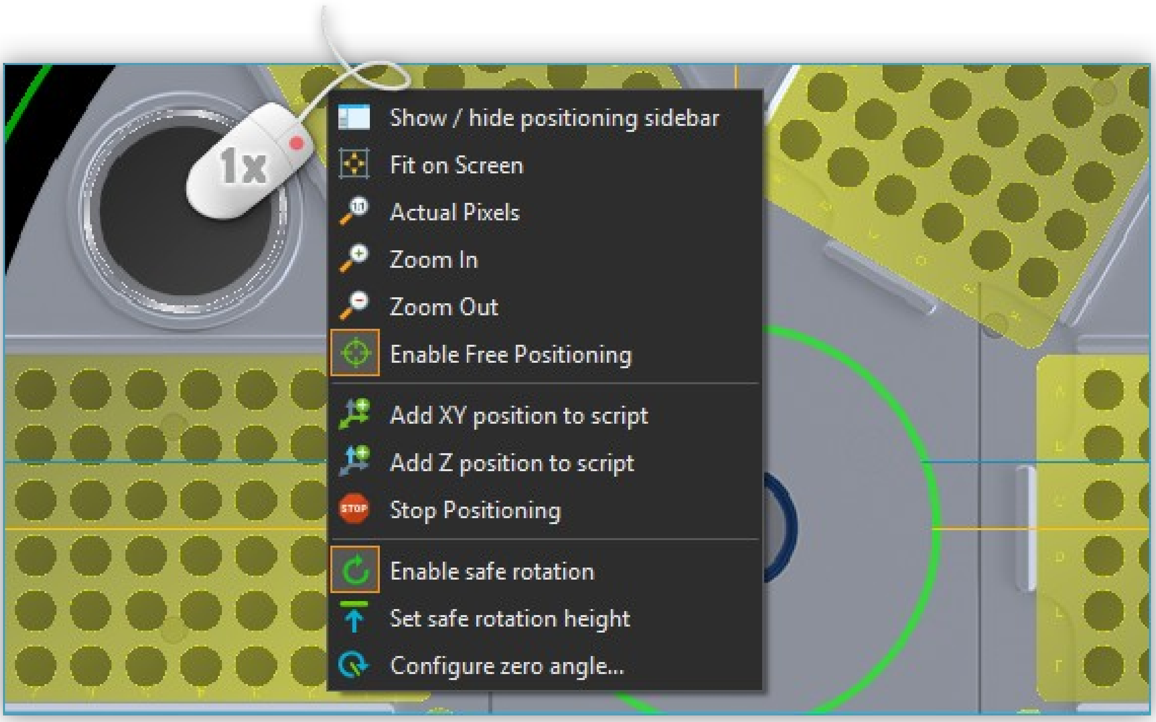
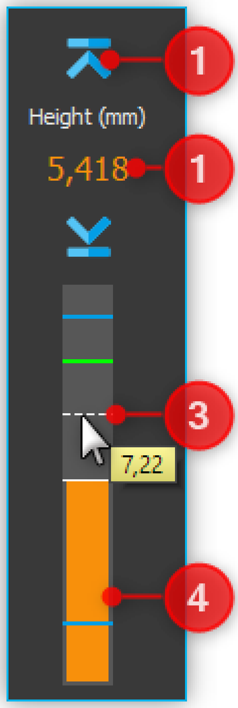


Figure 243: Open context menu

11.6Z panel for Positioning the Z axis

11.6.1 Overview

The Z panel on the right of the positioning map shows you the current position of the vertical axis, and gives you a number of ways of positioning the vertical axis. These are:



1 Buttons for predefined positions

2 Height indicator (position lift axis)

3 Slider to select the target position

4 Graphical height indicator

11.6.2 Buttons for predefined Positions

The two raise and lower buttons 1 simplify quick positioning of the vertical axis to previously defined positions. When you click one of the two buttons, the vertical axis travels at maximum velocity to the position previously defined for this button. With the two buttons you also define the travel range which can be selected with the position slider 3 .

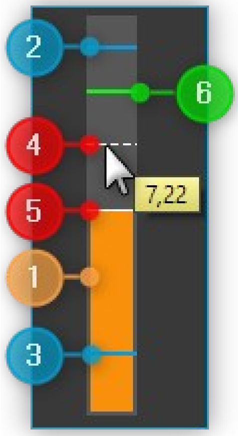
To assign the current position, right mouse click one of the two buttons, and then select *Assign actual position* from the context menu. (see Figure below).

Figure 244: Assign actual position

The assigned position is saved in the configuration data, and is still present when the software is restarted. The configured travel range with maximum and minimum position is indicated in the graphic height indicator by two blue markings (see figure height display).

11.6.3 Height Indicator

The numerical height indicator shows you the current height of the axis in millimetres. A bar in the graphic display shows the current position of the axis within the range of travel.



- 1 **Current position / height** - the colored bar shows the current position of the axis
- 2 **Software Limit Maximum** - shows the maximum position you can select with the slider. The section Buttons for predefined Positions describes how to set the limit.
- 3 **Software Limit Minimum** - shows the minimum position you can select with the slider
- 4 **Target selection mark** - shows the potential target position, which becomes the target position by mouse click. The value for this position is displayed in the message window next to the marker
- 5 **Target position** - if you click with the mouse in the height display, a new target position is defined and the axis moves to this position. The new target position can only be within the software limits. If you click outside the limits, the axis moves to the corresponding limit.
- 6 **Safe position for XY movement** - indicates the safe position for XY movements for certain axis systems (e.g. rotAXYS). This means that if the axis is lower than this position, the axis is moved to this height before any XY movement.

11.7 Status Bar

You can read the current XY position and the current status of the device in the status bar of the rotAXYS working area at all times.

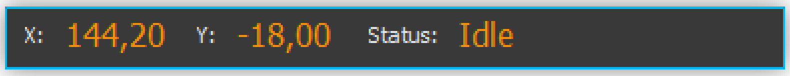


Figure 245: rotAXYS Status Bar

11.8 Z-Axis safety stroke

You can use this button in the toolbar to switch on and off the safety stroke of the Z-Axis. If it is active, then the Z-axis moves to a safe height before each movement in the XY plane. This prevents collisions with sample vessels or other objects on the worktable.

By pressing this button, you define the current position of the Z-axis as the clearance height. If the safety stroke is active, the positioning system will automatically go to this height before any XY positioning.

11.9 Configuring Well Plates (rotAXYS)

11.9.1 Introduction

The *Positioning Map* always shows the currently configured well plate. You can configure new well plates in the software and assign it to a rotAXYS device. To do this, select the main menu item *Edit* → *Configure Well Plate* (see figure below).

IMPORTANT. Well Plate configuration of single well plates is not supported for rotAXYS 360 devices yet.

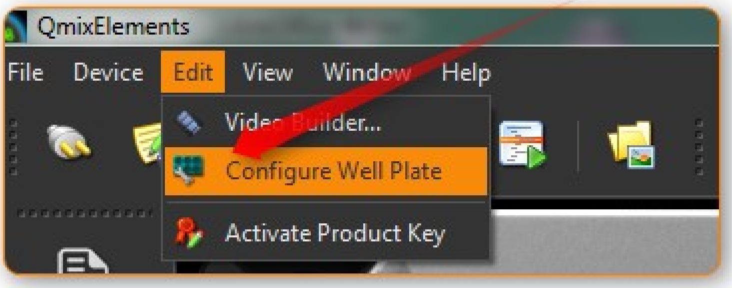


Figure 246: Show well plate configuration dialogue

The configuration dialogue for well plates appears. This configuration dialogue is structured as follows (see figure below):

- 1 list of available well plates
- 2 configuration parameters of current plate
- 3 graphical representation of the current well plate

11.9.2 Selecting an existing well plate

If the list of available well plates 1 already contains a suitable plate, simply select it by clicking on it and confirm the selection by clicking the *OK* button in the configuration dialogue.

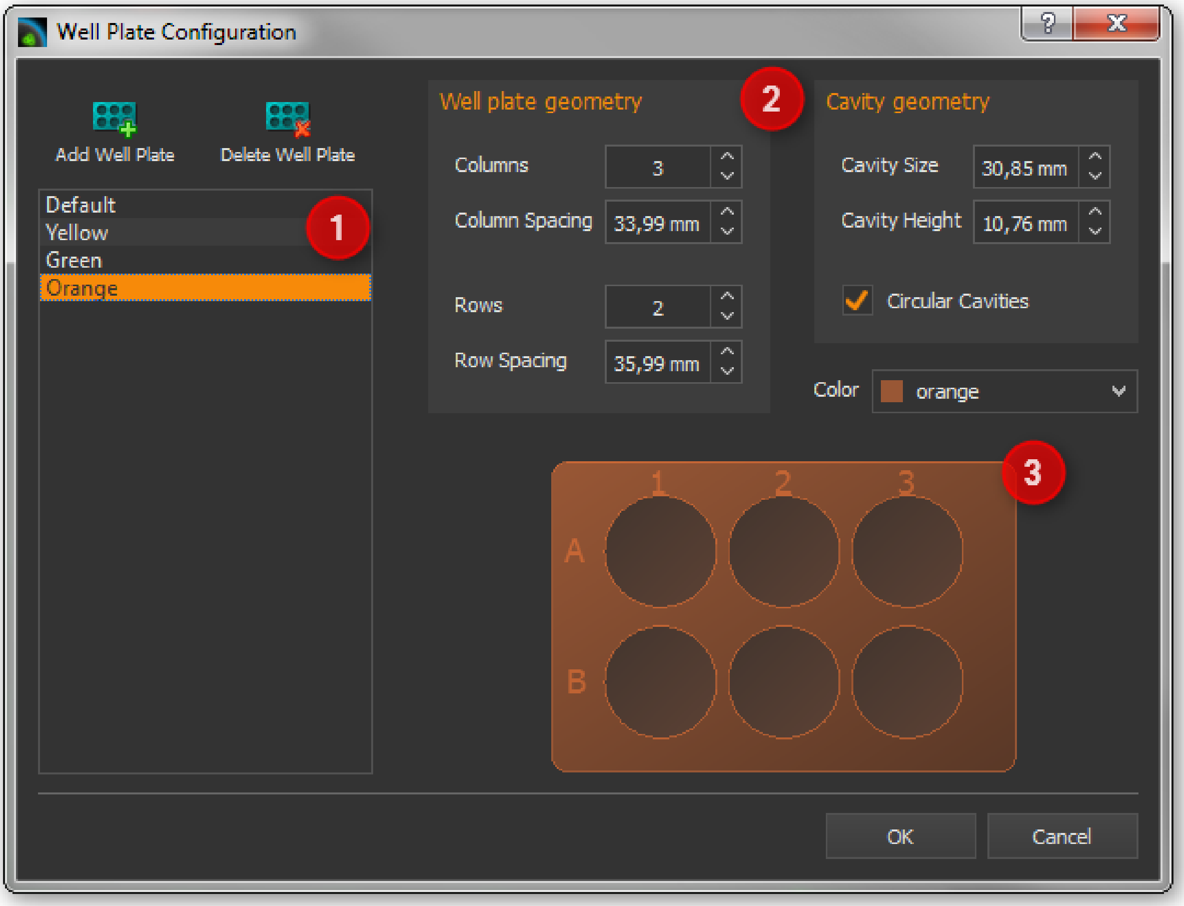


Figure 247: Configuration dialogue for well plates

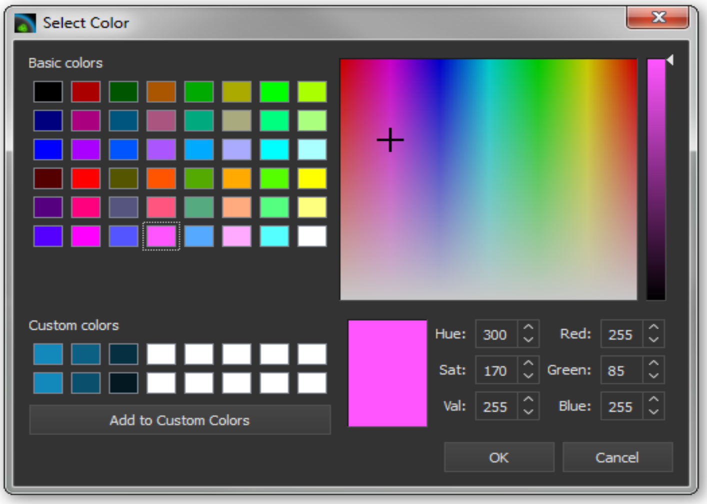
11.9.3 Creating a new well plate

To create a new well plate, perform the following steps:

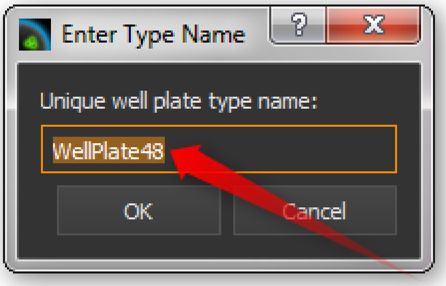
(1) First you need to configure the well plate geometry in the *Well plate geometry* panel 2. You can configure the number of *Rows* and *Columns* and the space between rows (*Row Spacing*) and columns (*Column Spacing*). You will find the required data in the data sheet of the well plate.

(2) Next, configure the geometry of the cavities in the *Cavity geometry* panel. Define the size of the cavities (*Cavity Size*) and their depth (*Cavity Height*). Use the selection field *Circular Cavities* to determine whether the new plate should have round or square cavities.

(3) Finally define the colour of the new well plate. Choose the desired colour in the *Color* selection box. If the right colour is not available, select the entry *More...* The application shows a colour selection dialogue (see figure below) that allows you to select any colour.



(4) Now click the *Add Well Plate* button to add the new plate to the list of available well plates. In the input dialogue that appears (see figure below), enter a unique name for the new well plate and press *Enter* or click *OK*.



(5) The new well plate will be added to the list and can be selected from now on.

11.10 Configuring the rotAXYS360 Worktable

11.10.1 Load a work table configuration

For the rotAXYS 360 positioning system, the individual microtiter plates can not yet be configured by the user, as with the rotAXYS positioning system. However, the rotAXYS 360 positioning system supports the loading of preconfigured work table configuration files, which describe the entire configuration of a particular workbench.

To load a work table configuration, activate the graphical user interface of the corresponding rotAXYS 360 device. Then select *Device* → *rotAXYS 360* → *Load Table Configuration* from the main menu.

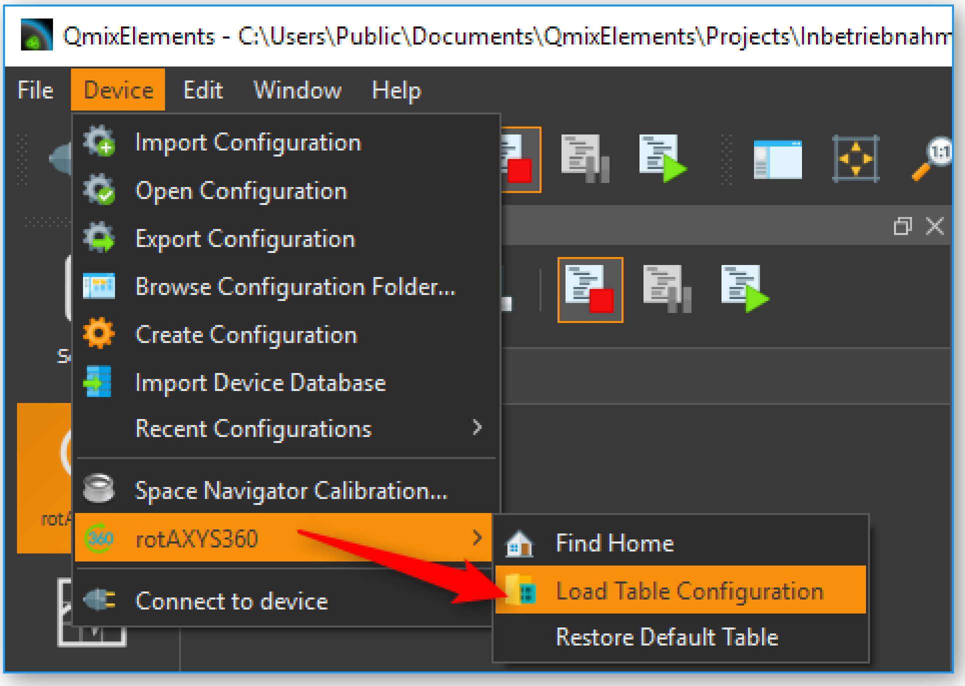


Figure 248: Load a work table configuration

Then, in the file selection dialog that appears, select the configuration file for the work table and load it. The graphical representation of the work table is then immediately updated to the new configuration (see figure example configuration).

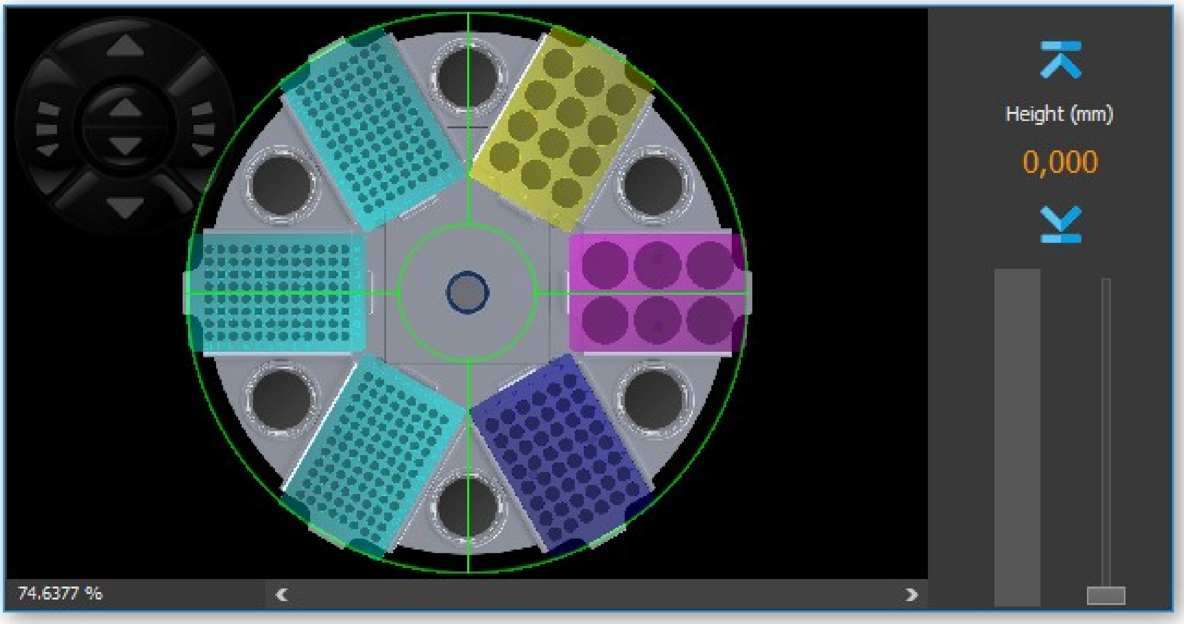


Figure 249: Work table sample configuration

11.10.2 Restore standard work table

To restore the default work table, select *Device* → *rotAXYS 360* → *Restore Default Table* from the main menu.

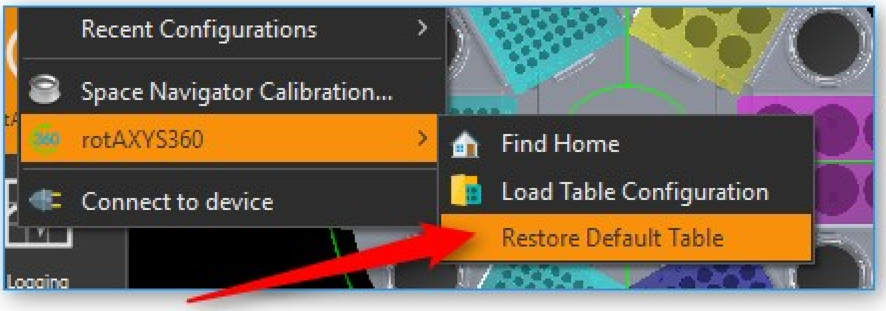


Figure 250: Restore standard work table

11.11 Zero angle configuration

The rotAXYS 360 axis system has an operation angle which is theoretically greater than 360 degrees. In order to prevent entanglement of the hose for fluid dosing, the turning range is limited to 360 °.

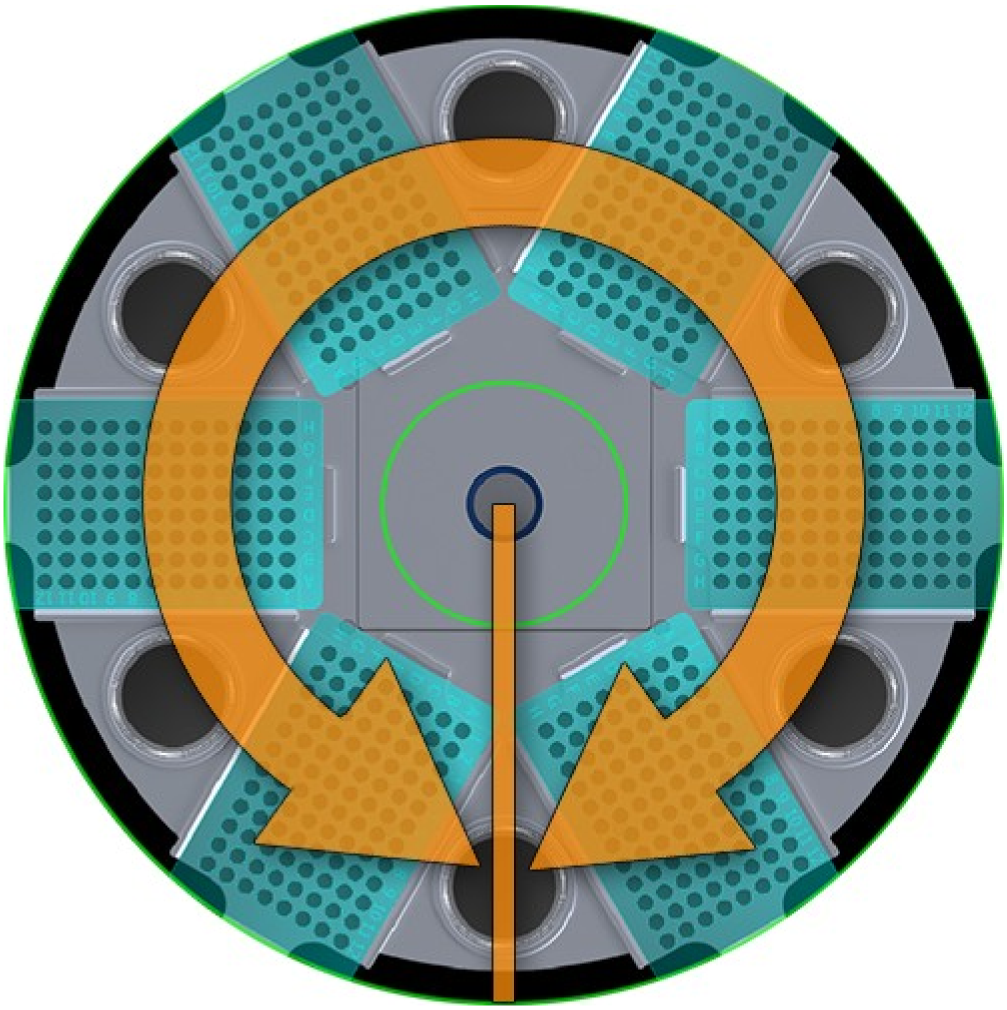
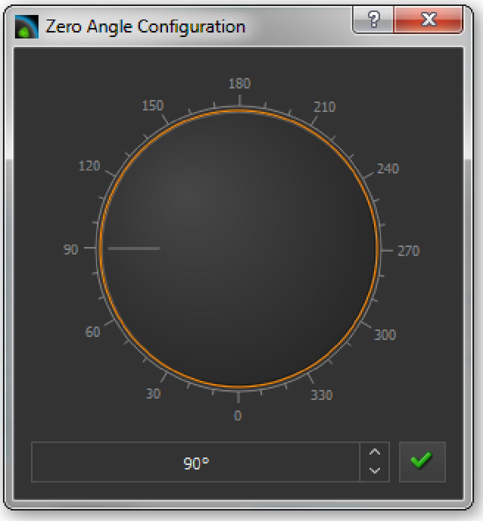
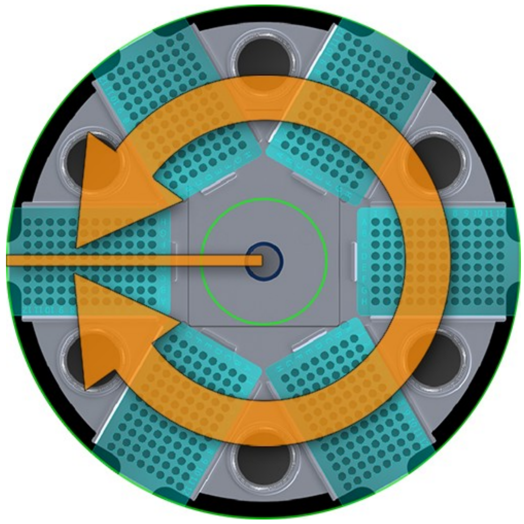


Figure 251: Zero angle configuration

The zero angle ist the angle that will never be crossed by the positioning system. It is the start- and end-point of the 360° rotation range. If this angle is situated between two sample containers that you would like to process, the positioning system will choose the longer route in the opposite direction, so as not to cross the zero angle.

You can move the zero angle to a different angular position in order to circumvent this problem. Press the button for zero angle configuration (image left) to set a new zero angle.





In the dialog which is now displayed (see illustration above), you can freely set the zero angle in the range from 0 - 360 °.

11.12 Positioning with the Space Navigator

11.12.1 Introduction

The Space Navigator is an intuitive input device which allows you to control all axes (rotation, radius and vertical axis) simultaneously.

To use the Space Navigator, connect it to an available USB port on your PC. Latest drivers can be found on our internet page:

http://www.cetoni.de/englisch/service/downloads_software.html

or on disk (CD or USB stick) you received when you purchased the software. Install the driver to use the Space Navigator.

Position the Space Navigator device, so that the cable leads away from you and that the *3D Connexion* label is readable from your position.

11.12.2 Positioning rotAXYS with Space Navigator

11.12.2.1 OVERVIEW

Rotates the rotary axis clockwise or counter-clockwise

Moves the radius axis away from or towards to the centre of rotation

Move the vertical axis up or down

IMPORTANT. Don't use the Space Navigator if the software moves the positioning system (e.g., during active script execution), in order to not interrupt movements.

11.12.2.2 POSITIONING ROTARY AXIS

In order to position the rotary axis, rotate the Space Navigator cap clockwise or counter-clockwise.

Figure 252: Positioning rotary axis with Space Navigator

The speed of the axes can be determined by the strength of the deflection of the Space Navigator cap. The harder you push or twist the cap, the faster the axes are moving. By a slight deflection of the cap, you can very finely and precisely control axis speeds.

HINT. The more you deflect the Space Navigator cap, the higher are the axes speeds.

11.12.3 Positioning radius axis

To move the radius-axis, slide the cap of the Space Navigator to the front or rear. To move the radius-axis to the outer radius, press away the control element from you. To move the radius axis to the centre of the rotation axis, pull the cap to your direction.

Figure 253: Positioning radius axis with Space Navigator

11.12.4 Vertical axis positioning

To position the Z-axis, press down the cap or pull it to the top. The Z-axis is moved in the direction in which you move the control element of the Space Navigator.

Figure 254: Positioning vertical axis with Space Navigator

11.12.5 Positioning neMAXYS with Space Navigator

11.12.5.1 OVERVIEW

 Move X-axis to the left or right

 Move Y-axis forwards or backwards

 Positioning vertical Z-axis

IMPORTANT. Don't use the Space Navigator if the software moves the positioning system (e.g., during active script execution), in order to not interrupt movements.

11.12.5.2 POSITIONING X-AXIS

To control the X-axis movement, slide the Space Navigator's control in the desired direction. Push the controller to the left, to move the X-axis leftwards and push it to the right to move the axis rightwards.

Figure 255: Positioning X-axis with Space Navigator

The speed of the axes can be determined by the strength of the deflection of the Space Navigator cap. The harder you push or twist the cap, the faster the axes are moving. By a slight deflection of the cap, you can very finely and precisely control axis speeds.

HINT. The more you deflect the Space Navigator cap, the higher are the axes speeds.

11.12.5.3 POSITIONING Y-AXIS

To move the Y-axis towards the back of the device, press the controller forward. To move the positioning system towards the front of the device, pull the controller towards you.

Figure 256: Positioning Y-axis with Space Navigator

HINT. You can control both directions at the same time if you move the controller forward or backward and to the left or right at the same time.

11.12.5.4 VERTICAL AXIS POSITIONING

To position the Z-axis, press down the cap or pull it to the top. The Z-axis is moved in the direction in which you move the control element of the Space Navigator.

11.13 Single Axis Control

In addition to the control of multi-axis systems the software also supports control of individual axis devices. This may be necessary if your axis system has additional tool axes (like second Z-axis or rotation axis) which are not part of the XYZ-multi-axis control.

You can show / hide the window for single axis control via the main menu item *Window → Single Axis Control* (figure below).

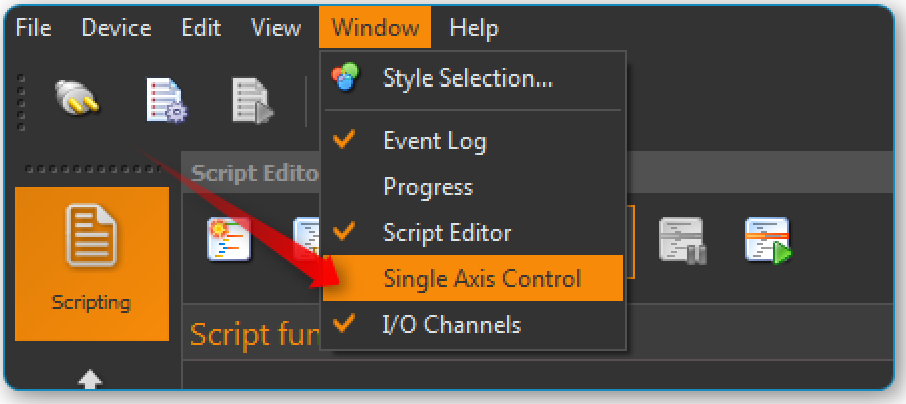


Figure 257: Display of single axis control panel

HINT. The single axis control window is a tool window and can be freely placed in the application window and docked in all four side areas.

The following control elements are available in the single axis control window:

1 Drop-down list *Axis Device* for selection of axis device to control.

2 In the *Position* area a progress bar shows the actual position in the range between minimum and maximum axis position. You can move the axis by pressing the plus / minus buttons. The movements continues as long as the buttons are pressed and stops immediately if a button is released.

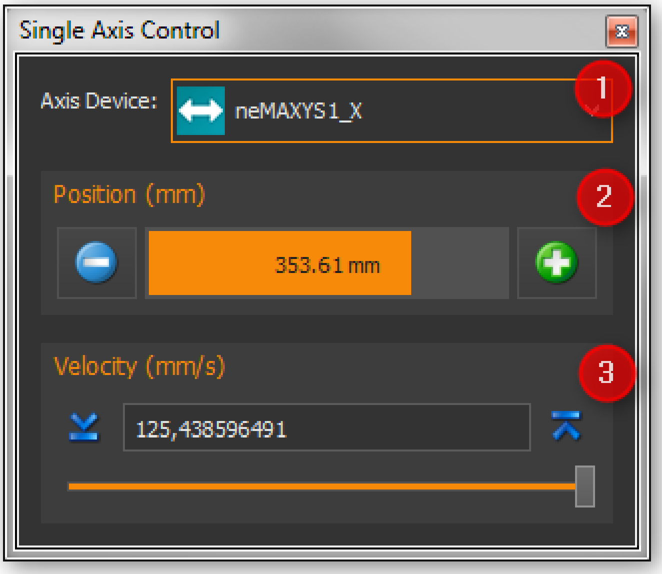
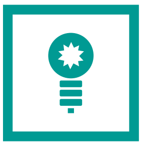


Figure 258: Single axis control panel

3 In the *Velocity* area you can adjust the speed of axis move.



HINT. With the Move Axis script function you can control individual axes from a script program.

11.14 Additional Functions

11.14.1 Calibrating axis system

If the travel path of an axis is blocked (for example in the event of a collision), the stepper motors in the device may "lose" steps, so that the position values are no longer displayed correctly. In this case, the axis system has to be calibrated. To start calibration, select the point in the main menu *Device* → *rotAXYS* -> *Find Home* (see Figure below). If you have a neMAXYS system, you can reach the calibration via *Device* → *neMAXYS* -> *Find Home*

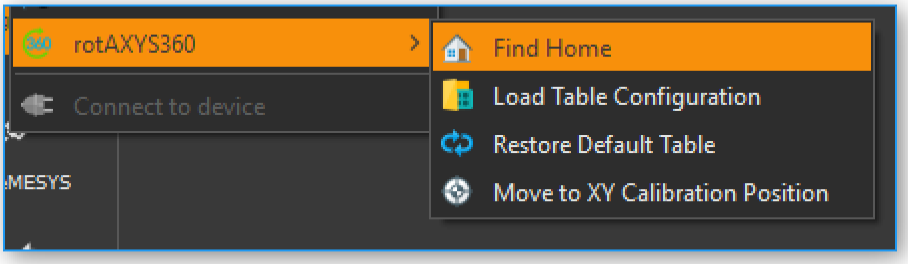


Figure 259: Calibrate positioning system

11.15 Positioning Script Functions

11.15.1 Introduction

The rotAXYS / neMAXYS plugin offers a number of script functions required for integrating the cetoni positioning systems in the Qmix scripting system. The following script functions are available:

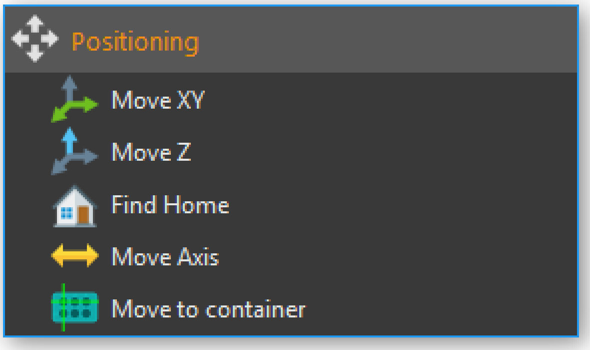
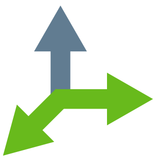


Figure 260: Positioning Script Functions

11.15.2 Move XY



This function is used in a Qmix script to position an XY axis system in the XY plane. You can select an axis system 1 and the target position (X and Y positions) 2 in the configuration area of this function. Alternatively, you can move the axis system manually to a specific position (for example with the positioning map - see Section 11.5), and then assign the current position values to the script function with the *Capture Actual Position* button (see Figure below).

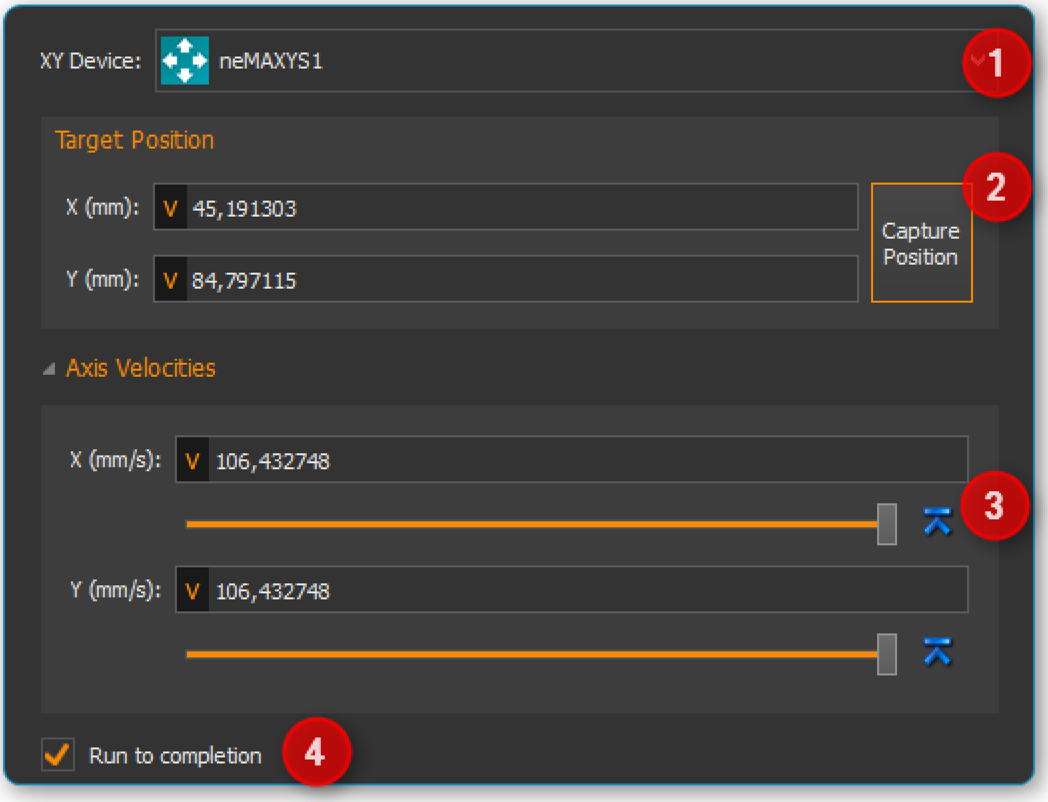


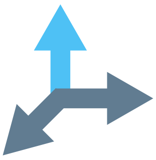
Figure 261: Move XY script configuration

If you make no further adjustments, all positionings are carried out with the maximum possible speed. If you want to change velocities, you need to expand the *Axis Velocities* control by clicking it. This control panel 3 .contains all control elements for entering velocity values for each single axis.

Use the *Run to completion* 4 check box to determine whether the next function

- will start immediately after the current positioning move started (*Run to completion* not checked)
- or if the next function is called after the target position has been reached (*Run to completion* checked).

11.15.3 Move Z



This function can be used to position the vertical axis (Z-axis) of an axis system. You can enter the axis system 1 and the position parameters (Z position 2 and velocity 3) in the configuration area of this function. Alternatively, you can move the Z-axis manually to a specific position (for example with the Z panel - see Section 11.6), and then assign the current Z position to the script function with the *Capture Actual Position* button.

Figure 262: Move Z script configuration

If you make no further adjustments, the positioning is carried out with the maximum possible speed. If you want to change velocities, you need to expand the *Velocity* control by clicking it. Another click on this element hides the velocity controls again.

Use the *Run to completion* 4 check box to determine whether the next function

- will start immediately after the current positioning move started (*Run to completion* not checked)
- or if the next function is called after the target position has been reached (*Run to completion* checked).

HINT. If **Run to completion** is not activated, you can move several axes simultaneously. You can e.g. start positioning the XY-system, and while the XY-axes are moving, you can perform positioning with the Z-axis.

IMPORTANT. The input fields for configuring the speeds are hidden by default, and must be displayed by clicking the **Velocity** label.

11.15.4 Find Home

This function allows you to start a reference move from the script. The positioning system moves to its reference position and initialises its distance measuring system. The *Run to Completion* parameter allows you to select whether the function should be terminated after the start of the reference move or when the reference move has been completed.

11.15.5 Move Axis

This function can be used to position a single axis device. Use the drop-down list *Axis device* 1 to select the axis to me moved. You can enter the parameters (position 2 and velocity 3) in the configuration area of this function. Alternatively, you can move the axis manually to a specific position (for example with the Single Axis Contol Panel – see Section 11.13), and then assign the current position to the script function with the *Capture Position* button.

Figure 263: Move Axis script configuration

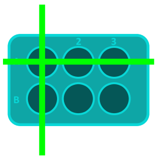
If you make no further adjustments, the positioning is carried out with the maximum possible speed. If you want to change velocity, you need to expand the *Velocity* control by clicking it. Another click on this element hides the velocity controls again.

Use the *Run to completion* 4 check box to determine whether the next function

- will start immediately after the current positioning move started (*Run to completion* not checked)
- or if the next function is called after the target position has been reached (*Run to completion* checked).

IMPORTANT. The input fields for configuring the speeds are hidden by default, and must be displayed by clicking the **Velocity** label.

11.15.6 Move to container



This function is used to move an axis system in the XY plane to the centre a specific container (beaker, fluid tank, cavity of a well plate).

First you select the positioning device in the *XY Device* combo box 1 (figure below). Then you select the target container in the container selection combo box 2. If the container has several cavities (e.g. the wells of a well plate), enter the column and row of the cavity in the *row* and *column* input fields 3. The numbering starts at zero. That means, for a microwell plate with 12 columns and 8 rows, you can enter values from 0 to 11 in the column input field and values from 0 to 7 in the row input field. If the container has only one single cavity (e.g. beaker or fluid tank) the input fields for *row* and *column* are greyed out.

HINT. You can use script variables in the row and column input fields. This allows you to process all wells of a well plate in a loop using the loop counter variables.

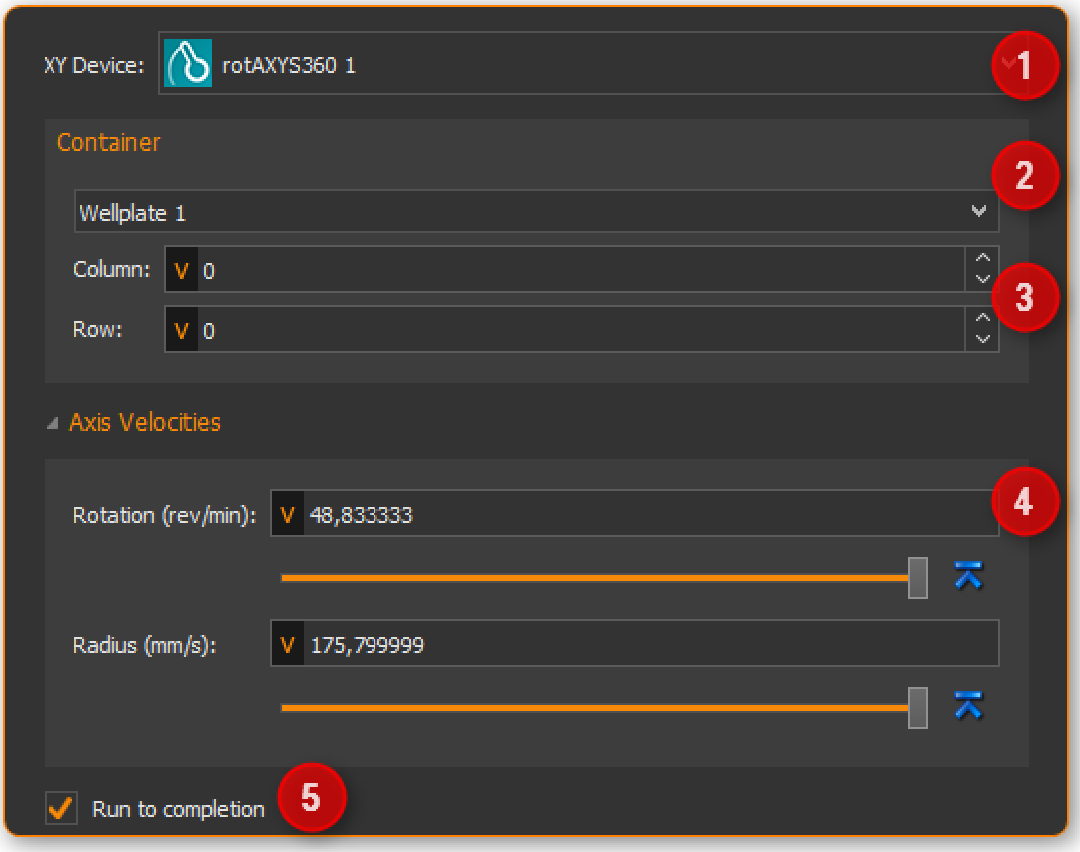


Figure 264: Move to container configuration

If you make no further adjustments, all positionings are carried out with the maximum possible speed. If you want to change velocities, you need to expand the *Axis Velocities* control by clicking it. This control panel 4 contains all control elements for entering velocity values for each single axis.

Use the *Run to completion* 5 check box to determine whether the next function

- will start immediately after the current positioning move started (*Run to completion* not checked)
- or if the next function is called after the target position has been reached (*Run to completion* checked).

12 Tubing Pump Plugin

12.1 Introduction

Press the button *Tubing Pump* in the side bar to switch to the operating window (Work bench) of the tubing pump plug-in.

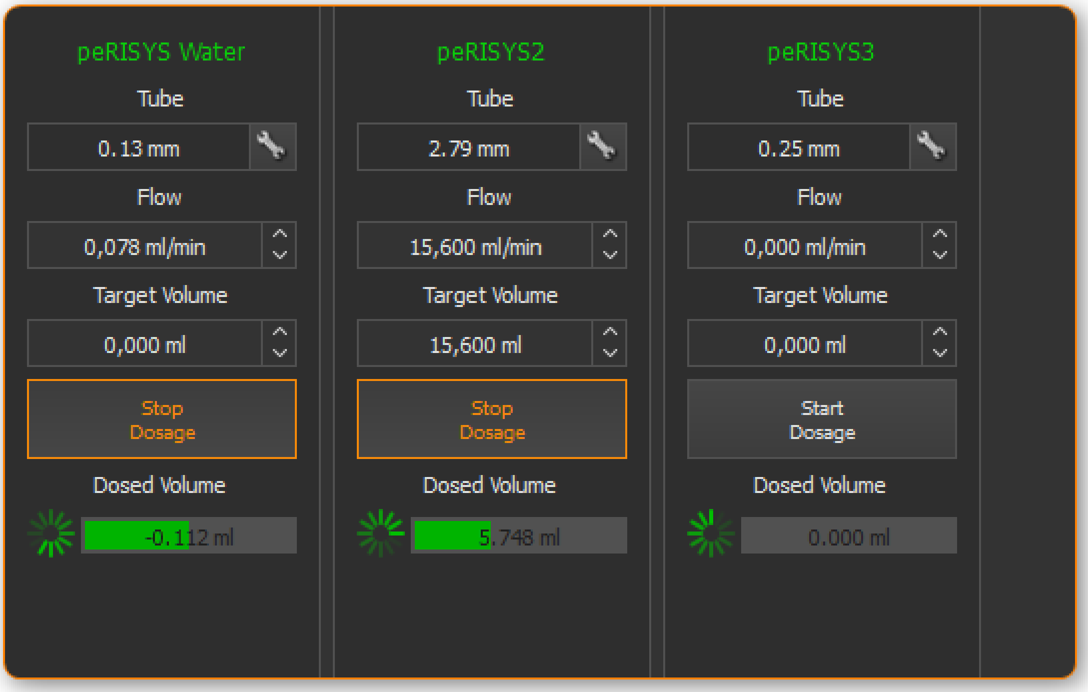
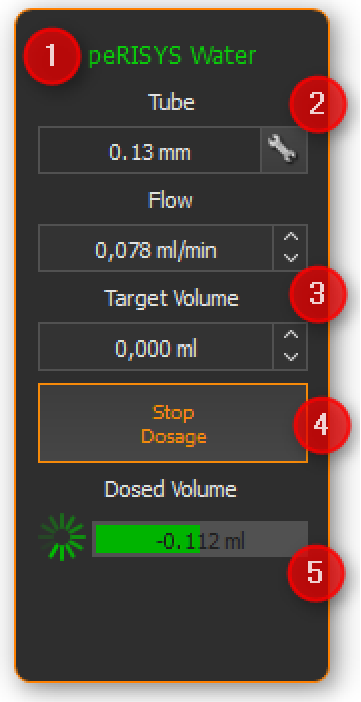


Figure 28: Tubing pump work bench.

The work bench shows the operating panels for all tubing pumps.

12.2 Tubing Pump Operating Panel

12.2.1 Operating Parameters



4 Pump name (customizable).

5 Tubing configuration (inner diameter).

6 Target values (flow rate, volume).

7 Start/stop dosing button.

8 Activity and dosed volume indicators.

12.2.2 Pump Name Customization

You can change the description of each pump at any time to reflect the function or fluids of the respective pump. To change the description, directly click the description 1 and type in the new name.

12.2.3 Tubing Configuration

The input box *Tube* 2 shows the name of the currently configured tubing.. Click on the wrench symbol right of the input area to change the tubing configuration. This will open the *Tube Selection* dialog (figure below).

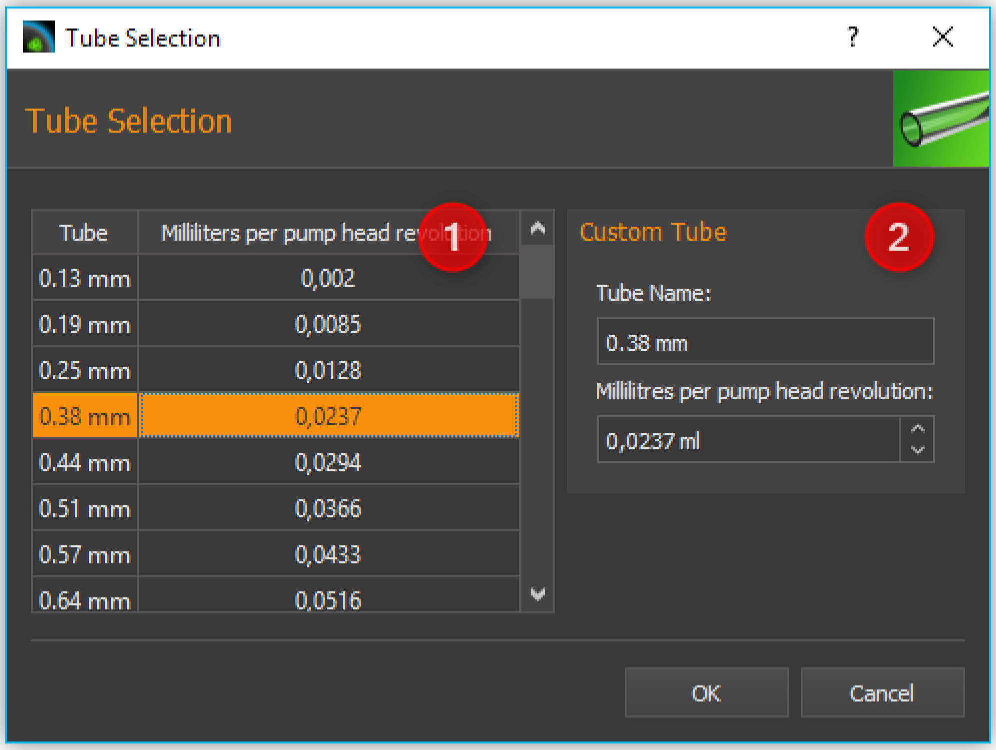


Figure 29: Tubing selection dialog.

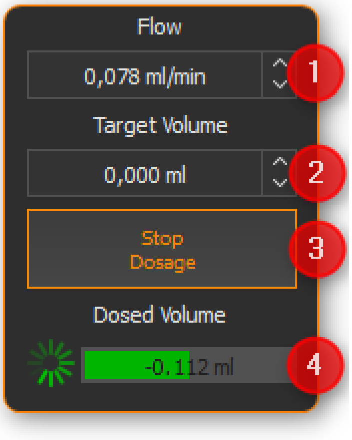
In the Tube Selection dialog you can either choose from a list of available tubes 1, or define your own tube in the *Custom Tube* section 2.

To select an existing tube, simply click on it in the list 1 and confirm the selection by clicking *OK*. In the list, the name of each tube is displayed in the first column (*tube*). The value in the second column (*Milliliters per pumphead revolution*) indicates how many milliliters are pumped during a complete revolution of the pump head.

If you want to configure your own tube, you can do this in the *Custom Tube* 2 section. Enter a name for the tube and the millilitres that are pumped through this tube during one revolution of the pump head. Confirm the configuration by clicking *OK*.

12.3 Manual Dosing

Proceed as follows to configure a manual dosing task:



1 Enter your target flow rate in the *Flow* input box. A negative flow value causes a change in the rotating direction of the pump head. So you can switch between dispensing and aspiration by switching the sign of the entered flow value.

2 Now enter the volume to be dosed in the input box *Target Volume*. If you set the target volume to zero, the pump will be working in flow mode, i.e., the pump will work until manually stopped. A negative volume causes a change in the rotating direction of the pump head.

3 Start the pump by clicking *Start Dosage* and stop it by clicking the same button again.

4 The *Dosed Volume* info box will show the progress of the pumping process.

HINT. The software treats a dosing volume set to zero as unlimited continual flow. This is, once started, the pump will continue to operate until it is manually stopped by the user.

HINT. You can switch between dispensing and aspiration by switching the sign of the entered flow or volume value.

12.4 Script Functions

12.4.1 Introduction

The plugin offers a number of script functions which can be used to program automatic sequences or for time-controlled dosing of liquids. The following script functions are available:

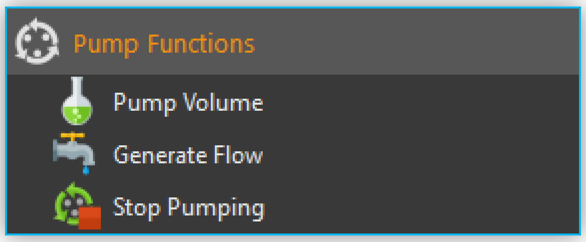
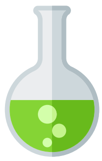


Figure 30: Pump script functions

12.4.2 Pump Volume



With this function, you can pump a specific volume at a precisely defined flow rate. You can set all the parameters in the configuration area, for example the dosing module 1, the volume to be dosed, and the flow rate 2.

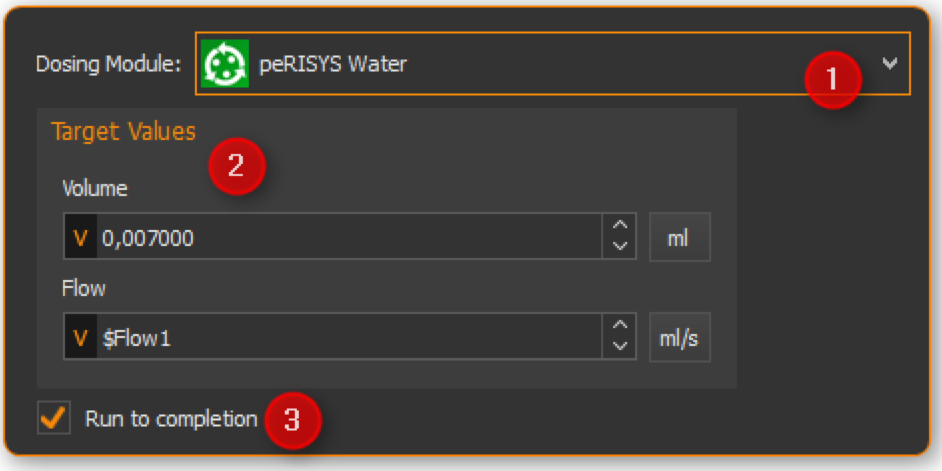


Figure 31: Pump Volume Script Parameter

You can also activate or deactivate the *Run to completion* 3 parameter in the configuration area.

When *Run to completion* is activated, the script execution is not continued until the complete volume has been dosed and the dosing process has ended. If this parameter is not active, the dosing is started, and then the next script function is executed immediately. This enables you, for example, to start a number of dosing modules almost simultaneously.

HINT. All the pump functions support the use of variables. That means, in all input fields marked with a coloured V in the script configuration panel (e.g. flow rate and volume) you can enter variables.

12.4.3 Generate Flow

This function is used to generate a constant flow rate. In the configuration area, you can select the dosing module and set the flow rate. If the *Run to completion* parameter is active, the next script function is not executed until the module has stopped or reached one of the limit positions.

12.4.4 Stop Pumping

You can immediately stop an active dosing process of a pump with this function.

13 LED Array Plugin

13.1 Introduction

Press the button *LED Array* in the side bar to switch to the operating window (workbench) of the LED Array plug-in (figure below).

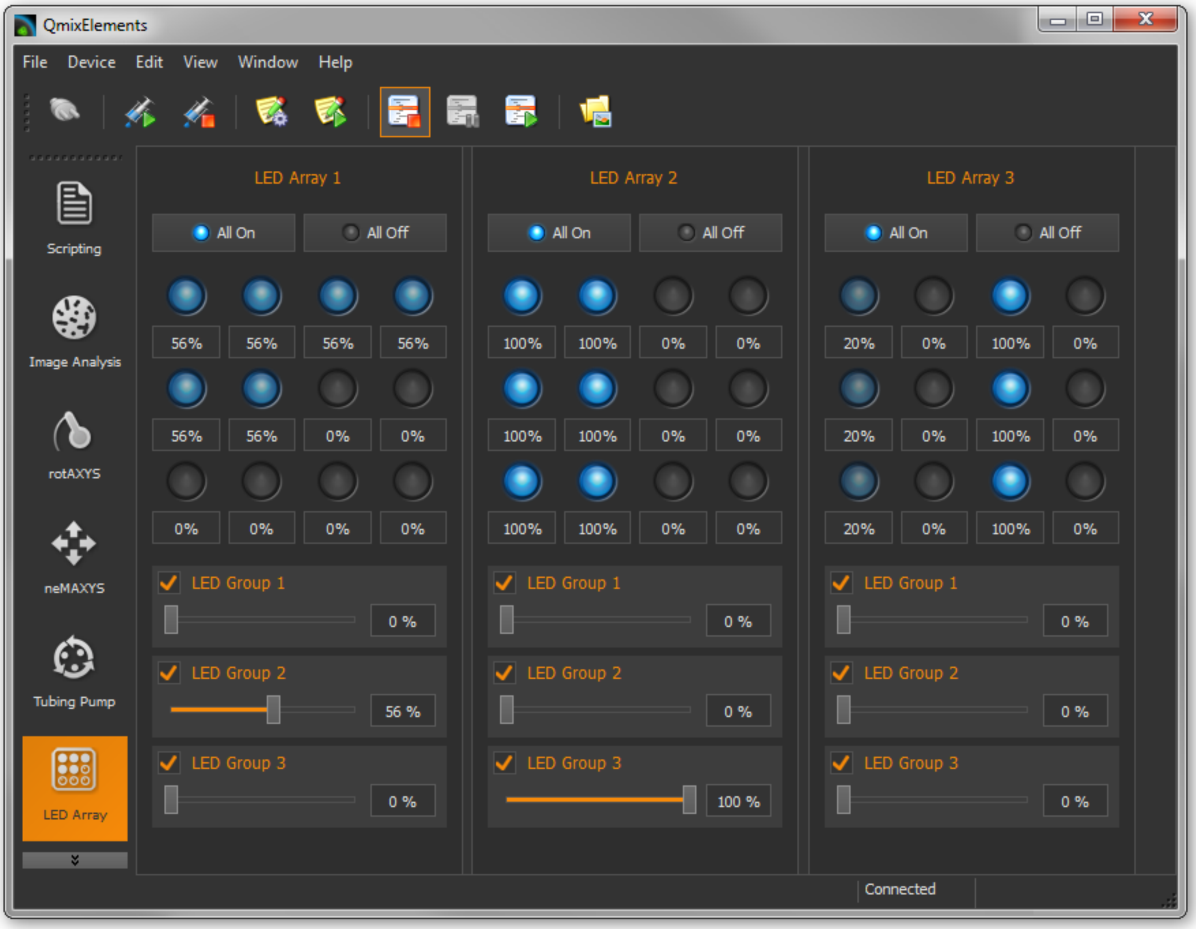
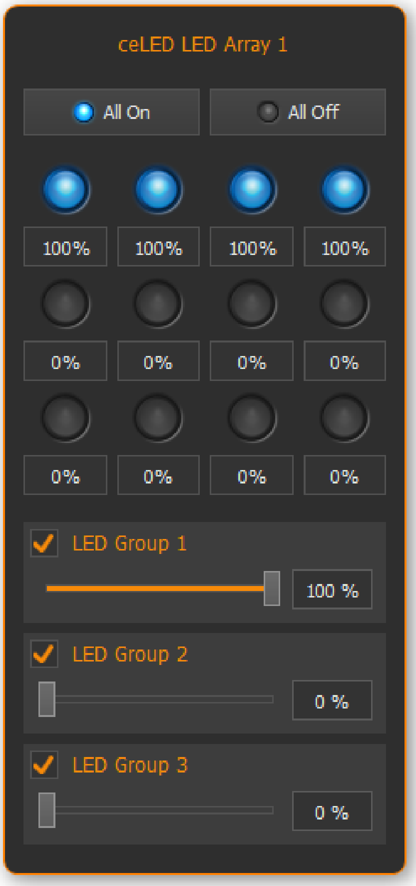
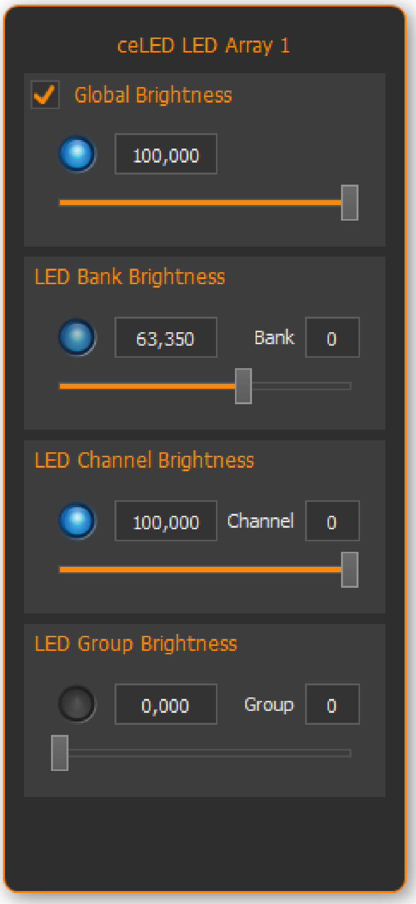


Figure 32: LED Array work bench.

The workbench contains a separate control panel for each connected LED array device:

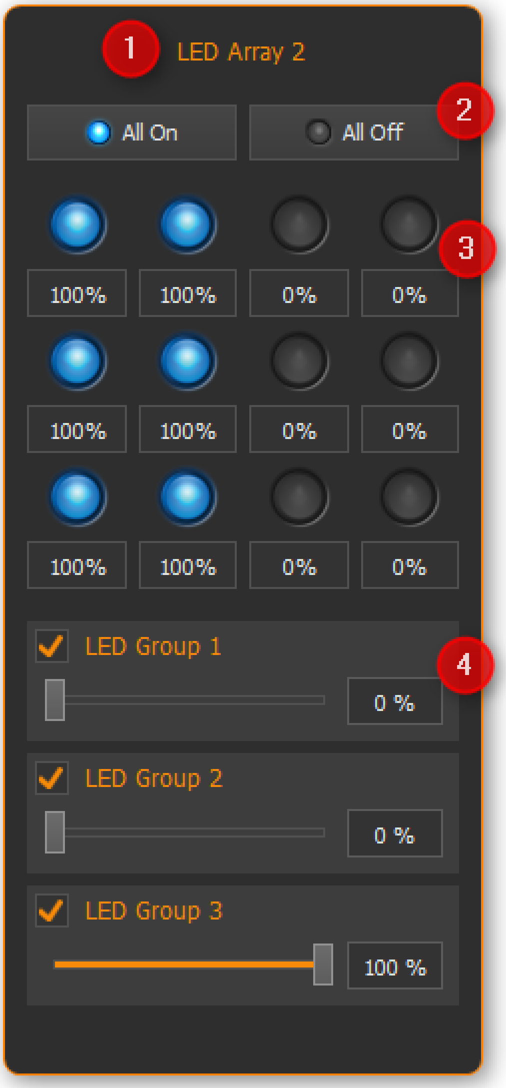
13.2 Hardware Version

Depending on the LED array hardware version a corresponding LED array control panel is displayed for every LED array. The following two hardware versions are supported:

LED Array Hardware V1	LED Array Hardware V2
	
Hardware version 1 supports 12 independent LED array channels with a resolution of 100 steps to adjust the brightness.	Hardware version 2 supports a virtually infinite number of independent LED channels with a resolution of 4096 steps to adjust the brightness.

13.3 LED Array Control Panel V1

13.3.1 Control Elements



5 Caption (customizable).

6 all LED channels on / off

7 adjust the brightness of individual LED channels (0 - 100%)

8 adjust the brightness of multiple LED channels (LED group) at the same time

13.3.2 Caption Customization

You can change the caption of each LED array control panel at any time. To change the caption, directly click the caption label 1 and type in the new name. This name will be saved and reloaded the next time you start the software.

13.4 Setting brightness of individual LED channels

You can set the brightness of each LED channel by entering the value directly in the field below the LED 2 (0 - 100%) or by clicking the right mouse button on an LED and dragging the brightness slider 1 (see figure below).

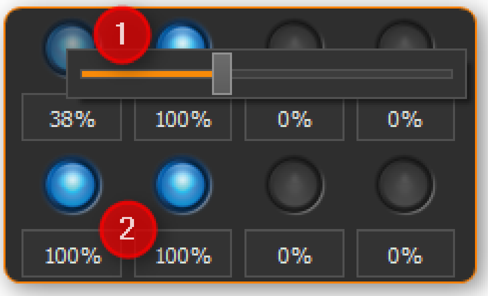


Figure 33: Setting brightness of individual LED channels

Click with the left mouse button on an LED to switch individual LED channels on and off.

Figure 34: Switching LED channel on / off

As all LED channels are normal analog output channels in the QmixElements software, you can also adjust the brightness of individual LED channels via the *I/O Channels* window (figure below).

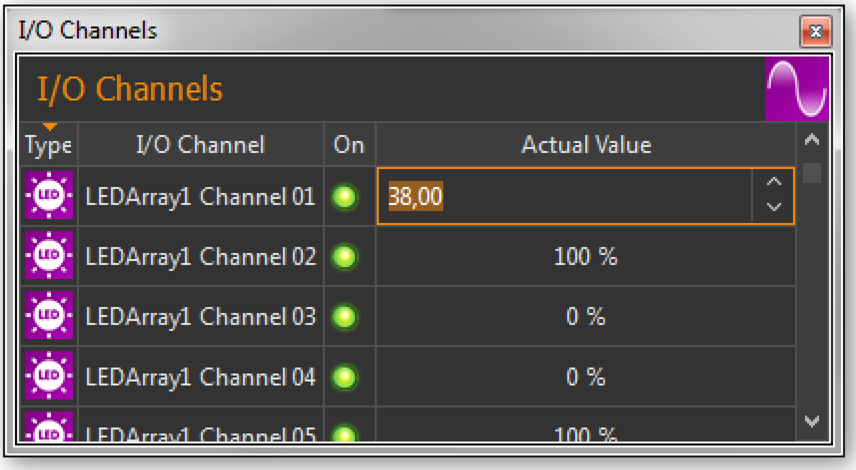


Figure 35: Setting LED brightness in I/O Channel window

13.5 LED Group Control

13.5.1 Introduction

When multiple LED channels should work synchronously, you can combine these channels into groups and control them together. The control panel contains three sliders to adjust brightness of three different LED channel groups.

13.5.2 Configuring LED channel groups

To select the channels to be combined into a group, click the right mouse button in the control area of a certain group to show its context menu. Then select the menu item *Configure Group Channels* (see figure below).

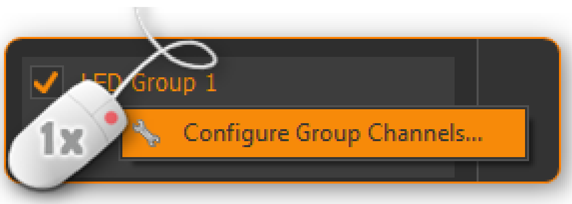


Figure 36: Show LED group configuration dialog

In the LED group configuration dialog (figure below) you can select all LED channels to be grouped together. Check each channel to be part of this group 1 and confirm your selection by clicking *OK* 2.

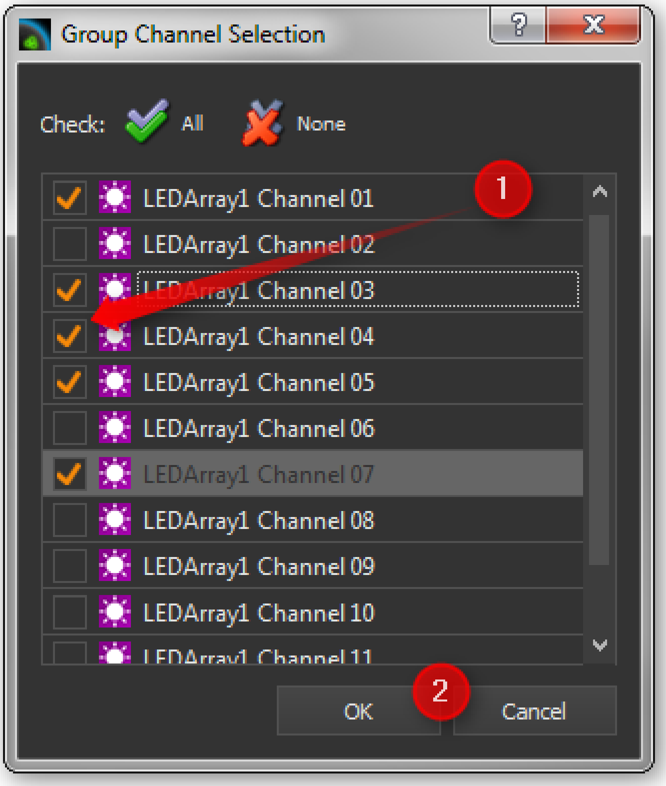


Figure 37: LED group configuration dialog

13.5.3 Controlling LED groups

You can change the brightness of an LED group with the slider 1 or by entering the brightness value in the input box 2 (see figure below).

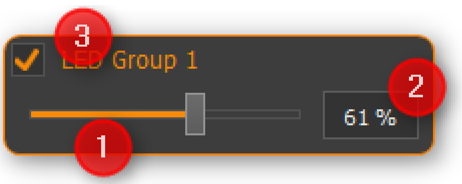


Figure 38: LED group control

With the check mark in the upper left corner 3, you can switch all LED channels in this group on /off simultaneously.

13.6 Configuring Standby-Timer

The LED Array has a standby timer functionality. This means, all sectors of the LED Array are automatically switched off after the last action of the user in the software and after the standby time is over. Every change in parameters in the software resets the standby timer and reactivates the array.

IMPORTANT. The standby timer value is saved in the LED Array. I.e. even if the connection to the PC is lost, the LED array is switched off after the standby time is over.

To configure the standby time, click with the right mouse button on a blank area in the LED array control panel to display the context menu. Then select the menu item *Configure Standby Timer* (figure below).

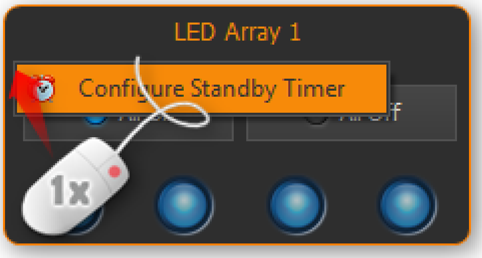


Figure 39: Show standby timer configuration window

In the configuration dialog displayed (figure below), you can configure the hours, minutes and seconds of the standby timer. Your changes are accepted by clicking on *OK* and the standby time will be saved in the device.

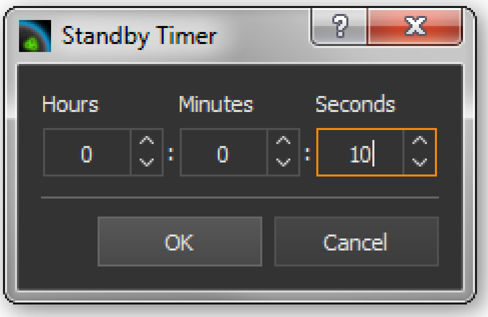
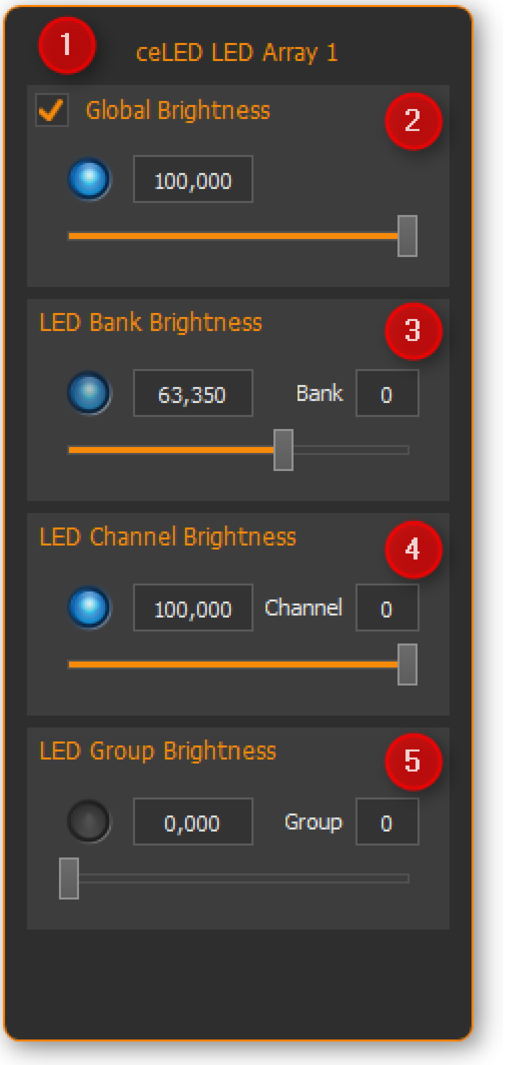


Figure 40: Standby timer configuration window

IMPORTANT. Enter a value of 0 for Hours, Minutes and Seconds to deactivate the standby timer..

13.7 LED Array Control Panel V2

13.7.1 Control Elements



1 Caption (customizable)

2 adjust the global brightness of all channels and switch the Enable signal

3 adjust the brightness of a complete bank

4 adjust the brightness of individual LED channels (0 – 100%)

5 adjust the brightness of a user-specific LED group consisting of multiple channels

13.7.2 Caption Customization

You can change the caption of the LED array at any time. To change the caption, click the caption label 1 and enter a new name for the LED array. This name will be saved and reloaded the next time you start the software.

13.8 Setting the Global Brightness

You can set the global brightness of all LED channels at the same time by using the slider 2 or the input field of *Global Brightness*. With the check mark in the top left corner 1 you can switch the global Enable signal of the LED array. The array can be switched on and off this way without changing the brightness of the individual channels.

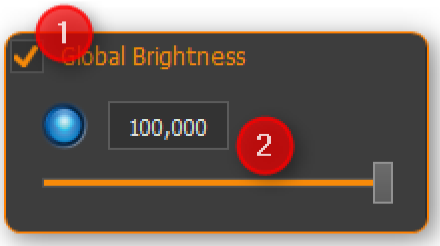


Figure 41: Globale brightness and Enable Signal

IMPORTANT. The Enable signal must have been activated for the LEDs of the array to light up.

13.9 Setting the Brightness of LED Banks

LED banks are device-specific and hardware-specific groups of individual LED channels in groups. LED banks group LED channels that also form a physical group in the hardware, e.g. all LEDs on one board or all LEDs of a certain type (e.g. warm white or cold white). These groups are anchored in the firmware and cannot be changed by the user.

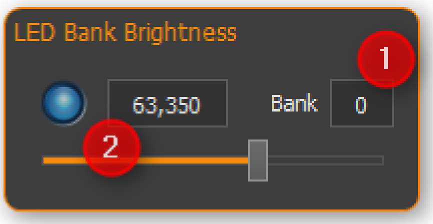


Figure 42: Control elements for LED bank brightness

To change the brightness of a bank, select the bank in the *Bank* 1 input field and then set the brightness of the selected bank using the slider or input field 2.

13.10 Setting the Brightness of Individual LED Channels

In *LED Channel Brightness* you can set the brightness of individual LED channels. For this purpose select the channel in the *Channel* 1 input field. You can change the brightness by using the input field 2 or the slider.

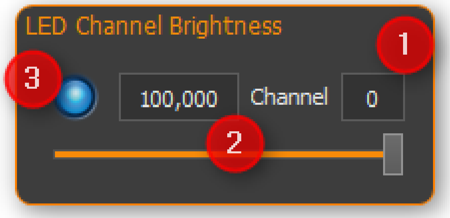


Figure 43: Changing the brightness of individual LEDs

To switch a channel on or off, click the left mouse button on the relevant LED 3.

As all LED channels are analogue output channels in the QmixElements software you can also change the brightness of individual channels via the I/O Channels window (see figure below).

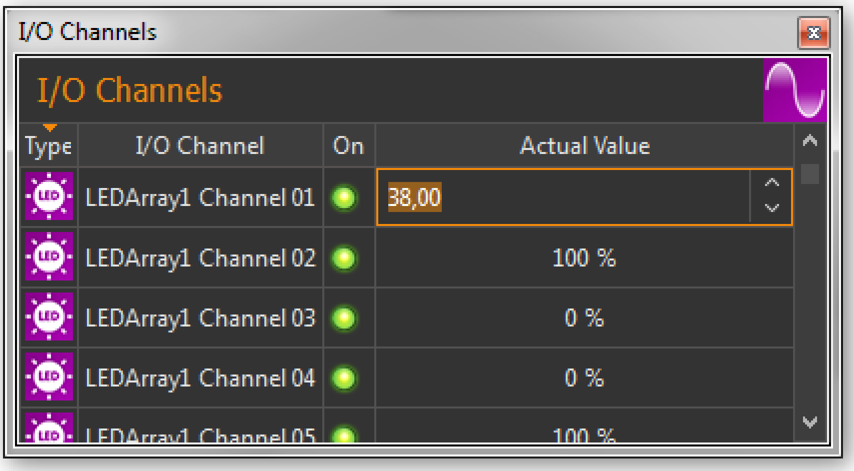


Figure 44: Window of I/O channels

13.11 LED Channel Groups

13.11.1 Introduction

When multiple LED channels should work synchronously, you can combine these channels into groups and control them together. The group channels can then be used to jointly control all channels of an LED group.

13.11.2 Configuring the LED channel groups

To configure LED channel groups, use the right mouse button and click on *LED Group Brightness*. Then select the menu item *Configure LED Groups* from the context menu.

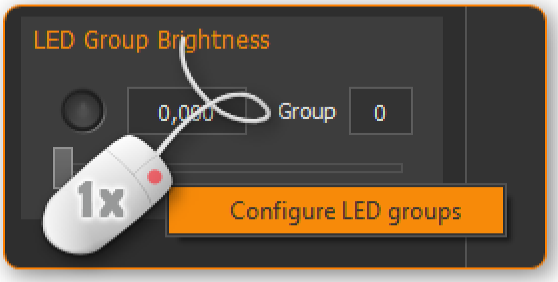


Figure 45: Activate LED group configuration

The dialogue for the LED group configuration will be shown now (see figure below).

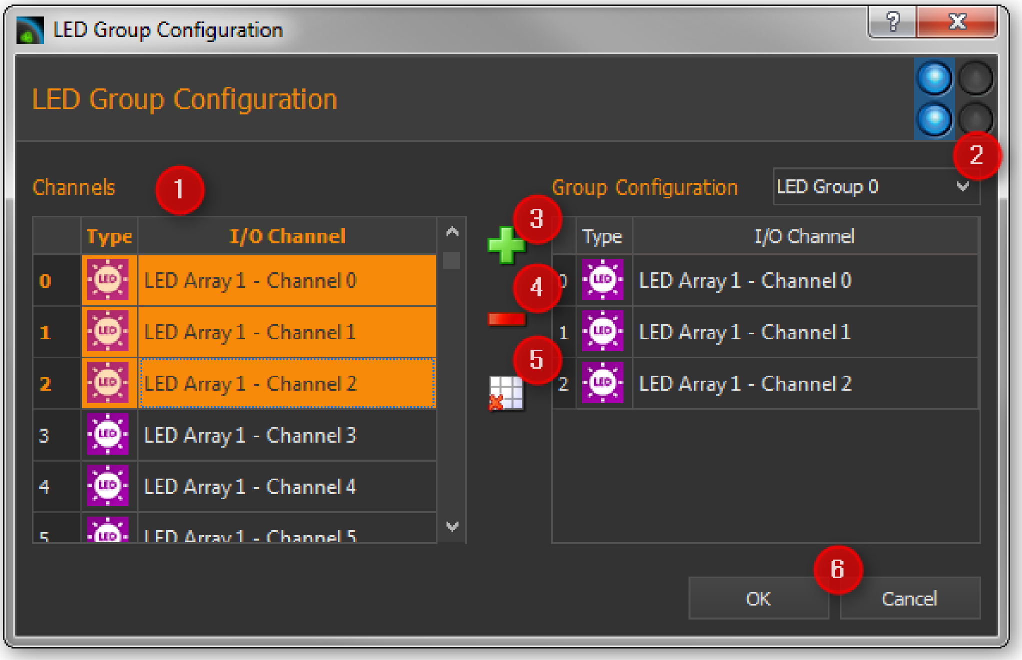


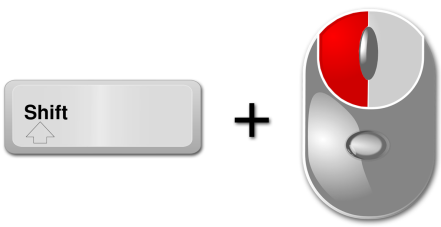
Abbildung 46: Konfigurationsdialog für LED Kanalgruppen

To add LED channels to a group, proceed as follows:

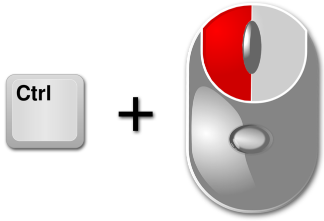
(6) First, go to the group list box 2 and select the group to be configured.

(7) Now, select from the channel list 1 the channels to be added to the group by clicking with the mouse button.

- Select individual channels by clicking.
- Select multiple connected channels by clicking with the mouse on the first channel on the first channel. Keep the *shift key* pressed while clicking on the last channel.



- Multiple independent channels can be selected by keeping the control key pressed while clicking.



(8) Now, add the selected channels to the group by clicking the *Plus button* 3. To delete individual channels from the group, select the channels from the group list and then click the Minus button 4. To delete all channels from the group, click the *Clear LED Group* 5 button..

(9) After having configured all groups click *OK* 6. The group configuration will then be transmitted to the device. If you want to permanently save the group configuration in the device, click Yes in the message window displayed (see figure below).

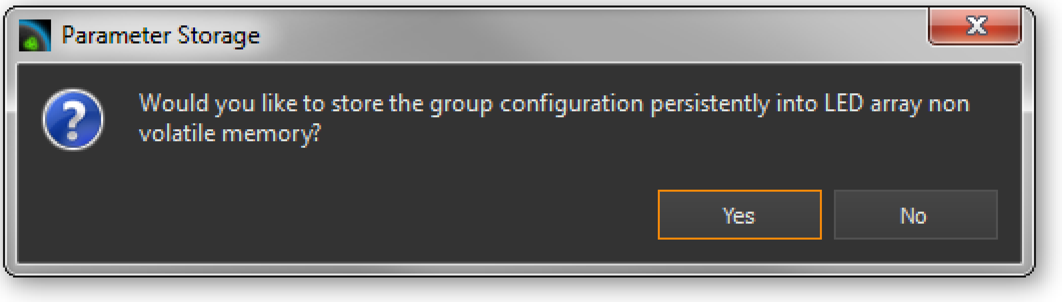


Figure 47: Save group configuration permanently in device

13.11.3 Setting the brightness of LED groups

To change the brightness of a group, select the group from the *Group* 1 input field and then set the brightness of the selected group using the slider or the input field 2.

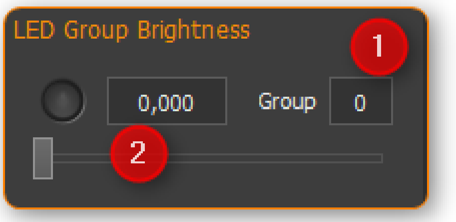


Figure 48: Control elements for the LED group brightness

13.12 Script Functions

13.12.1 Introduction

The LED Array Plugin provides various script functions that can be used to program automated exposure sequences or for time-controlled exposure. The script functions are available in the categories of *LED Array Functions* and *I/O Functions*.

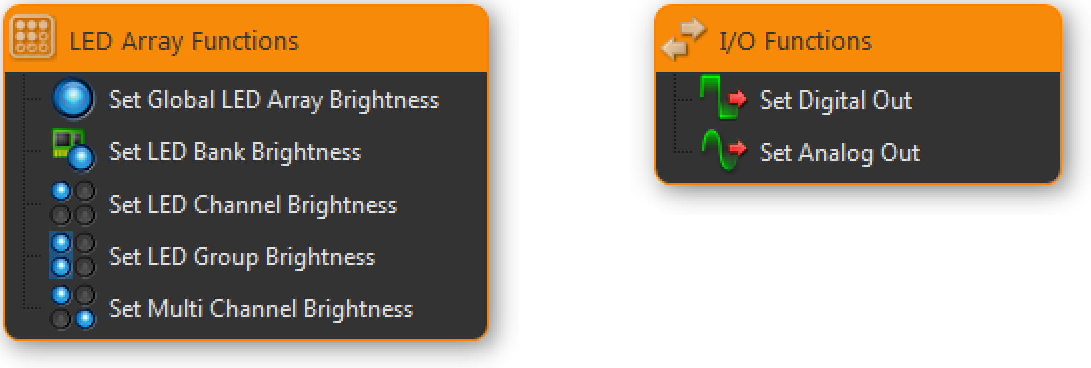


Figure 49: LED Array script functions

13.12.2 Global Brightness Function – *Set Global LED Array Brightness*

The global brightness of all LED channels of an LED array can be set jointly by this script function. To configure the script function, proceed as follows (see figure below):

- 1 First, select the LED device from the list of devices.
- 2 Then, set the brightness (0 – 100%).

HINT. This function supports script variables. Variables can be used in the Brightness field.

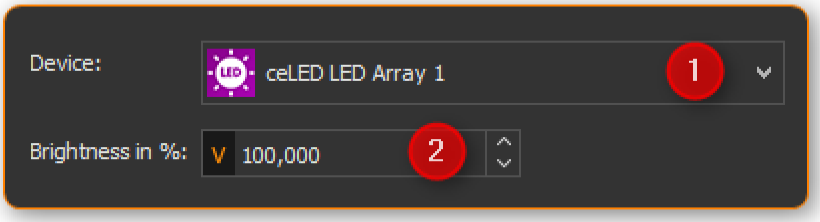
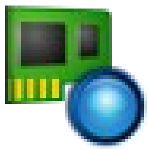


Figure 50: Set global LED Array Brightness script function

13.12.3 Bank Brightness function – *Set LED Bank Brightness*



You can set the brightness of entire LED banks by this script function. To configure the script function, proceed as follows (see figure below):

- 1 Select the LED Array device.
- 2 Select the bank the brightness of which is to be changed.

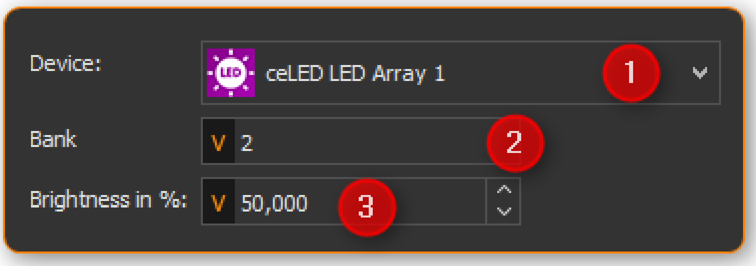
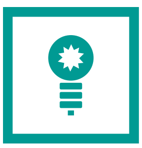


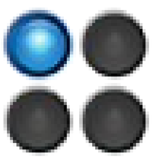
Figure 51: Set LED Bank Brightness script function

- 3 Set the brightness (0- 100%).



HINT. This function supports script variables. Variables can be used in the Bank and Brightness field.

13.12.4 LED brightness function – Set LED Channel Brightness



This script function can be used to set the brightness of individual LED channels. To configure the script function, proceed as follows (see figure below):

- 1 Select the LED Array device.
- 2 Select the channel the brightness of which you want to set.

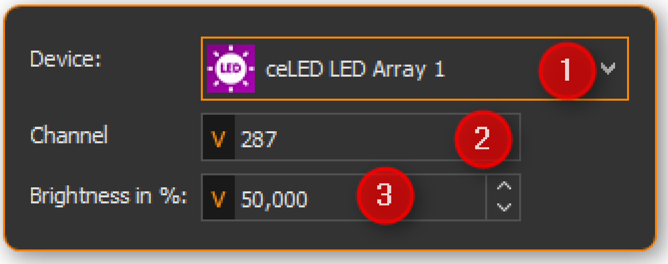
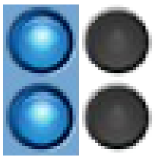


Figure 52: Set LED Channel Brightness script function

- 3 Set the brightness (0- 100%)

HINT. This function supports script variables. Variables can be used in the Channel and Brightness field.

13.12.5 Group brightness function – Set LED Group Brightness



You can use this function to set the brightness of an LED group in a script-controlled manner. To configure the script function, proceed as follows (see figure below):

- 1 Select the LED Array device.
- 2 Select the group the brightness of which you want to set.

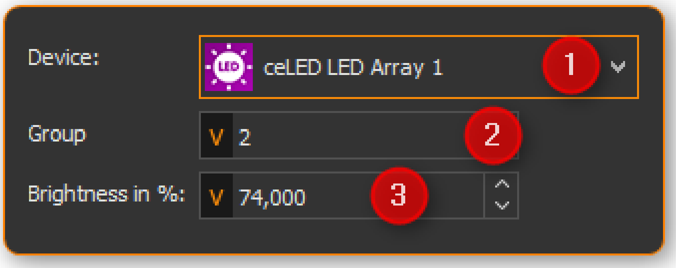
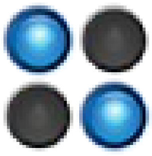


Figure 53: Set LED Group Brightness script function

- 3 Set the brightness (0- 100%).

HINT. This function supports script variables. Variables can be used in the Channel and Brightness field.

13.12.6 Multi-channel brightness function – *Set Multi Channel Brightness*



You can use this function to set simultaneously the brightness of multiple LED channels in a script-controlled manner. To configure the script function, proceed as follows (see figure below):

- 1 Select the LED Array device.
- 2 Set the brightness (0- 100%).
- 3 Check the every channel of the list which is to be controlled.

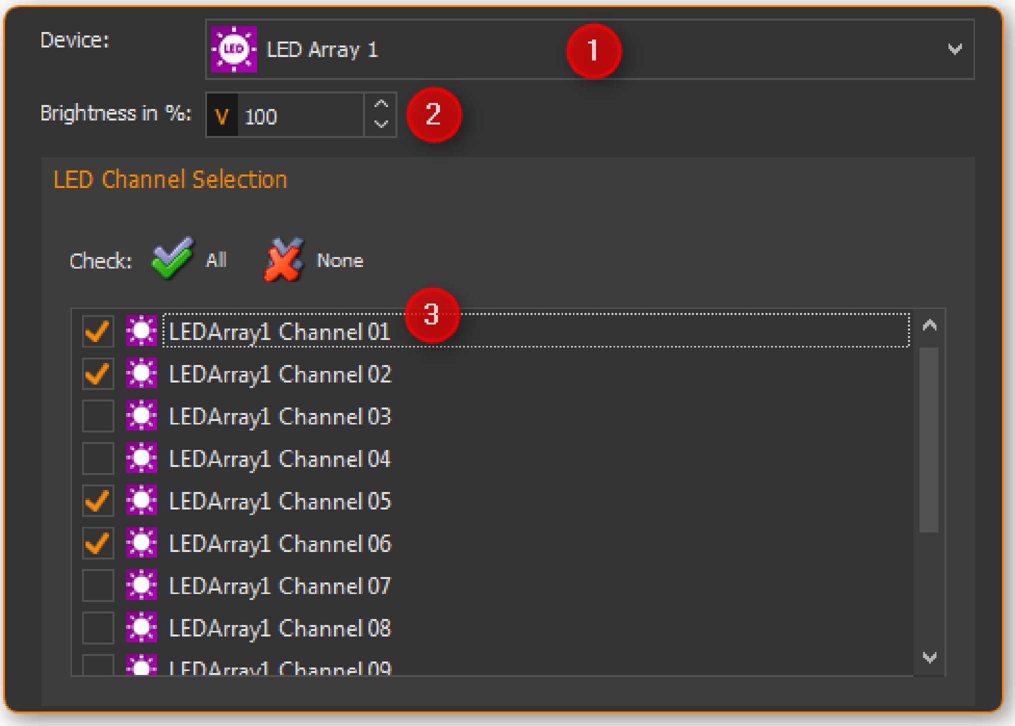
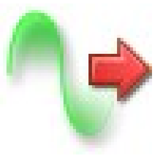


Figure 54: *Set LED Group Brightness script configuration*

IMPORTANT. If a large number of channels have been selected, the data transfer for all channels may take some time so that not all channels will be switched exactly synchronously. If such delay is undesired, use LED groups.

HINT. This function supports script variables. Variables can be used in the Brightness field.

13.12.7 Setting the analog output function - *Set Analog Out*



All LED channels are normal analog output channels in the QmixElements software. To adjust the brightness of individual channels, you can use the *Set Analog Out* function from the *I/O Functions* category.

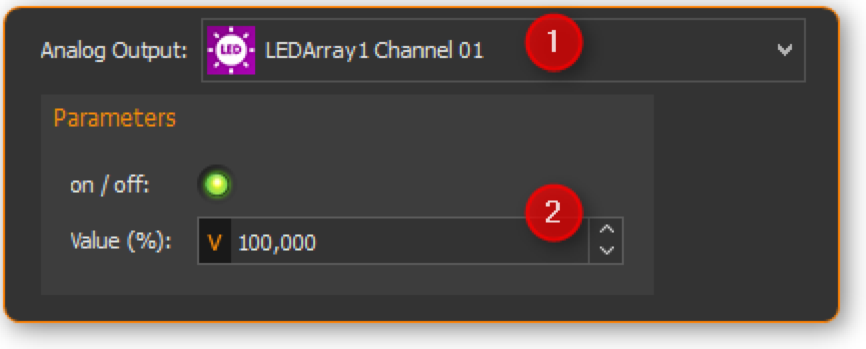


Figure 55: Set Analog Out script configuration

Select the relevant LED channel from the configuration menu (*Analog Output*) 1 and enter the brightness value 2 (0 – 100%) in the *Value* field.

HINT. This function supports the use of variables, i.e. you can enter in the Value field 2 the name of the variable instead of a value which contains the brightness value at the run time of the script (see figure below).

14 Balances Plugin

14.1 Introduction

This plugin is for integrating external laboratory balances into the software. The following devices are currently supported:

MANUFACTURER	SUPPORTED DEVICES
Sartorius	Entris, ED-, GK- and GW-scales

14.2 Configuration of Sartorius Scales

14.2.1 Adjusting the Device Settings

In order for the scale to work optimally with the QmixElements software, please set the following parameters on the device:

14.2.1.1 DEVICE / RS232 MENU

In the DEVICE / RS232 menu, activate the following values marked in red:

Parameter	Setting values	Explanations
DAT.REC.	XBPI	Extended range of commands to control numerous balance functions with binary protocol for direct communication with the device.
	SBI*	Enables SBI communication. The data is output to a PC or control unit. Enables the use of ESC commands from a PC to control the basic balance functions with ASCII protocol.
	REM.DISP.	Enables data output on another display.
	BARCODE	Allows for the connection of an approved barcode scanner.
	YDP20	Sets the standard settings of YDP20 printers.
	YDP30	Sets the standard settings of YDP30 printers.
	OFF	Deactivates the automatic data output.
BAUD	600, 1200, 2400, 4800, 9600*, 19200, 38400, 57600, 115200	Sets the baud rate to the selected value.
PARITY	ODD*	Applies an odd parity.
	EVEN	Applies an even parity.
	NONE	Does **not** apply a parity.
STOPBIT	1 BIT*	Sets the number of stop bits to 1.
	2 BITS	Sets the number of stop bits to 2.
HANDSHK.	SFTWARE	Sets the handshake protocol to software handshake.
	HRDWARE*	Sets the handshake protocol to hardware handshake.
	NONE	Does **not** set a handshake protocol.
DATABIT	7 BITS	Sets the number of data bits to 7.
	8 BITS*	Sets the number of data bits to 8.

* Factory setting

14.2.1.2 DEVICE / USB MENU

In the DEVICE / USB menu, activate the following values marked in red:

Parameter	Setting values	Explanations
DAT.REC.	XBPI	Extended range of commands to control numerous balance functions with binary protocol for direct communication with the device.
	SBI*	Enables SBI communication. The data is output to a PC or control unit. Enables the use of ESC commands from a PC to control the basic balance functions with ASCII protocol.
	REM.DISP.	Enables data output on another display.
	BARCODE	Allows for the connection of an approved barcode scanner.
	YDP20	Sets the standard settings of YDP20 printers.
	YDP30	Sets the standard settings of YDP30 printers.
	OFF	Deactivates the automatic data output.
BAUD	600, 1200, 2400, 4800, 9600*, 19200, 38400, 57600, 115200	Sets the baud rate to the selected value.
PARITY	ODD*	Applies an odd parity.
	EVEN	Applies an even parity.
	NONE	Does **not** apply a parity.
STOPBIT	1 BIT*	Sets the number of stop bits to 1.
	2 BITS	Sets the number of stop bits to 2.
HANDSHK.	SFTWARE	Sets the handshake protocol to software handshake.
	HRDWARE*	Sets the handshake protocol to hardware handshake.
	NONE	Does **not** set a handshake protocol.
DATABIT	7 BITS	Sets the number of data bits to 7.
	8 BITS*	Sets the number of data bits to 8.

* Factory setting

14.2.1.3 DATA.OUT. / COM. SBI MENU

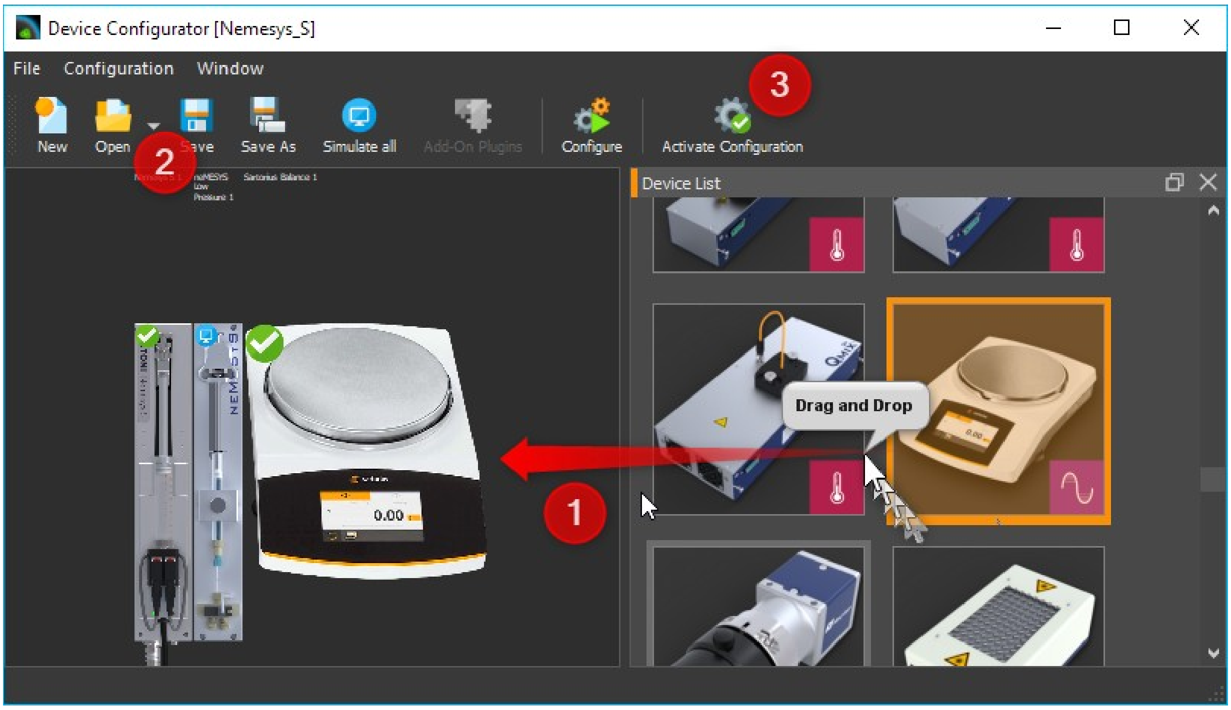
In the DATA.OUT. / COM. SBI menu, activate the following values marked in red:

Parameter	Setting values	Explanations
COM. OUTP.	IND.NO*	Activates the manual data output without stability.
	IND.AFTR	Activates the manual data output after stability.
	AUTO.W/O	Activates the automatic data output without stability.
	AUTO W/	Activates the automatic data output after stability.
STOP.AUT.	OFF*	Deactivates the option to abort the automatic data output.
	ON	The automatic data output is aborted by pressing the [Print] button or a software command.
AUTO.CYCL.	EACH VAL*	Starts the automatic data output with a cycle after each value.
	AFTER 2	Starts the automatic data output with a cycle after every 2nd value.
	INTERV.	The output rate can be set from 0 - 9999 seconds under "INPUT/INTERV.".
FORMAT	22 CHARS*	The data output provides 22 characters per line (16 characters for the measured value and 6 characters for identifiers).
	16 CHARS	The data output provides 16 characters per line for the measured value.
	EXTR.LIN.	The data output provides an additional line with the date, time, and weight value.
AUTO.TAR.	OFF*	Deactivates automatic taring after data output.
	ON	The device automatically tares after data output.

* Factory setting

14.3 Adding a scale to a CETONI Elements device configuration

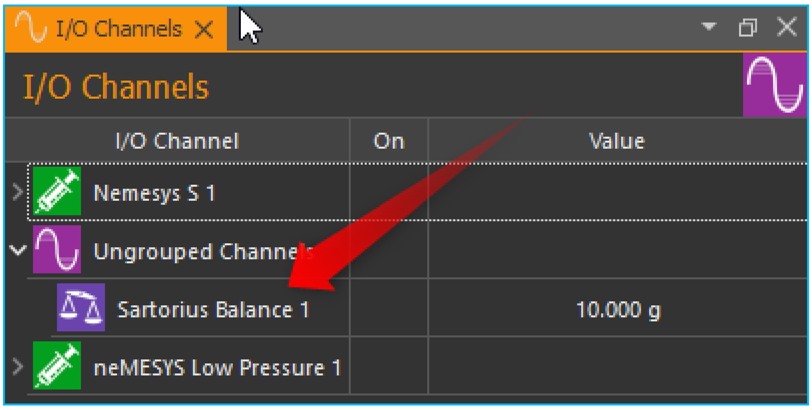
In the *Device Configurator*, simply add a scale to the current device configuration using drag & drop or double-click 1. Then save the device configuration 2 and activate it by clicking the *Activate Configuration* button 3.



14.4 Operation

14.4.1 Display / Read the values

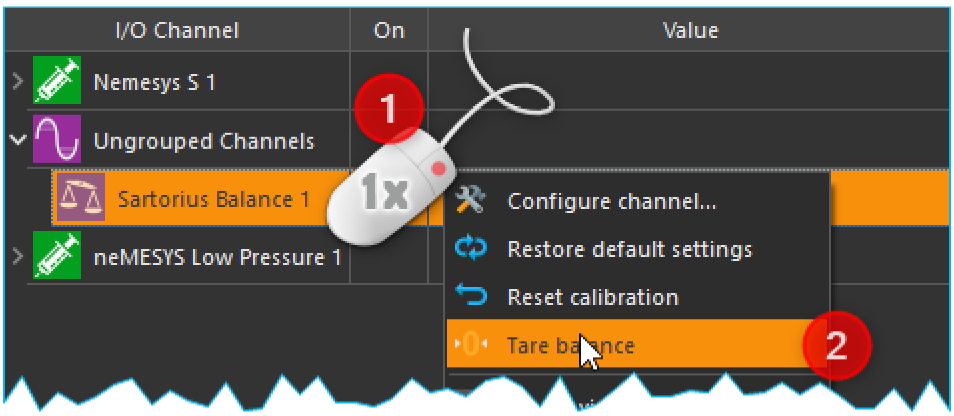
For each balance that has been added to the device configuration, an analogue input channel is displayed in the list of I/O channels. In the picture below, this is the *Sartorius Balance 1* channel.



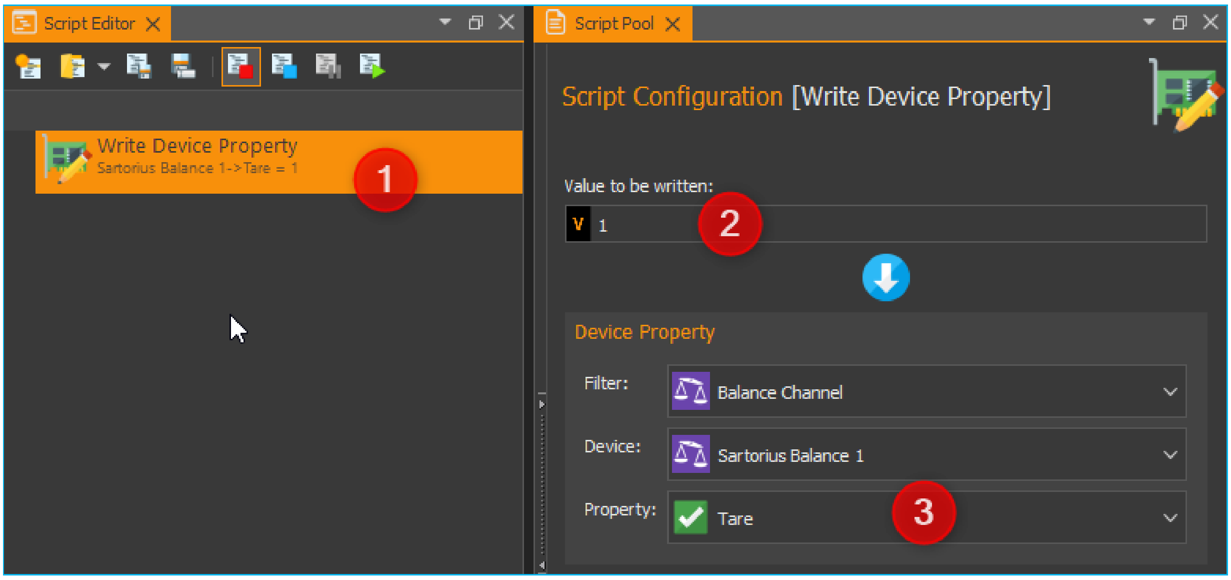
This analogue channel shows the current value measured by the scale in the *Value* column. Like any other analogue channel, this channel can be added in the graphical logger or CSV logger or read in the script.

14.4.2 Tare balance

To tare the balance, right-click on the channel 1 and then select *Tare balance* 2 from the context menu.



The taring of the scale can also be carried out in a script. To do this, add the script function *Device Functions → Write Device Property* 1 to the script. Configure this function so that the value 1 2 is written to the property *Tare* 3 of the scale channel (see figure below).



15 Image Analysis Plugin

15.1 Introduction

This plug-in allows you to view, edit, and save images using the most common image file formats (PNG, JPG). In addition you can use it to:

- zoom in and out of images,
- easily navigate within an image,
- measure the dimensions of details of an image.

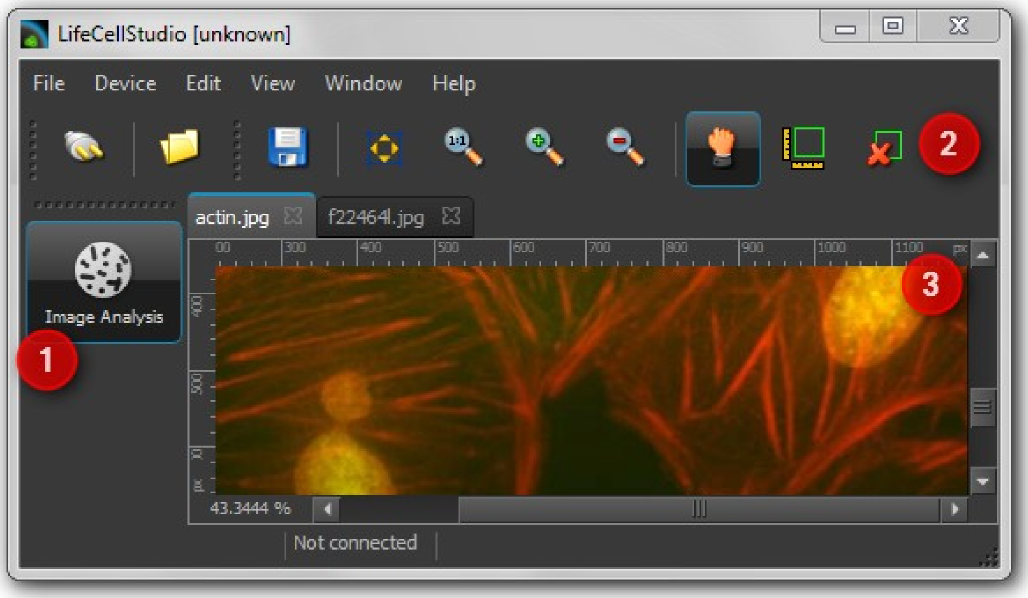


Figure 56: Image Analysis work space.

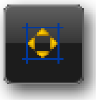
The work space of the *Image Analysis* plug-in can be activated using the button *Image Analysis* in the side bar 1. If no picture has been loaded or taken yet, this work space will be empty.

The user interface of the plug-in consists of two main elements: the toolbar 2 and the image viewer 3.

15.2 Toolbar

Open image file.

Save current image.



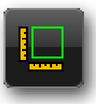
Adjust display size of image to completely fill image viewer area.

Resize image to match original pixel size.

Zoom in to image.

Zoom out of image.

Pan tool to move the currently displayed image section.



Draw rectangular measuring box.

Remove all measurement overlays.

HINT. You can also open image files easily via drag & drop. Simply drag an image file from your file system over the image analysis workbench and drop it there.

15.3 Image Viewer

15.3.1 Overview

The image viewer contains the following elements:

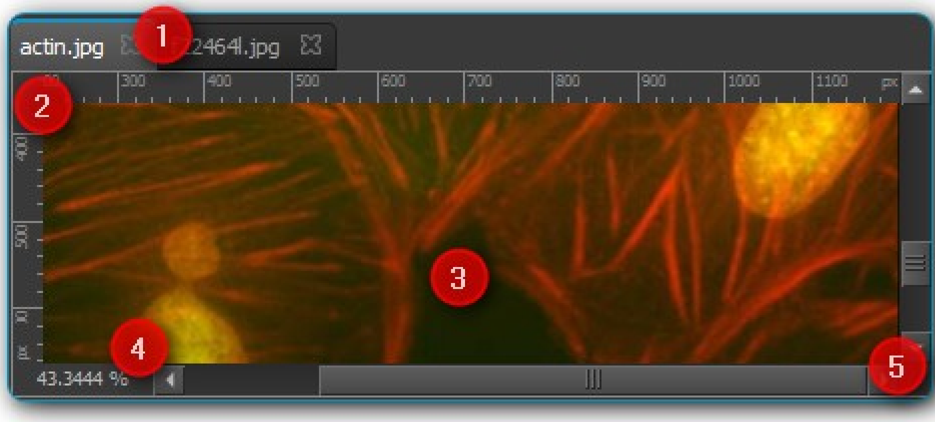


Figure 57: Image viewer.

6 Tabbed list of available images.

7 Ruler with customizable dimensions.

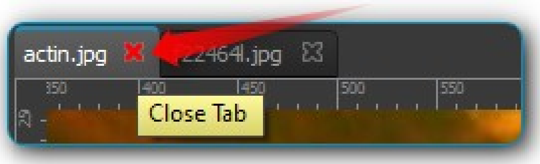
8 Display area.

9 Zoom level status bar.

10 Scroll bars.

A new tab displaying the file name will be added for each opened image, e.g., when you open an image file or take a screen shot using the camera. The active tab is highlighted.

To close a tab, left-click on the cross icon that is displayed on the right hand side of every tab (see figure below).



To easily access the most frequently used functions, use the mouse and right-click within the image viewing area. This will open a context menu with most of the same buttons that are shown in the toolbar.

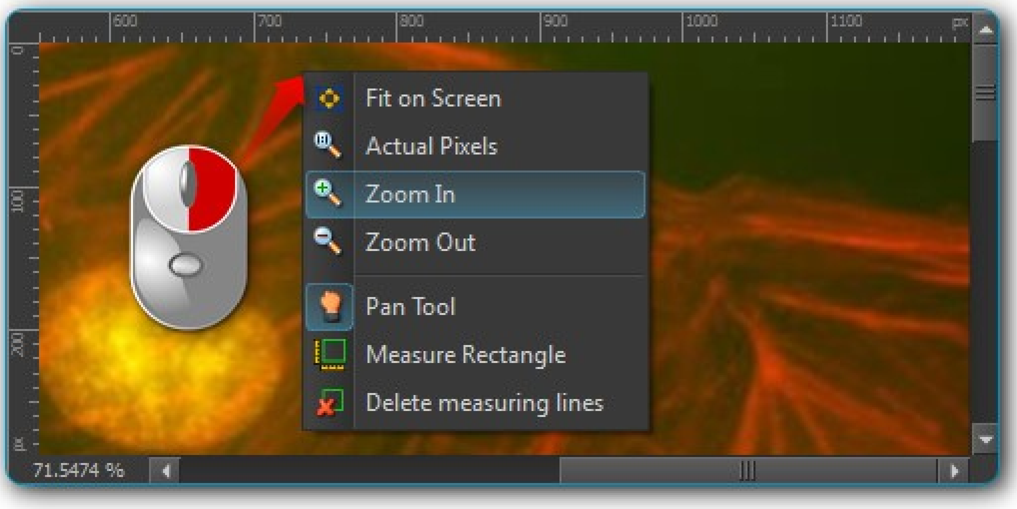


Figure 58: Context menu of the image viewer.

15.3.2 Select Scale Unit

You can select between different scale units for the ruler to measure objects displayed in the image viewer window. To change the scale unit right-click anywhere within the ruler. This will open a context menu with the list of available scale units (see figure below).

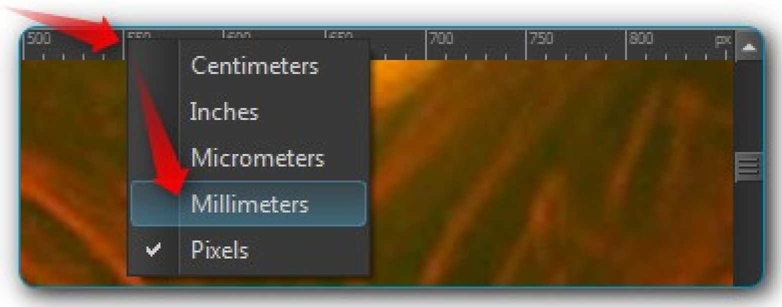
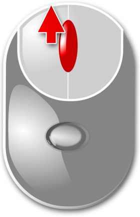


Figure 59: Scale unit selection.

15.3.3 Select Zoom Level

Use the *Zoom In* or *Zoom Out* buttons in the toolbar to adjust the zoom level. Alternatively, simply place the cursor into the image viewer area and use the mouse wheel to re-scale the panoramic image.



Increase zoom level.

Decrease zoom level.

15.4 Image Navigation

15.4.1 Navigation via Scrollbars

If the image is larger than the viewing area (e.g., due to a large zoom level), scrollbars will be displayed below and to the right of the image viewer. Use these scrollbars to adjust the displayed section of the image.

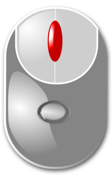
15.4.2 Navigation via the Pan Tool

Activate the Pan Tool in the tool bar to move the displayed image section manually. Click-and-drag into the image viewer to move the image in the viewing area.

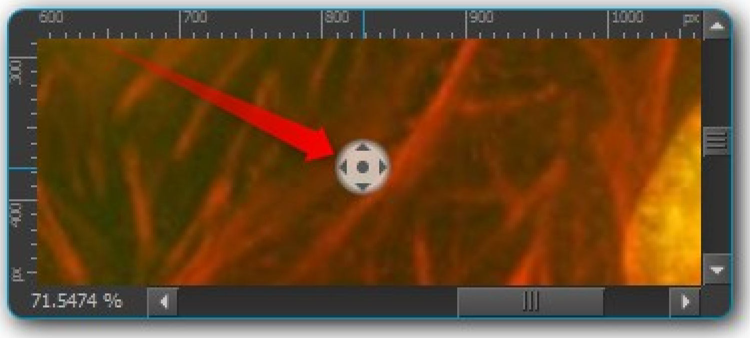
15.4.3 Navigation via Mouse

You may also use the mouse to interact with the panoramic view of an image:

(10) Middle-click into the image.

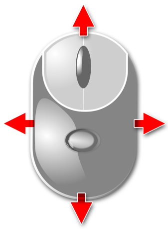


(11) This will activate a navigation cross, which indicates that mouse panning is activated.



(12) By moving the cursor away from the navigation cross, the image will move into the same relative direction. The speed of the movement depends on the distance between the cross and the cursor – the greater the distance, the faster the adjustment.

(13) Mouse panning can be deactivated by another middle-click or a left-click into the image.



15.4.4 Navigation via the Space Navigator

When the *Space Navigator* is installed, loading the relevant plug-in allows for yet another way for image manipulation.

The *Space Navigator* is an intuitive tool to move an image within the viewing area in both X- and Y-axes concomitantly.

To be able to use the *Space Navigator,* it must be installed at a free USB port of your computer. The required drivers have already been installed together with the main Qmix application.

Place the *Space Navigator* so that its cable points away from and the *3D connexion* label towards you.

Figure 60: Positioning the Space Navigator.

To adjust the digital zoom of the currently active image, press down or pull up the navigation knob of the *Space Navigator* (figure below).

Figure 61: Zoom level adjustment with the Space Navigator.

To adjust the section of the image displayed in the image viewing area, pivot the navigation knob of the *Space Navigator* into the desired direction.

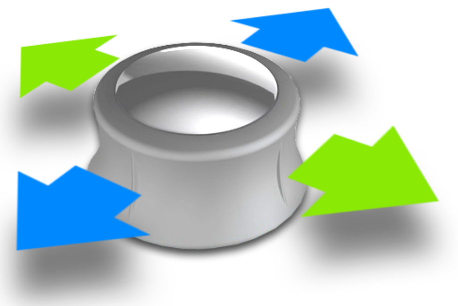


Figure 62: Moving an image using the Space Navigator.

The speed with which the adjustment is carried out can be regulated by adjusting the degree of pivoting – the stronger the navigation knob is pushed or pulled, the faster the image will move.

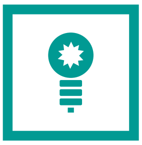
Move image section left & right.

Move image section up & down.

Zoom in/out.

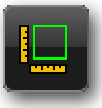
15.5 Measurement of Objects

15.5.1 Draw a Measuring Box



HINT. Before drawing a measuring box, select the desired scale unit.

You can draw and adjust a measuring box using the following steps:

(1)  Click the toolbar button *Measure Rectangle* to activate the measuring box tool.

(2) Left-click into the image viewing area to define the upper left-hand corner of the measuring box 1.

(3) While holding the left mouse button, drag the cursor to size the box as required 2.

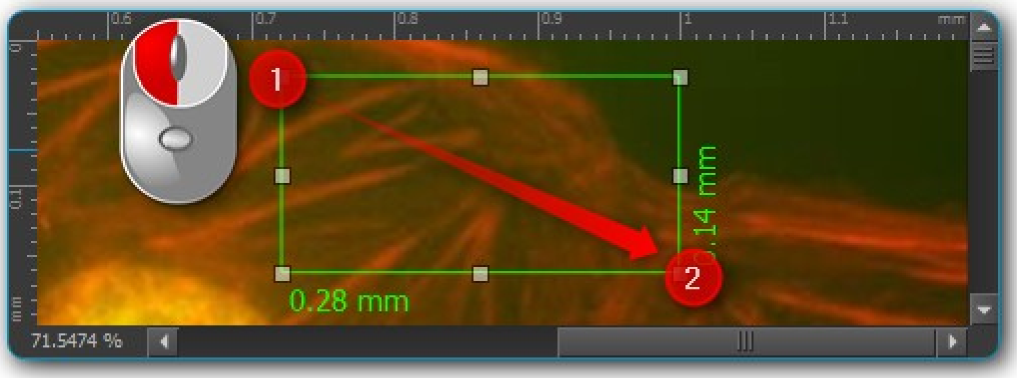


Figure 63: Drawing a measuring box.

15.5.2 Moving the Measuring Box

To move an entire box to a new location within the viewing area, first place the cursor within the box so that it changes into a crossed arrow 1. Left-klick to drag the measuring box to its new position 2.

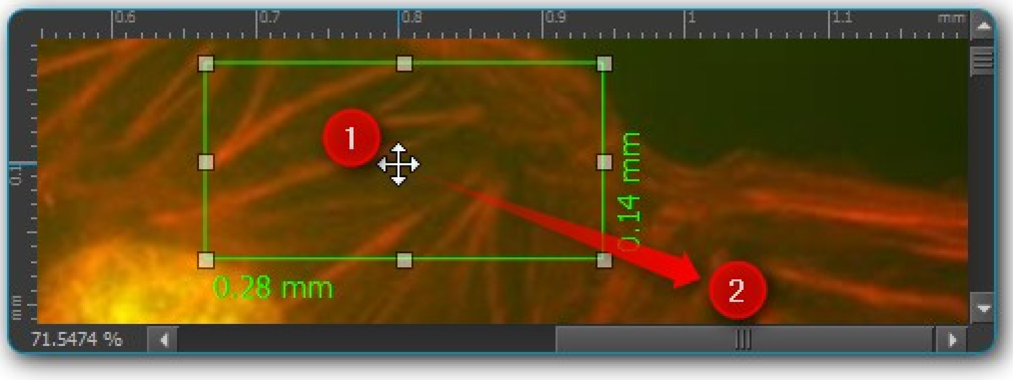


Figure 64: Moving a measuring box.

15.5.3 Adjusting the Size of a Measuring Box

To change the size of a measuring box, move the cursor to one of the check marks at the corners or the sides of the box (figure below). The cursor will change to a double-sided arrow 1, which indicates that you can now resizing the box by dragging the respective mark to a new position.

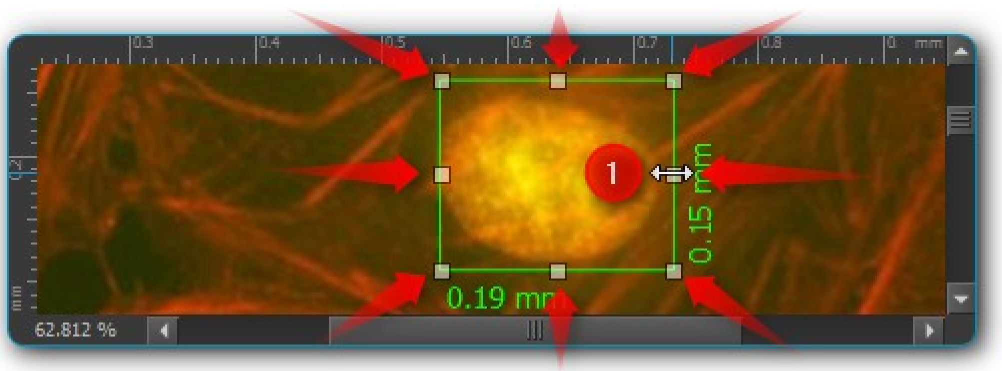


Figure 65: Adjusting the size of a measuring box.

15.5.4 Deleting a Measuring Box

You can delete individual measuring boxes. First click into the respective box to activate it – the activated box can be identified by the presence of the check marks. Press the Delete button of your keyboard to remove the box.

Delete all measuring boxes within an image by clicking the button *Delete measuring lines* in the toolbar or via the context menu.

16 Camera Plugin

16.1 Introduction

The camera plugin allows the user to integrate a broad range of cameras into the software. The plug-in supports, *e.g.*, Firewire 1394 cameras (DCAM standard), USB cameras, as well as all standard Windows DirectShow cameras.

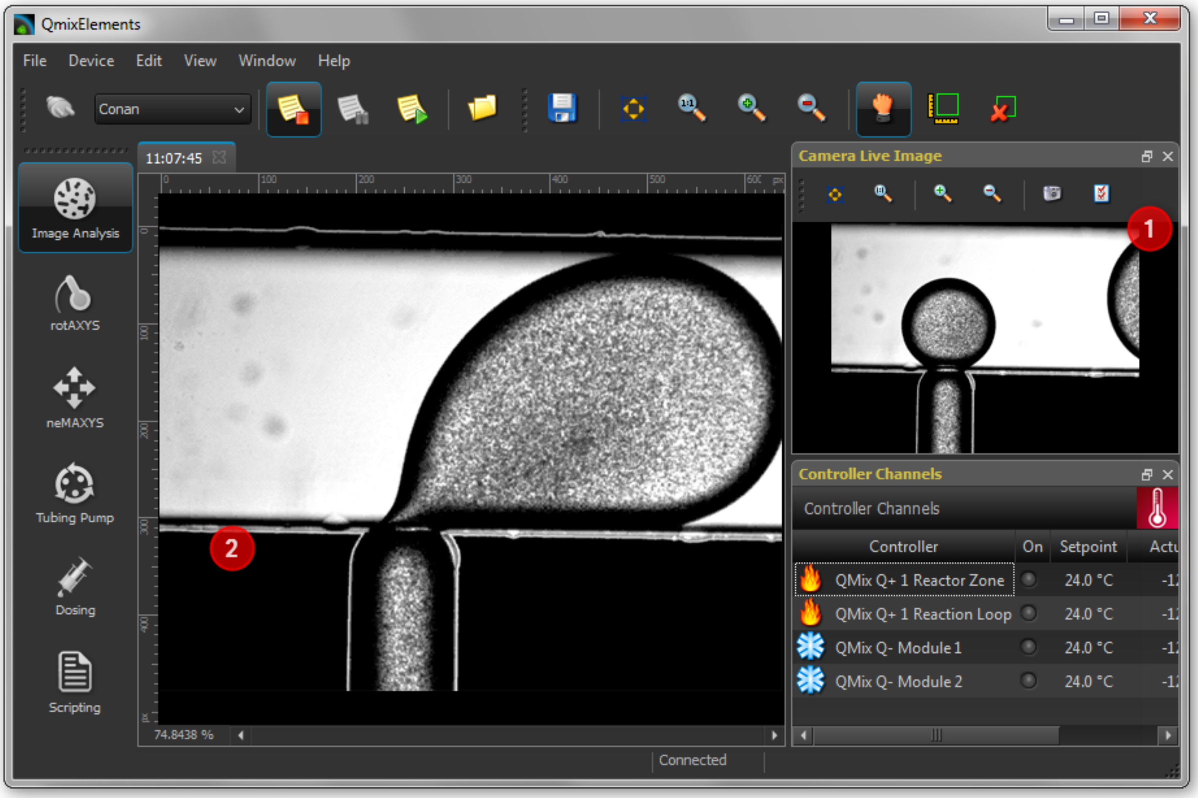


Figure 66: Camera live image and image viewer.

The figure above shows the typical work environment of the plug-in, including the *Camera Live Image* in a detachable tool window 1 as well as the *Image Viewer* 2 displaying captured images.

16.2 Camera Live Image

16.2.1 Overview

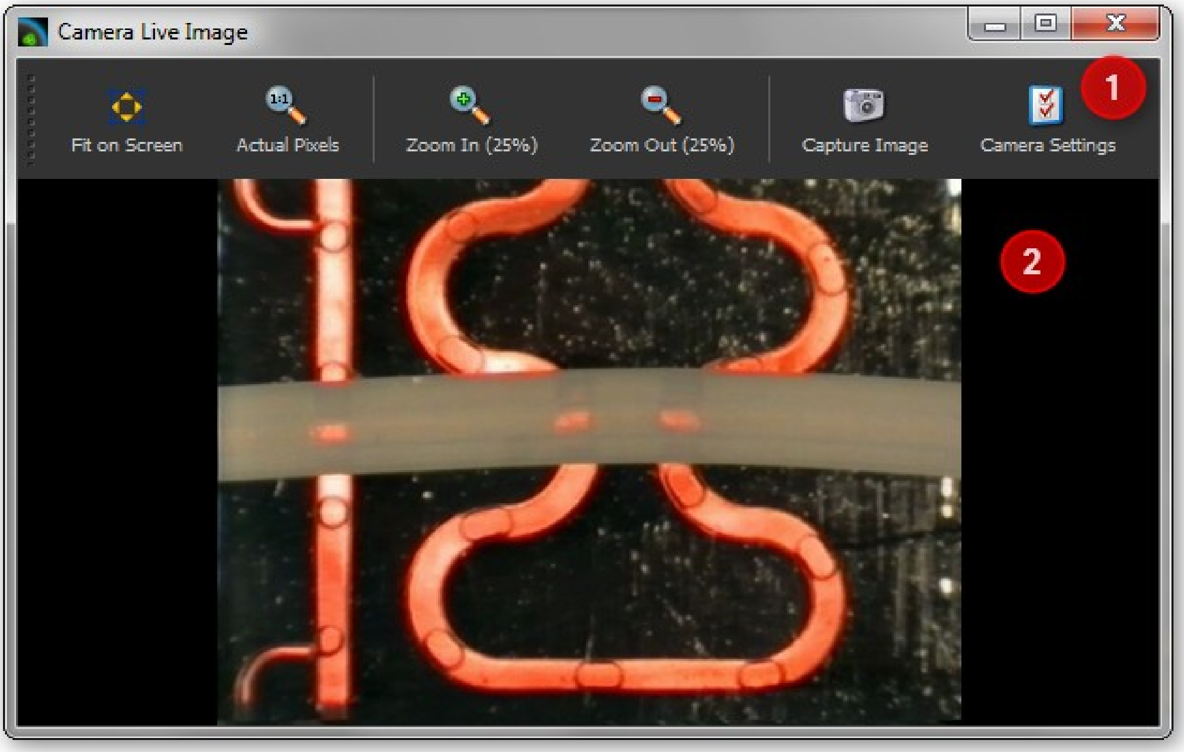


Figure 67: Camera Live Image window.

The camera's live image is displayed in a tool window that the user may move freely on the screen desktop via drag-and-drop of the window's title frame. Thus the user can place the live image on a second monitor and may observe the live image whilst maintaining complete oversight and control over the software settings on the first monitor.

1 Toolbar

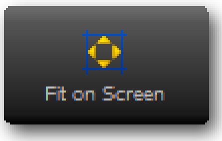
2 Camera live image

16.2.2 Toolbar

A variety of tools are available via buttons in the top part of the live image window so that you can adjust the displayed image or camera settings (see figure below).

Figure 68: Toolbar of the Camera Live Image window.

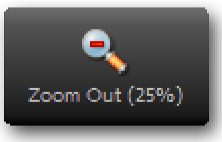
These buttons have the flowing functions:



Resize image to fit window to fit the image into the available display space.

Resize image to its native pixel resolution relative to the screen.

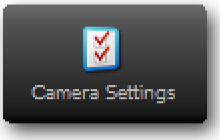
Increase digital zoom



Decrease digital zoom

Capture the currently displayed image.

Record video

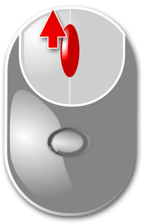


Opens dialog for configuration of camera settings

16.2.3 Adjust Zoom Level

The digital zoom may be adjusted using one of three alternatives:

- Toolbar – using the toolbar buttons will change the zoom between preset levels.
- Context menu – a right-click into the live image opens a context menu that contains the relevant items to adjust the zoom level.
- Mouse wheel – changes the zoom level smoothly



Increase magnification level (zoom in).

Decrease magnification level (zoom out).

16.2.4 Manual Capture

Click on the toolbar button *Capture Image to* capture the currently displayed camera live image: The captured image will be saved to the software's image viewer for further adjustments.

16.2.5 Manual Video Recording

If you want to record the current camera live image as a video, simply press the *Record Video* button. The video recording will remain active until you stop it by clicking the *Record Video* button again.

Video recording inactive

Video recording active

In the *Event Log* of the application you will get status information about the recording state and about the file name of the saved video file.

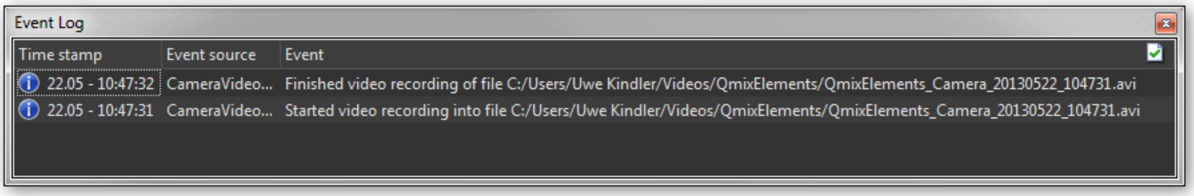


Figure 69: Status information video recording

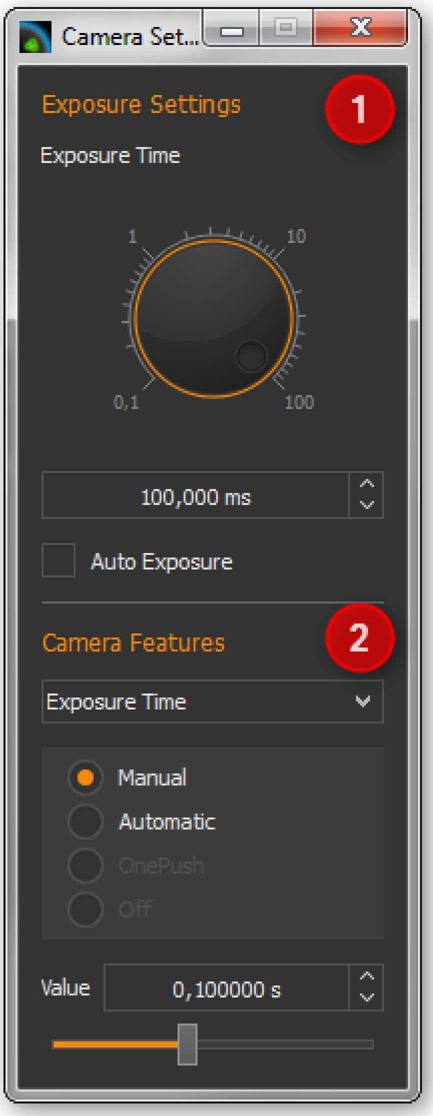
The file name of each video file is extended with a time stamp that indicates the start of the recording. E.g. the recording of the video file *QmixElements_Camera_20130522_104731.avi* started 2013-05-22 at 10:47 o'clock.

IMPORTANT. The XVID codec is used for video recording. If you want to play back the recorded videos with your media player, it needs to support this codec or the XVID codec should be installed on you computer system.

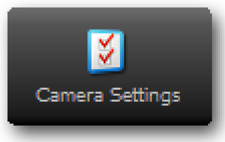
16.3 Camera Settings

16.3.1 Introduction

The quality of the images depends on the lighting conditions and the chosen camera settings. The following will provide an overview of the most important parameters that may be changed via the software.



The window pane to adjust the camera settings may be opened via the *Camera Settings* toolbar button:



The window has to main areas:

1 *Exposure Settings* – to set the exposure settings.

2 *Camera Features* – to adjust general camera settings.

IMPORTANT. Depending on the used camera type (Direct Show, USB, or Firewire) or model, the actual camera settings dialog may look differently.

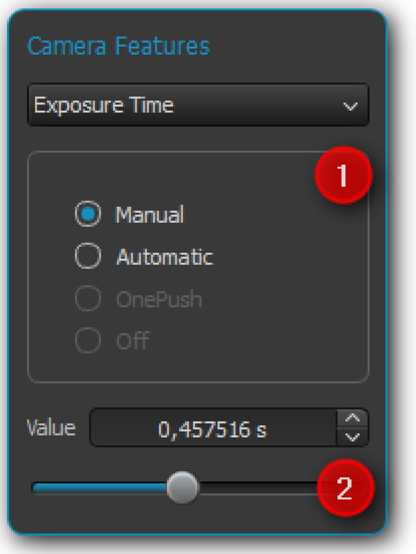
IMPORTANT. Some cameras may not be supported by the Camera Settings dialog and will show their own configuration panel instead.

16.3.2 Setting Exposure Parameters

With *Exposure Settings* you may set the exposure time of the image sensor inside of the camera. It may operate either in manual mode or in automatic mode. The *Auto Exposure* mode will adjust the exposure time according to the brightness of the image.

16.3.3 Setting Camera Parameters

All settings of the camera may be adjusted via the *Camera Features* controls. The feature selection and their respective value ranges may vary according to the type of camera used.



If supported by your camera, you may toggle between manual and automatic modes 1. In manual mode the values may be changed either numerically via the input box or adjusted via the sliding controller 2.

IMPORTANT. Depending on the type of camera used, the available parameters and their value ranges may differ.

16.4 Camera Script Functions

16.4.1 Introduction

The camera plug-in contains script functions for capturing still images and for recording videos:

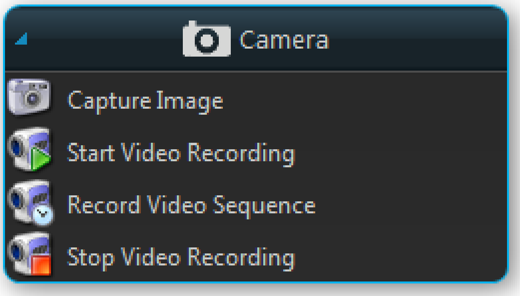


Figure 70: Camera script functions.

16.4.2 Still Image Capture

The function *Capture Image* is to take individual shots of the current camera view. The *Image File* section is to set the file path and file name of the captures image files 1.

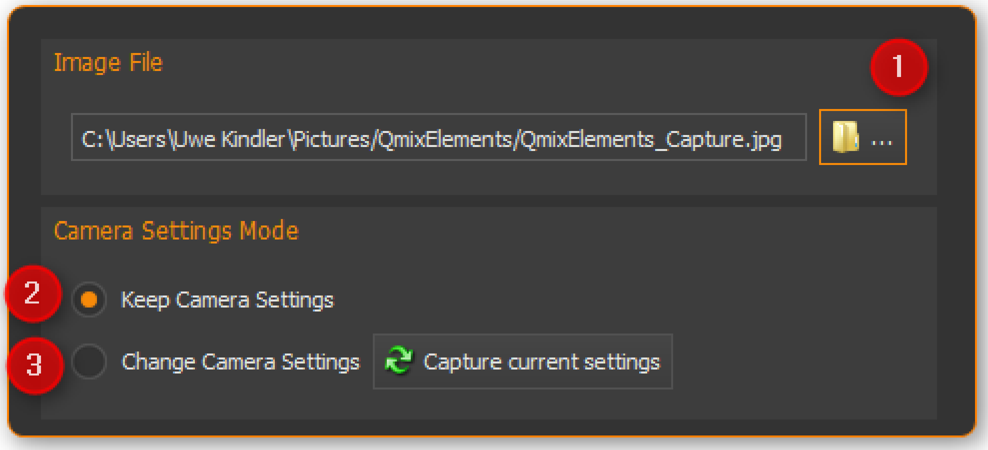


Figure 71: Capture Image configuration.

Subsequently you may set whether the camera settings should be used unaltered 2 (*Keep Camera Settings*) or whether to reset them to pre-defined and saved values 3 (*Change Camera Settings*) before capturing an image.

The current camera settings may be saved as function parameters by pressing the button *Capture current settings* to be able to reset the camera to those parameters just prior to capturing another image at a later point. Thus you can take images of different objects with completely different camera settings.

HINT. You may select from different image formats by setting the appropriate file extension (.png or .jpg).

Every image file name will be appended with a time stamp at the time of capture. This is, when you have defined the file name as *Photo.jpg* the resulting file will be saved as *Photo_20120921_154502.jpg* according to the syntax *name_YYYYMMDD_hhmmss.jpg*.

HINT. You may capture images at set intervals to subsequently compile them to a stop-motion movie using the Video Builder plug-in.

16.4.3 Start Video Recording

With this function you can start recording of the camera live image into a video file. The record will remain active until it is stopped with the *Stop Video Recording* function. The video recording will stop automatically when the script execution ends.

In the configuration area you can configure all the video recording parameters. Enter a unique name for this recording session in the input field *Recording Session Name* 1. This name is required to stop the recording later with the *Stop Video Recording* function.

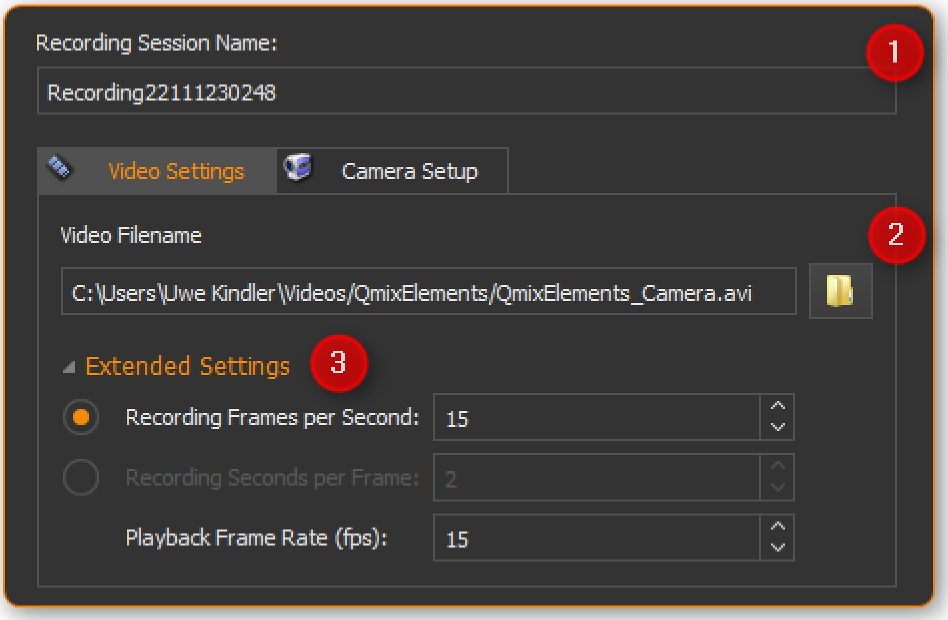


Figure 72: Configuration of Start Video Recording function

16.4.3.1 CONFIGURING VIDEO SETTINGS

In the *Video Settings* tab, you configure all video settings. Click on the button with the folder icon 2 to select a file name and a storage location for the video file in the *Video Filename* field.

IMPORTANT. Every video file name will be appended with a time stamp that identifies the start of its recording. This is, when you have defined the file name as Video.avi the resulting file will be saved as Video_20130522_154502.avi.

By clicking *Extended Settings* 3 you can display additional settings. In the field *Recording Frames per Second* set the number of images to be recorded per second. To get a real-time video playback later, you should use the current camera frame rate.

HINT. If you create a new recording function, the current camera frame rate is automatically entered into the **Recording Frames per Second** field.

If you want to monitor slower processes, you can also choose a smaller frame rate for recording. If you check the check box *Recording Seconds per frame*, you can choose very slow frame rates (several seconds or minutes per image).

This mode is suited for creating time-lapse recordings of very slow processes. So you can for example, take a picture every 30 minutes for several days and then play back these captured images in a video with a frame rate of 24 frames per second.

In the input field *Playback Frame Rate (fps)*, set the frame rate at which the film is played back in media player later. In the fields of video production or motion picture a frame rate of 24 images per second is the standard frame rate. If you want to play back the movie in real time, that means with its real recording speed, you should choose a frame rate here, that matches the recording frame rate.

16.4.3.2 CONFIGURING CAMERA SETTINGS

Select the *Camera Setup* 1 tab to display the controls for configuring the camera settings:

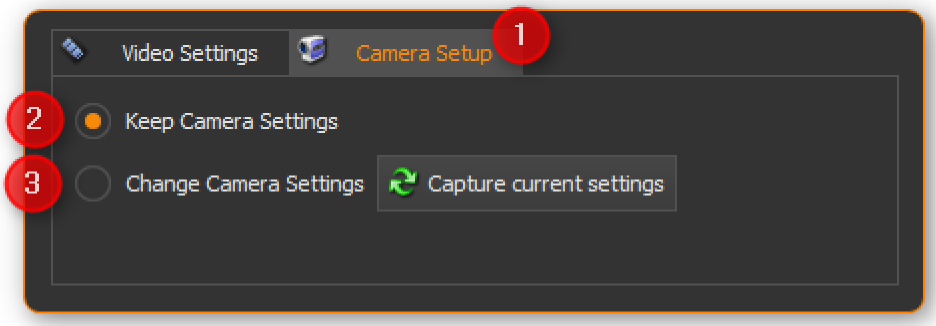


Figure 73: Configuring camera settings of Start Video Recording function

Here you may set whether the camera settings should be used unaltered 2 (*Keep Camera Settings*) or whether to reset them to pre-defined and saved values 3 (*Change Camera Settings*) before capturing an image.

The current camera settings may be saved as function parameters by pressing the button *Capture current settings* to be able to reset the camera to those parameters just prior to capturing another image at a later point. Thus you can take images of different objects with completely different camera settings.

16.4.4 Record Video Sequence

Use this function to record a video sequence with a fixed duration. You can configure the desired recording duration and the recording stops automatically after the expiry of the recording time.

All video settings and camera settings for this script function are identical to the settings of the *Start Video Recording* function. Additionally you may set the recording duration for this function.

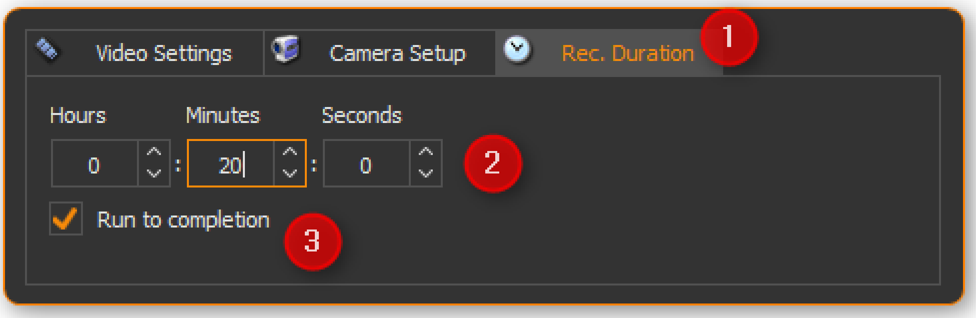


Figure 74: Configuration of Record Video Sequence function

Select the tab *Rec. Duration* 1. Now you can set the recording duration accurate to the second 2. Use the selection field *Run to completion* 3 to determine, when the recording function is finished and when the next script function will be executed:

- *Check box active* – the next function is executed when the recording is completed
- *Check box inactive* – the recording is started and then script execution immediately continues with the next script function

HINT. The recording of a video sequence can be stopped at any time by calling the Stop Video Recording function.

16.4.5 Stop Video Recording

A running video recording that was started with the functions *Start Video Recording* or *Record Video Sequence* can be terminated at any time with this function. Simply enter the name of the recording session you want to quit into the input field *Recording Session*.

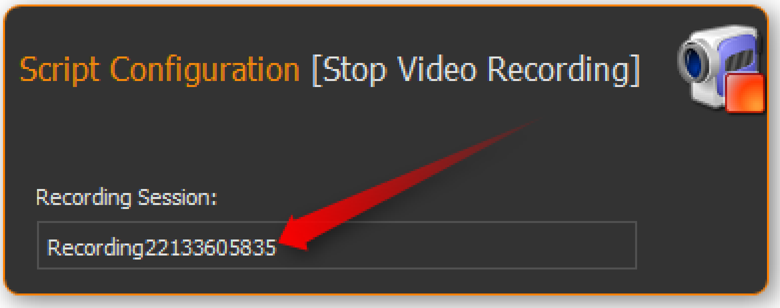


Figure 75: Configuration of Stop Video Recording function

17 Video Builder Plugin

17.1 Introduction

The *Video Builder* plug-in allows you to quickly compose videos from individual images taken by timing or scripting via the camera or the microscope plug-ins. This is useful, for instance, for time laps recordings of slow microfluidic processes or for long-exposure microscope pictures.

To open the *Video Builder* plug-in, select the relevant item from the main menu via *Edit* → *Video Builder*.

The *Video Builder* dialog contains the following elements (see figure below):

3 Tool bar,

4 List of image folder(s),

5 Video parameter settings,

6 Thumbnail of the currently processed picture,

7 Progress bar of the processing step.

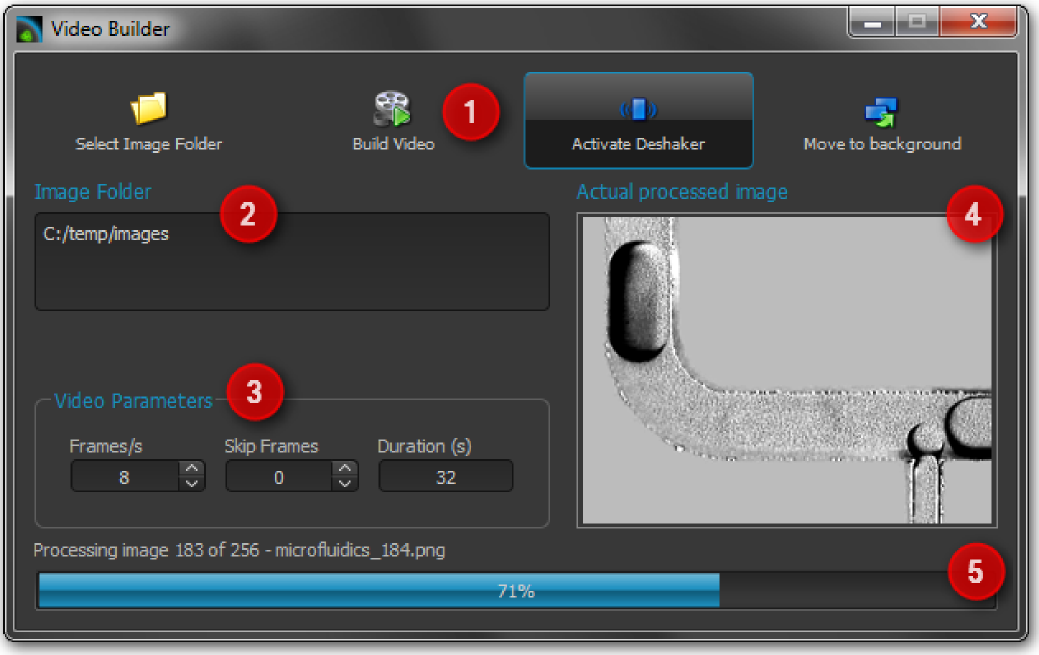
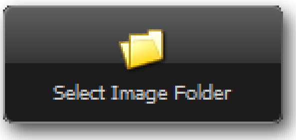


Figure 76: Video Builder dialog box.

17.2 Video Assembly

The following description shows you how to assemble a video from individual images.

17.2.1 Step 1 – Select Image Files



To select the image files that you want to process, first locate the relevant folder and open it via *Select Image Folder* in the tool bar.

This opens a dialog box (see figure below) that allows you to navigate to the folder containing your image files 1. Open the folder by clicking on *Choose* 2.

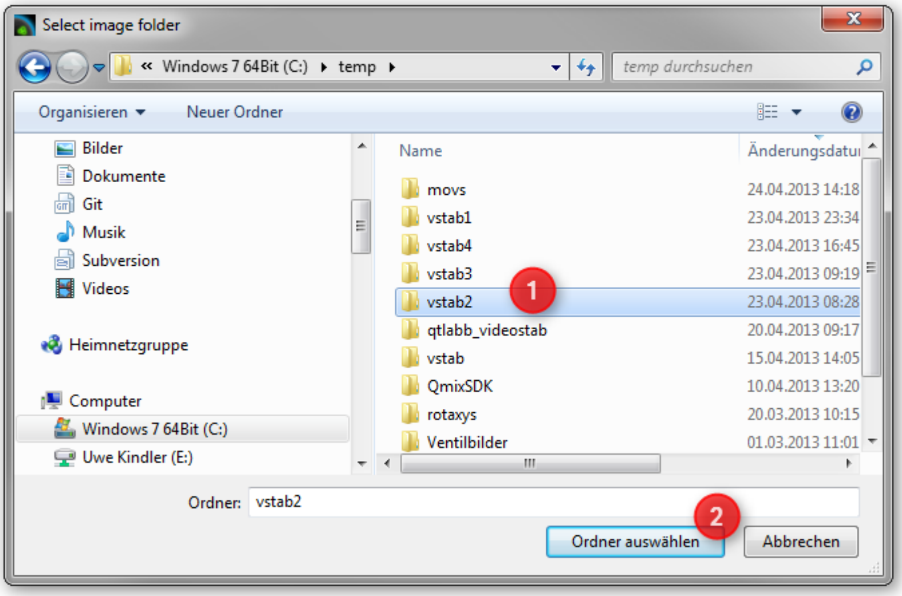
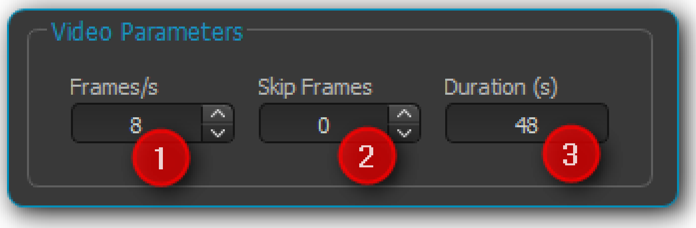


Figure 77: Select directory containing individual image files.

17.2.2 Step 2 – Video Assembly Parameters

The box *Video Parameters* is to configure the parameter settings for the image assembly process.



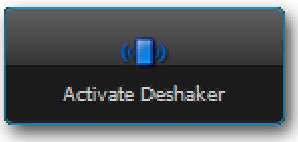
You can set the following parameters:

1 *Frames/s* – configures the frame rate, i.e., how many images are to be shown per second.

2 *Skip Frames* – set the number of images that are excluded from the video. If this value is set to zero, all image files will be used; if set to one, for instance, every other image will be skipped and thus only every second image will be used to assemble the video.

Based on the number of available images and the two parameters just set, the third box (*Duration (s)* 3) will show the length of the final video.

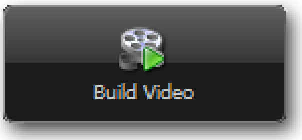
In addition, the software offers the possibility to de-shake (stabilize) the video sequence:



HINT. Video sequences (e.g. of images captured from microscope camera) can be stabilized using the de-shaker function.

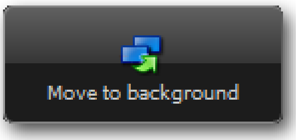
IMPORTANT. The de-shaking step will increase the time for building a video.

17.2.3 Step 2 – Start Video Assembly



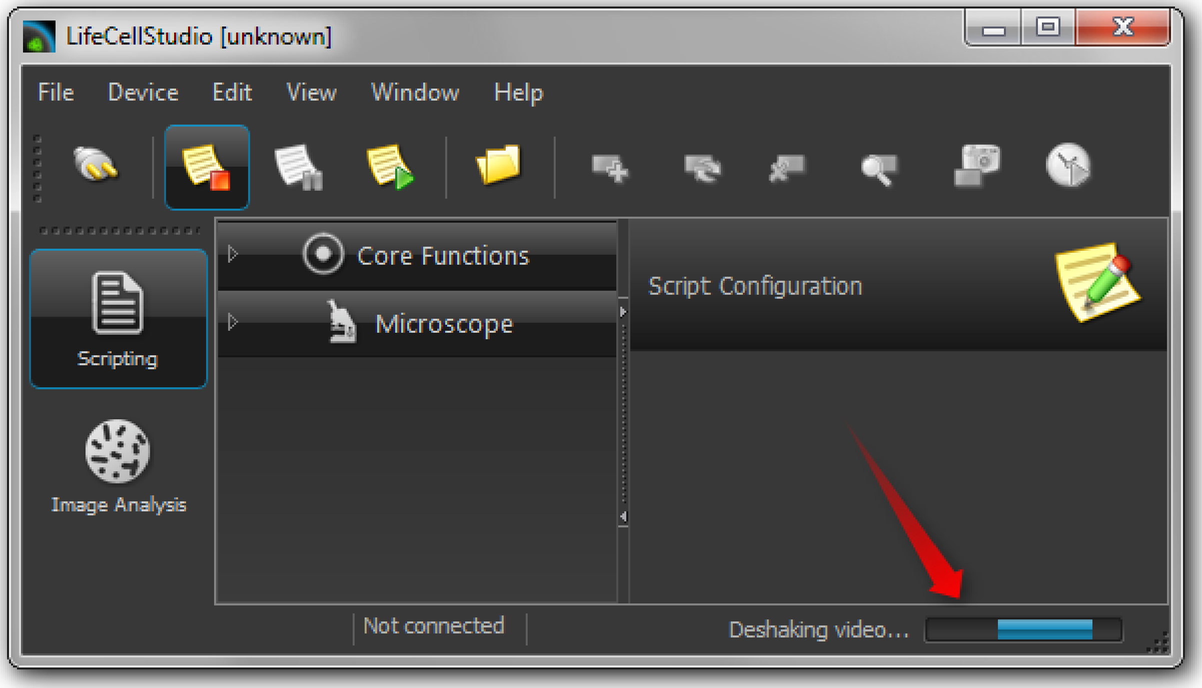
Start the assembly process by clicking on the *Build Video* button.

This will open a dialog box that is to define the target directory and the name of the video file. Processing will start immediately thereafter.



The assembly process may take several minutes, in particular when a large number of images is to be processed. Processing may be continued as a background task by activating the *Move to background* feature of the Qmix Elements plug-in. This will close the *Video Builder* dialog and allows you to continue to work with the Qmix Elements application.

If a process is being carried in the background, this will be indicated in the status bar of the main application window (see figure below).



You will be notified via the application's *Event Log* as soon as the video assembly process has been completed (see figure below). Now you may start the process with a new set of pictures or with different settings.

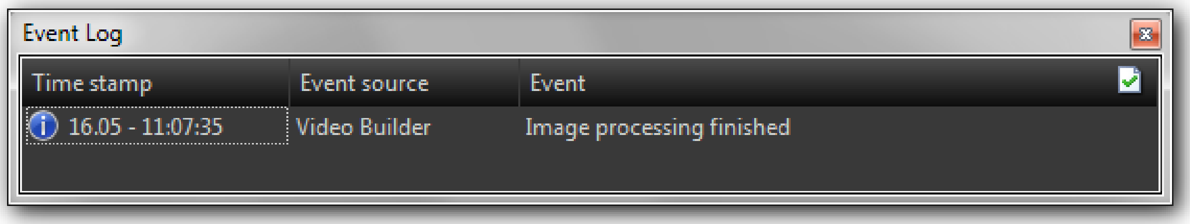


Figure 78: Event Log message about process completion.

18 CSV Data Logger Plugin

18.1 Introduction

The data logger plug-in provides the user with a powerful tool to record any of the data provided by connected devices in a user-defined time interval. Data are written into a file of the CSV format (CSV: comma/character-separated values). This text file format is commonly used to save and exchange simply-structured data.

HINT. CSV files can be opened and worked with in spreadsheet applications, such as Microsoft Excel if the correct value-separating and decimal sign has been used.

18.2 Configuration Dialogue

18.2.1 Open the Configuration Dialogue

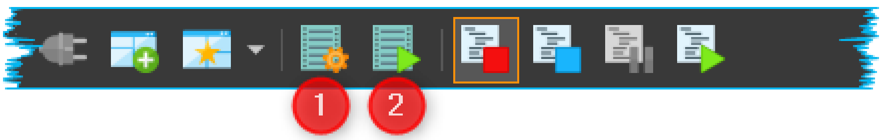
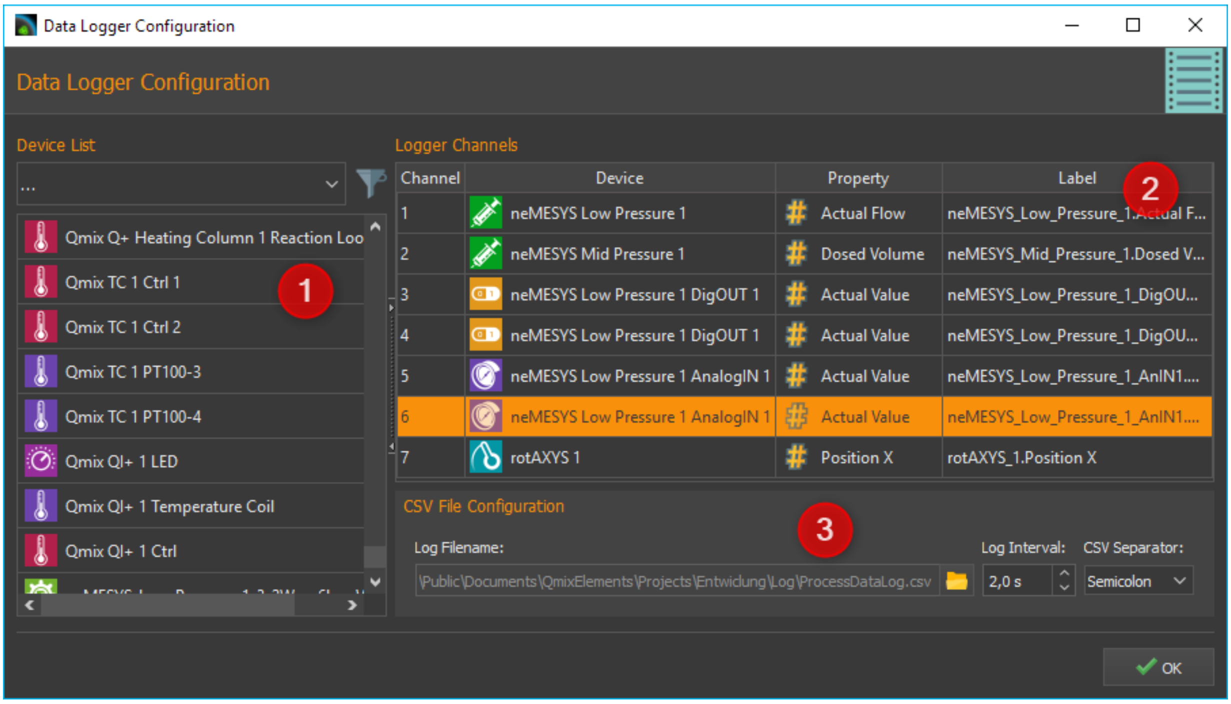


Figure 79: Toolbar for data logging.

When the data logging plug-in has been loaded, the toolbar will display two additional buttons for the configuration of the logging of data 1 and to start/stop the logging process 2.

18.2.2 Overview Data Logger Plug-in

Once the data logging configuration has been activated, the following configuration dialogue will be displayed:



The configuration dialogue contains the following elements:

- **1** *Device List* – displays all devices or modules that provided recordable data. The filter selector above is to limit the list to specific device types, e.g. valves.
- **2** *Logger Channels* – lists all channels that may be recorded by the logger.
- **3** *CSV File Configuration* – allows the user to set various settings for the data logging file.

18.2.3 Overview Table Logger-Channels

Channel	Device	Property	Label
1	neMESYS Mid Pressure V3 1	DosedVolume	neMESYS_Mid_Pressure_V3_1.DosedVo...
2	neMESYS Mid Pressure V3 1 AnalogIN 1	ActualValue	neMESYS_Mid_Pressure_V3_1_AnIN1.Ac...
3	Qmix V 1	ActualPosition	QmixV_1.ActualPosition
4	neMESYS Mid Pressure V3 2 DigIN 1	On	neMESYS_Mid_Pressure_V3_2_DigIN1.On

The table *Logger Channels* shows the configuration of the process data logger. Each row in that table corresponds to one column in the recorded csv file. The following columns may be recorded:

- *Channel* – shows the channel number of the corresponding channel.
- *Device* – contains the device name for which the data will be recorded and its device icon.
- *Property* – this is the name of the device property/process data value that will be recorded. Its type (numeric or boolean) can be identified by the displayed icon.

 Numeric value

 Boolean value

Text value

- *Label* – allows you to define a customized description for the selected channel. This description will be used as the column header in the csv file.

In order to add a channel to the data logging process, simply follow the steps below.

18.3 Configuration of the Logging Process

18.3.1 Step 1- Adding of Channels

Drag-and-Drop the device for which you want to log the data from the *Device List* into the *Logger Channels* list. The new channel will be inserted into the list at the desired position (see figure below).

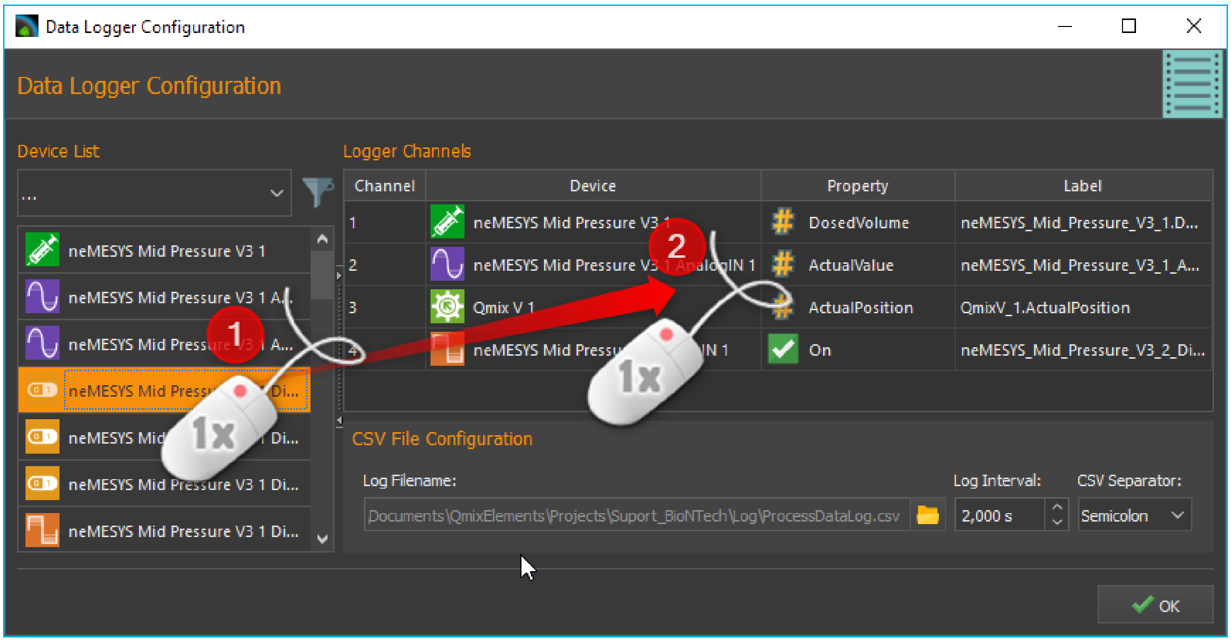


Figure 81: Adding channels using drag-and-drop.

HINT. To simplify the device selection, the device list can be filtered according to device type.

18.3.2 Step 2- Select Device Property

In the *Logger Channels* list you now need to select the *Property* of the device that you want to record. For this, double-click into the respective filed within the column *Property* and select the device property from the opening list (see figure below).

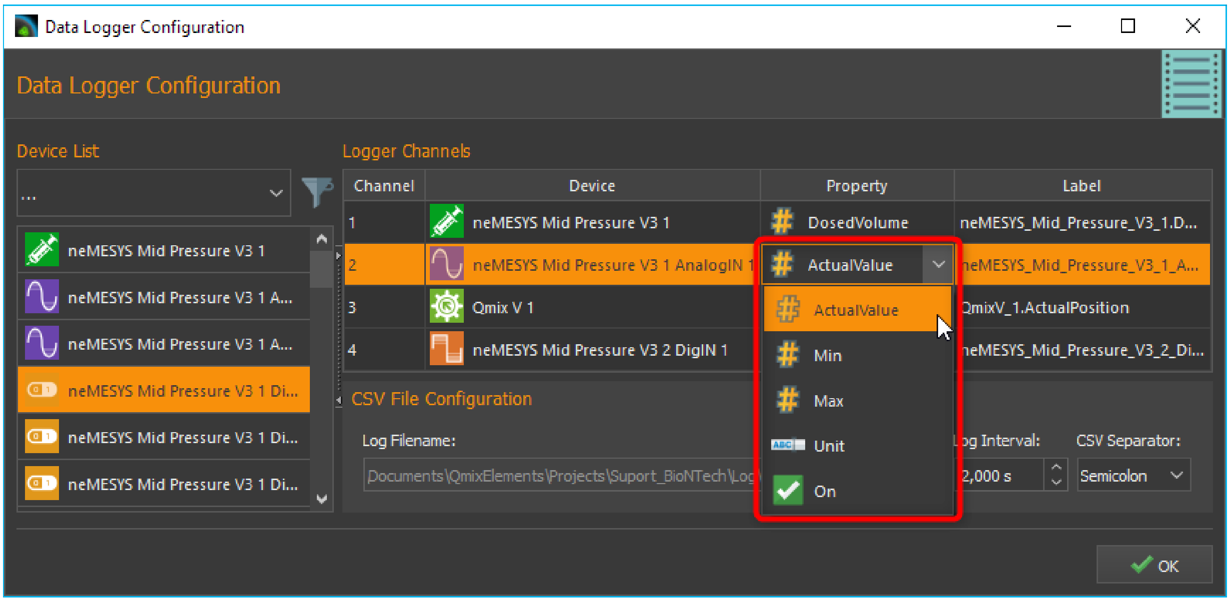


Figure 82: Selecting the device property that is to be recorded.

18.3.3 Step 3 – Channel Description

In the column *Label* you can customize the description for each channel. This label will be used as the column header of the csv file for the selected channel.

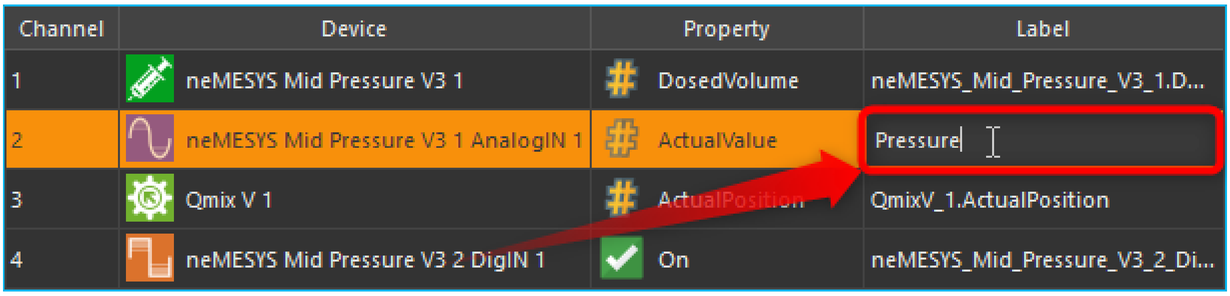


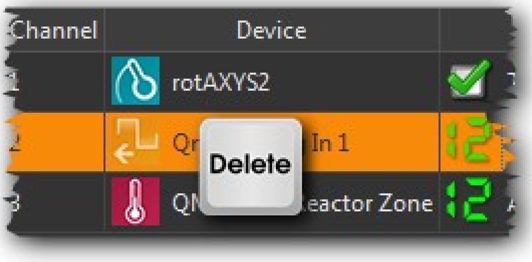
Figure 83: Customizing the channel label.

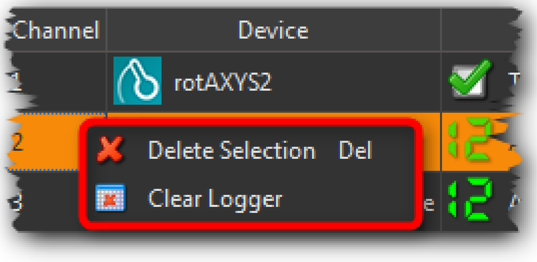
To do this, double-click into the respective table cell that is to be changed and insert the new description (see figure above).

IMPORTANT. Upon choosing a new device property, a new channel description will be assigned automatically. That is, you should change the channel label only once the correct device property has been selected.

18.3.4 Deleting Channels

Highlight the desired channels using the mouse to delete one or more channels from the list, and then use either the *Delete* key or the *Delecte Selection* item of the right-click context menu:





To delete the entire channel list, use the context menu item *Clear Logger*.

18.3.5 Step 4 – Configuration of CSV Properties

In the *CSV File Configuration* section you can set the global properties of the CSV logger as well as the format of the recorded data (see figure below).

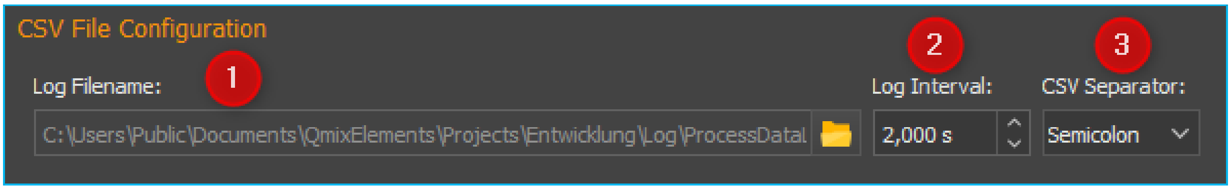


Figure 84: Configuration of global csv properties

18.3.5.1 SELECT FILE NAME AND FOLDER

Set the file name and location of the log file via *Log Filename* 1. For this, click on the folder symbol on the right, select the target folder and give a file name.

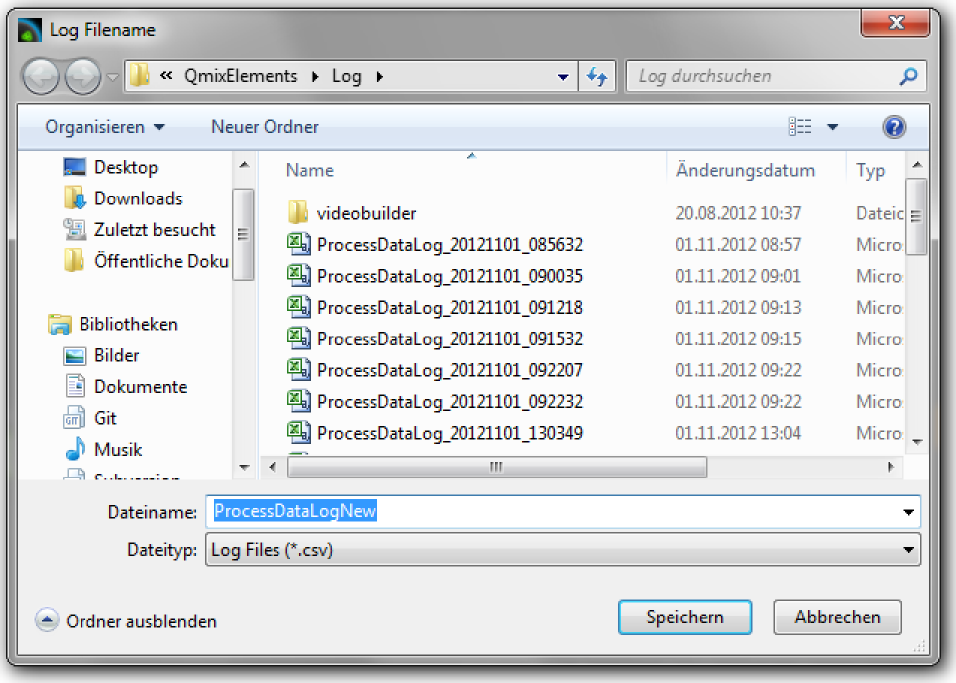


Figure 85: Setting file name and folder for the log file.

18.3.5.2 SETTING THE RECORDING INTERVAL

Set the time interval at which the data of all channels is to be recorded via the field *Log Interval* 2. The time unit for the interval is seconds and you may set it in 0.1 second increments.

IMPORTANT. Choose the recording interval as large as possible and as small as necessary. This will minimize amount of data that will be recorded.

18.3.5.3 SET THE SEPARATING CHARACTER

The character that will be used to separate individual data (columns) needs to be set via the selection field *CSV Separator* 3. Depending on the software that is to be used for data evaluation, the character that needs to be used may change.

HINT. To obtain a CSV file that can be imported into Microsoft Excel, the semicolon (;) should be used.

IMPORTANT. All configuration settings of the process data logger will be saved upon closing the configuration dialogue and are available when the application will be restarted.

18.4 Start/Stop of the Logging Process

Use the relevant toolbar button to start and stop the data logging process.

A new data file will be created each time the logging process is started. A time stamp with date and time will be added as a suffix to file name (*_YYYYMMDD_hhmmss)*. That means, if the file name *ProcessDataLog.csv* has been assigned by the user, starting the logging process on November 05, 2012 at 10:32 am and 9 seconds will create a logging file with the name *ProcessDataLog_20121105_103209.csv*. This ensures, that a new logging file with a unique time stamp will be created each time the logging process is started.

18.5 Log File Data Format

The recorded CSV files have the following structure:

- Each CSV file consists of multiple data sets organized in rows and separated by line breaks.
- Each data set consists of a number of data fields (columns) that are separated by a specific character (e.g., semicolon).
- The first column always contains the relative time point (in seconds) of the corresponding data set.
- The first row shows the channel labels as configured by the user.

To obtain the absolute time stamp for a data set, you may simply add an extra column to the spreadsheet and calculate the time by adding the data set's relative time to the absolute time given in the file names time stamp.

	A	B	C	D
1	Time (s)	rotAXYS2->Actual Position X	QmixIO1 Dig In 1->Actual Value	QMixQPlus1ReactorZone->Actual Value
2	0	144,5	0	-130,95
3	0,1	144,5	0	-130,95
4	0,2	144,5	0	-130,95
5	0,3	144,5	0	-125,882
6	0,4	144,5	0	-125,882
7	0,5	144,5	0	-125,882
8	0,6	144,5	0	-125,882
9	0,7	144,5	0	-125,135

Figure 86: CSV log file opened in Microsoft Excel.

HINT. The absolute time stamp at which logging started is given in the file name of the log file.

18.6 Script Functions

18.6.1 Introduction

To automate the capture of data or to synchronize data capture with other processes, the CSV data logger can be started and stopped using QmixElements script functions. The corresponding functions can be found in the *Logging* category in the list of the available script functions.

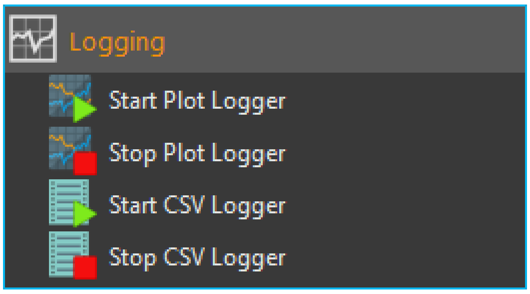


Figure 87: Logger script functions

18.6.2 Start CSV Logger

This function is used to start the CSV logger with the currently configured settings and channels. A new log file is created with the current time stamp.

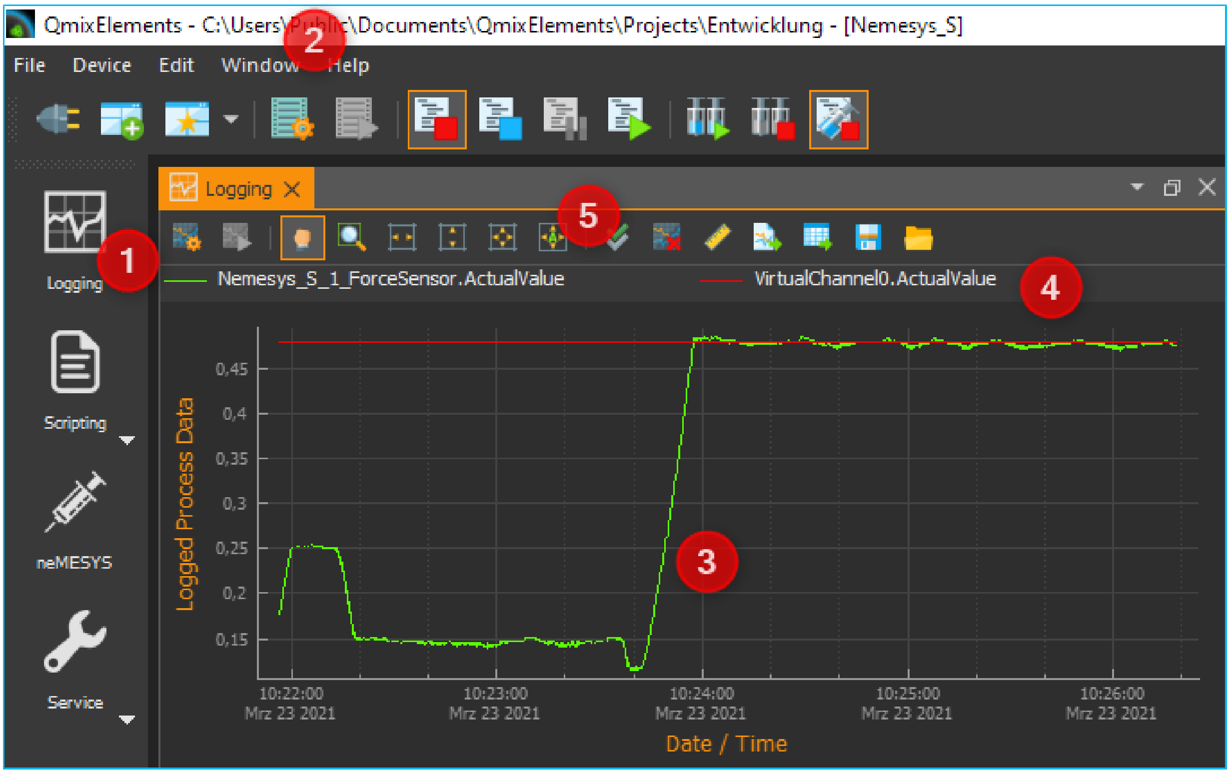
18.6.3 Stop CSV Logger

This function stops the current logging and closes the open log file.

19 Process Data Graph

19.1 Introduction

In addition to the ability to record data in CSV files, the data logger plug-in allows you to visualize process data. This allows for real-time visualization of, e.g., process data changes. Use the push button *Logging* 1 (see figure below) in the sidebar to display the process data graphs or show the logging view via the main menu (*Window* → *Show View* → *Logging*) 2.



The main elements of the process data graph area are as follows:

1 *Logging button* – Click this to show the process data graphs.

2 *View Menu* - this can also be used to show and hide the process data graph

3 *Graph canvas* – This displays the curves of all process data sets that are being recorded.

4 *Legend* – The legend lists all data sets that are displayed with their respective colors. Here you

can toggle between whether or not a curve is being displayed.

5 *Toolbar* – Here you find buttons to configure the data logging, to start and stop the recording and to navigate the display.

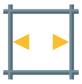
19.2 Toolbar

Opens the configuration dialog of the graphic process data logger.

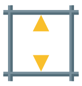
Toggles the recording of process data.

Panning tool to move the currently displayed section of the graph.

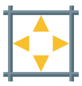
Draws a zoom-in frame to enlarge a desired area of the graph.



Auto-scales the X axis to fit all process data on the screen.



Auto-scales the Y axis to fit all process data on the screen.



Auto-scales both X and Y axes to fit all process data on the screen.

Activates auto-scaling: during a recording, both x- and y-axes are continuously rescaled to fit all process data on the screen.

Show all curves. If curves are hidden, they are displayed again.

Clear plot data. Deletes all data from the diagram.

Toggle X-axis scale. Skalierung umschalten. This switches the scaling of the X-axis between absolute date/time stamp and relative time in seconds and milliseconds since the start of recording.

Export plot image. Exports an image of the currently displayed section.

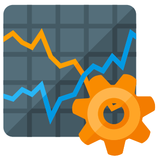
Export CSV-File. Exports all data of the plot as CSV file

Saves the plot data to a file that can later be reloaded into the plot

Loads previously saved plot data

19.3 Configuration Dialog

19.3.1 Overview



Click on the button Configure process data graph in the toolbar to open the configuration dialog.

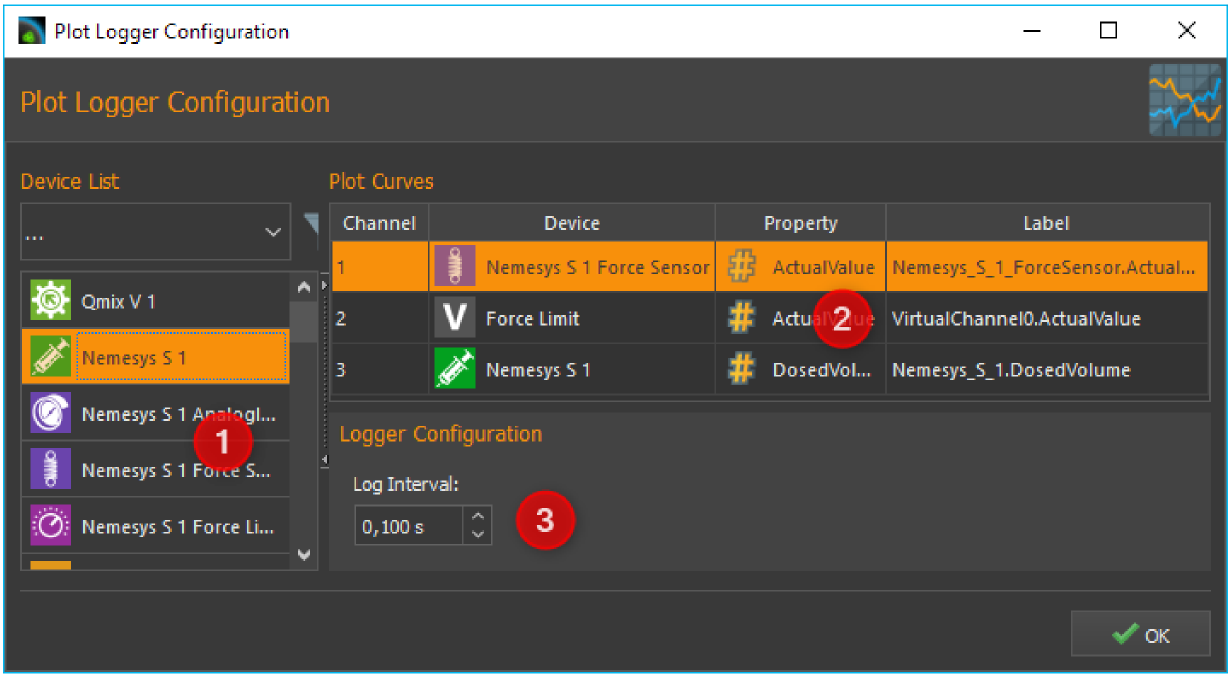


Figure 88: Configuration dialog of the graphic data logger

This opens the *Plot Logger Configuration* dialog that contains the following main sections:

1 *Device List* – shows all devices that return data that may be logged. The filter selection box allows to pre-select a specific device type (e.g., valves).

2 *Plot Curves* – lists all data series or curves that are being recorded and displayed in the diagram.

3 *Logger Configuration* – in this section you find various settings to configure the data recording.

19.3.2 *Plot Curves* Table

Channel	Device	Property	Label
1	neMESYS Mid Pressure V3 1	DosedVolume	neMESYS_Mid_Pressure_V3_1.DosedVo...
2	neMESYS Mid Pressure V3 1 AnalogIN 1	ActualValue	neMESYS_Mid_Pressure_V3_1_AnIN1.Ac...
3	Qmix V 1	ActualPosition	QmixV_1.ActualPosition
4	neMESYS Mid Pressure V3 2 DigIN 1	On	neMESYS_Mid_Pressure_V3_2_DigIN1.On

Plot Curves tabulates the selected configuration of the graphic data logger. Each row represents exactly one curve in the diagram. The following columns are shown:

- *Channel* – returns the channel number.
- *Device* – lists the device name for each respective channel including its icon.
- *Property* – shows the property of the respective device that is to be recorded. The data type is identified via a data-type specific icon.

  Numerical value

  Boolean value

  Text value

- *Label* – allows you to define a user-specific name for each channel. This label will also be used in the legend of the plotted graph.

To add and configure process data channels to the display logger, please proceed as detailed in the following sections.

19.4 Configure data logging

19.4.1 Step 1 – Adding Channels

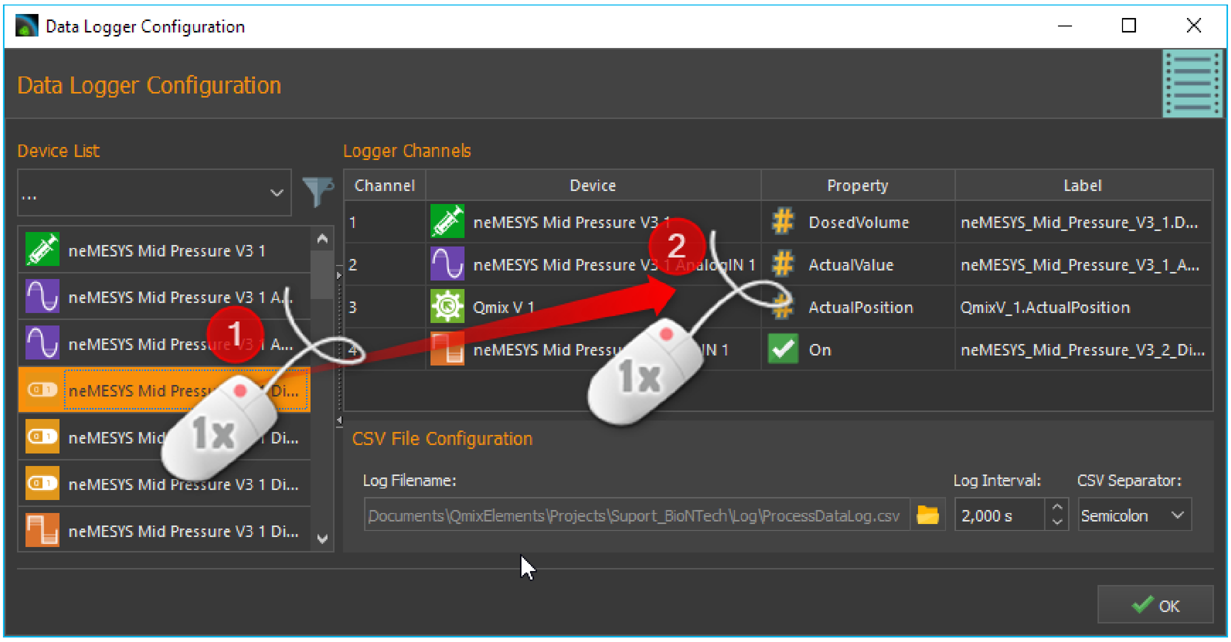


Figure 89: Adding a channel to the channel list

To add a channel you first have to add the relevant device to the *Device List* of the *Plot Logger Configuration*. To do this, move the relevant item from the device list to the *Plot Curves* table using Drag-&-Drop. The new channel will be added at the position where you release the mouse button (see figure below).

HINT. To simplify the device selection process, the device list may be filtered for a relevant device type.

19.4.2 Step 2 – Selecting the Device Property

Select the device property that you want to record by double clicking into the *Property* field of the respective channel from *Plot Curves* table. This will display a drop-down list with all available device properties from which you may select the desired item (see figure below).

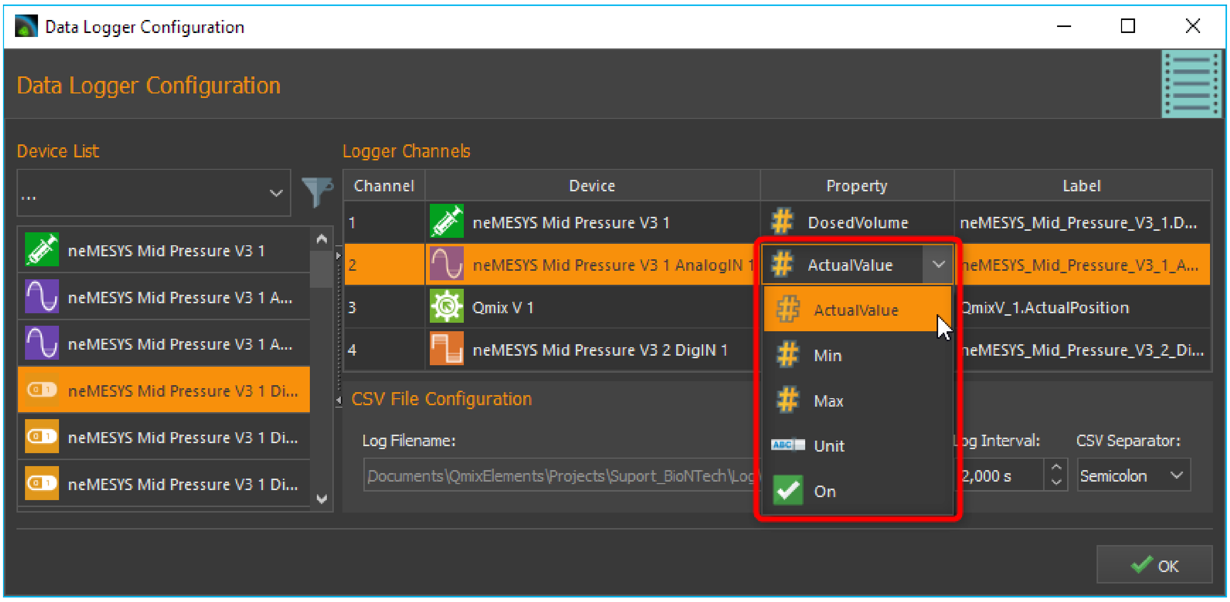


Figure 90: Selecting the device property to be recorded

19.4.3 Step 3 – Changing the Channel Label

You may give a recorded property a customized name by changing the description in the column *Label*. This label will also be used to identify the respective curve in the diagram. To do this, double click into the respective field (see figure below) and type the new description.

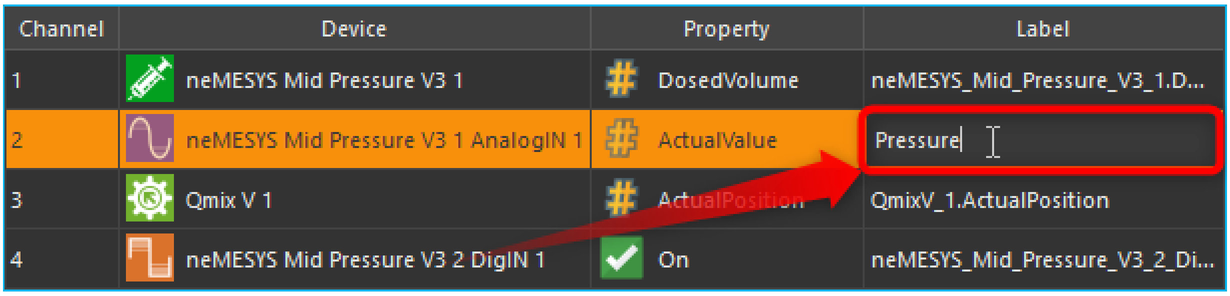
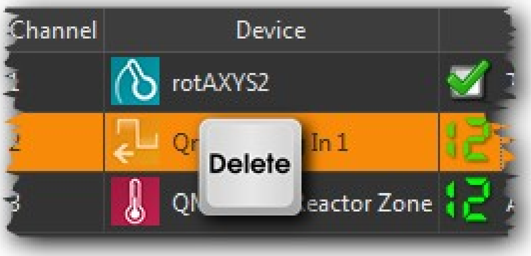


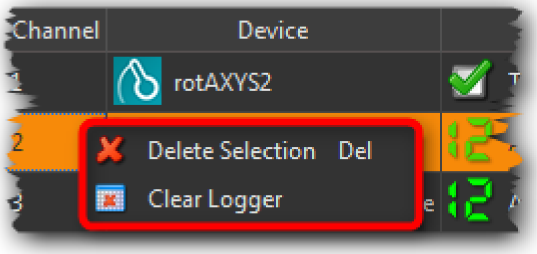
Figure 91: Changing the channel label

IMPORTANT. When a different device property is being selected, a new channel label will be assigned automatically. Therefore, the channel label should be changed after the device property has been selected.

19.4.4 Deleting Channels

In order to delete one or multiple channels from the *Plot Curves* list, first you have to mark the respective channels using the computer mouse. Now you may use either the keyboard's *Delete* key or select the relevant item (*Delete Selection*) from the right-click context menu.





You may also delete the entire list in a single step by using the *Clear Logger* item of the context menu.

19.4.5 Step 4 – Defining the Recording Interval

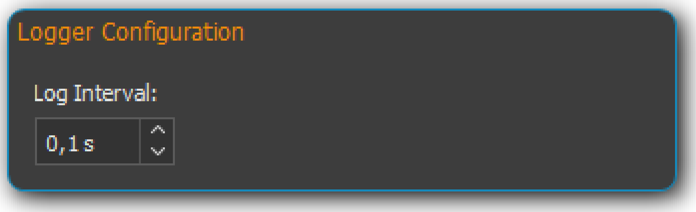


Figure 92: Configuration of the data logger interval

The *Log Interval* input box in the *Logger Configuration* section is to define the time interval at which data points for all channels are to be recorded. The minimum resolution is 0.1 seconds.

IMPORTANT. Choose a log interval that is as large as possible and as small as necessary in order to minimize the amount of data that needs to be recorded and transmitted by the system.

The configuration will be saved and reloaded automatically upon exiting the *Plot Logger Configuration* dialog.

19.5 Start/Stop Data Logging

The data logging process may be started/stopped via the relevant button in the toolbar.

19.6 Diagram Navigation & Use

19.6.1 Overview

The processdata graphing plug-in offers a number of possibilities to customize the way data are displayed. This includes resizing parts of a curve and showing or hiding individual curves.

Figure 93: The process-data diagram section

The diagram consists of a plot area 1 plus both an X-axis (time) 3 and a Y-axis (process data) 4. The time axis shows date and time as absolute values. The process-data axis shows the respective measurement data; it is without units as it potentially represents a variety of very different values and measurement units.

A right mouse click within the plot area will open a context menu 2 with a number of additional functions.

19.6.2 Changing the Displayed Section

The *Pan Tool* provides you with a simple way to move the displayed section of the plot area. It may be activated via its toolbar button and the displayed section may then be moved around using the mouse whilst keeping the left button pressed.

IMPORTANT. Panning of the displayed plot section will deactivate the auto-scaling of the diagram axes.

19.6.3 Display Curve Values

When the *Pan Tool* is active, you can move the mouse pointer over a curve to display the value at that specific position.

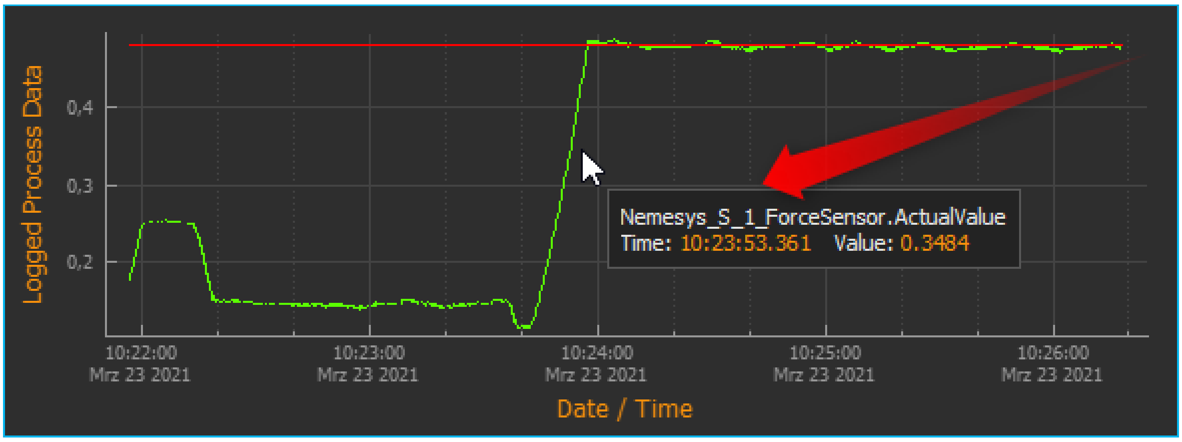
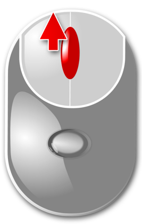


Figure 94: Display Curve Values

19.6.4 Zooming via the Mouse Wheel

Turning the mouse wheel whilst the pointer is within the plot area will allow you to adjust the displayed section of a graph by increasing (zooming in) or decreasing (zooming out) its zoom level.



Increase zoom level (zoom in)

Decrease zoom level (zoom out)

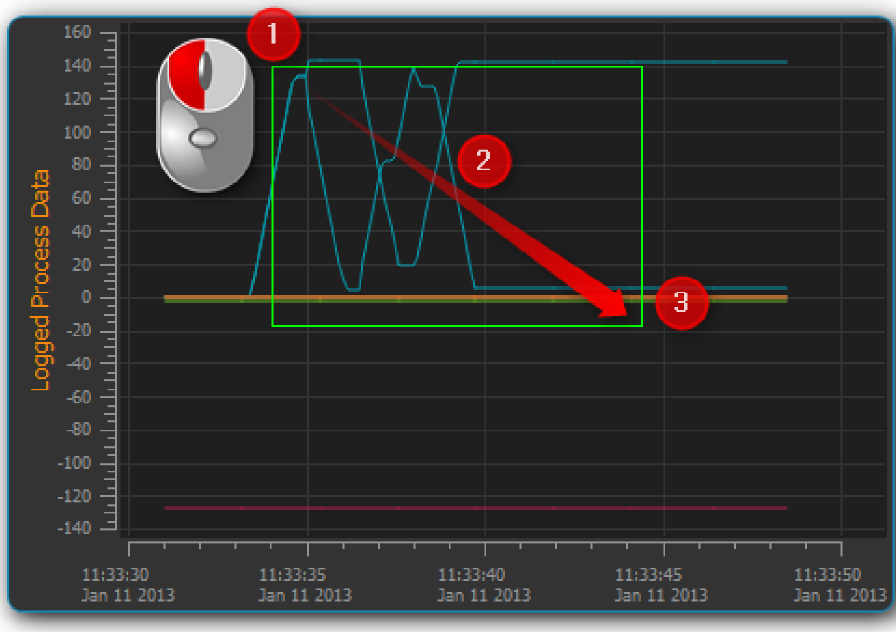
19.6.5 Defining a Display Section

The *Zoom Tool* allows you to directly select a specific area of the plot and increase its resolution. To do this, please proceed as follows (see figure below):

1 Using the mouse, left-click-and-hold into the plot area to set the first corner of the zoom frame.

2 Move the mouse pointer to define the size of the frame as desired.

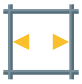
3 Releasing the mouse button will finalize the size of the frame. The selected area will be scaled to the current graph size (zoom in).



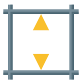
19.6.6 Auto-Fit & Auto-Scale

The toolbar and the context menu both contain a number of tools to adjust what is displayed in the diagram, in particular to ensure that all or specific data are visible.

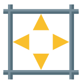
The following possibilities exist:



Rescales the X axis to display all current time data values for a given process data resolution.



Rescales the Y axis to display all current process data values within a given time period.



Rescales both X and Y axes to display all currently available data.

(Re-)activates auto-scaling: as long as data are being recorded, both X and Y axes will be adjusted dynamically to ensure all data are being displayed.

You may also activate auto-scaling for X and Y axes individually via the context menu:

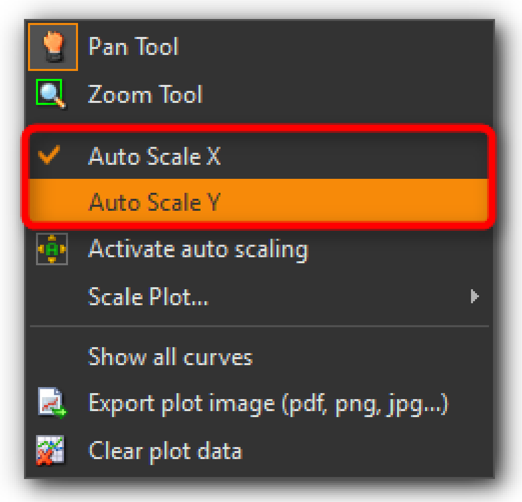


Figure 95: Auto-scaling toggle for X and Y axes

IMPORTANT. Zooming or panning within the displayed plot section will deactivate auto-scaling.

19.6.7 Show/Hide Individual Curves

To improve scaling and visibility, you may show or hide individual curves. To do this, right-click the desired item in the plot legend and select the desired function to either hide the corresponding curve only (*Hide Curve*) or all other but the corresponding curve (*Show only this curve*) as indicated in the figure below.

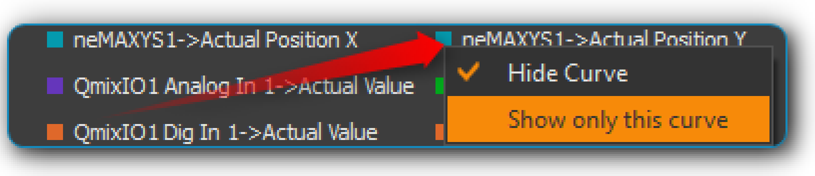
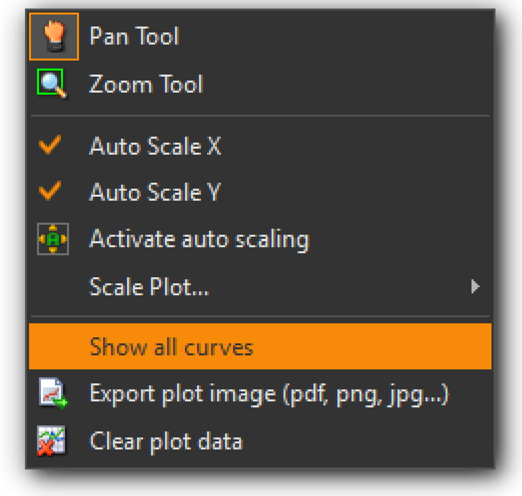


Figure 96: Context menu legend item

To revert to displaying all curves, activate the context menu from within the plot area and select the menu item *Show all curves* (see figure below).



19.6.8 Select Curve Color

To choose a different curve color, right click an item in the plot legend. From the context menu select the menu item *Select Color* (see figure below).

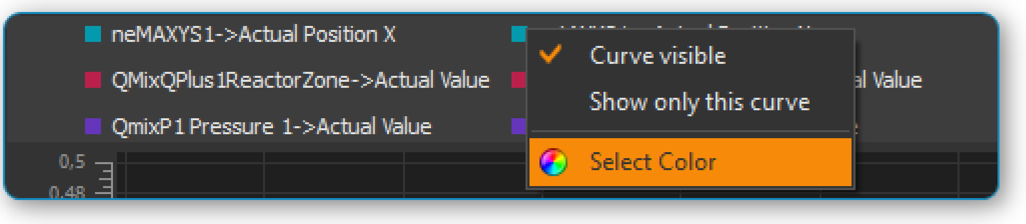


Figure 97: Context menu legend item – Color Selection

In the color selection dialog which is now shown (figure below), you can choose any color.

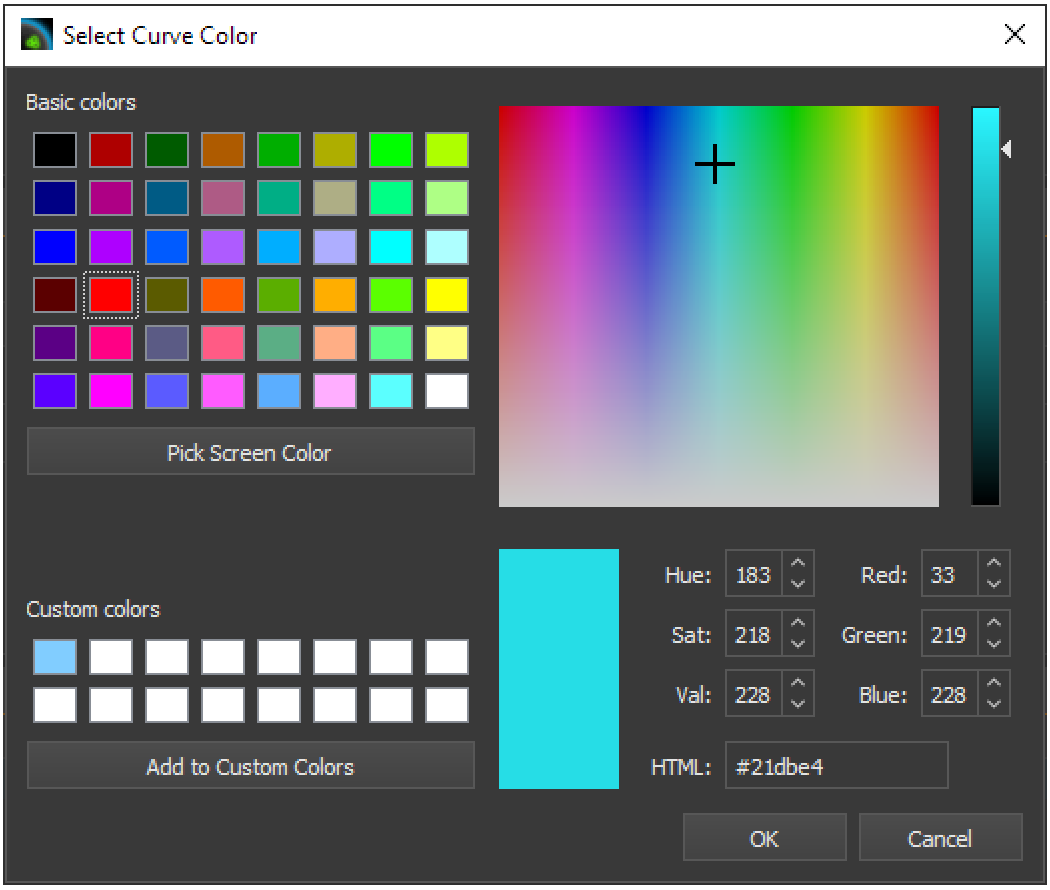


Figure 98: Color Selection Dialog

19.6.9 Exporting Plot Image

You may export a picture of the current diagram using the right-click context menu and selecting *Export plot image*.

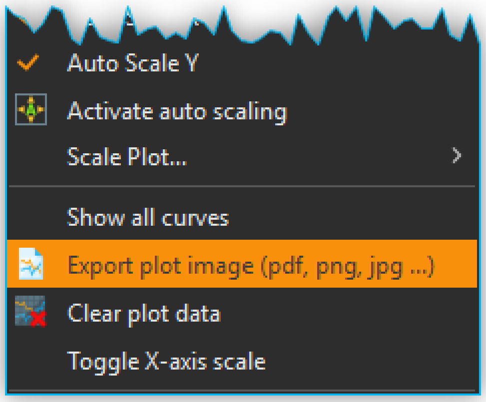
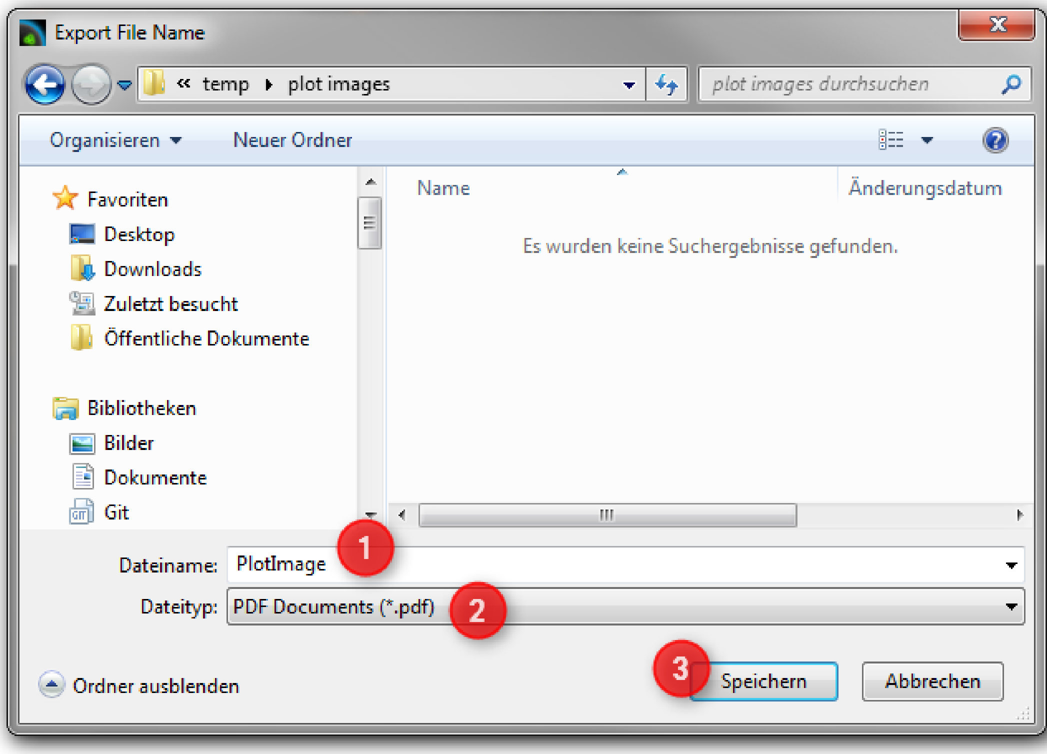


Figure 99: Diagram image export

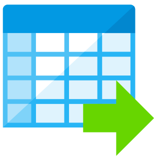
This will open a dialog box (see figure below) to define the location (folder) where the image is to be saved:



Please enter a name for the image file 1 and select the desired file type 2. The export function supports standard image file formats (*png, jpg...*) as well as scalable vector graphic formats (*pdf, svg...*).

To close the dialog and to start the image export, click *Save* 3.

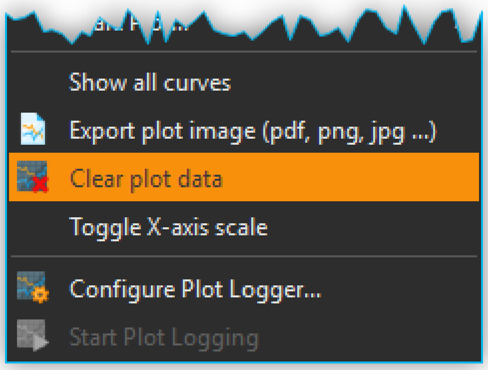
19.6.10 CSV Export



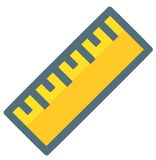
You can export all diagram data to a CSV file using the *Export CSV file* menu item.

19.6.11 Deleting of Diagram Data

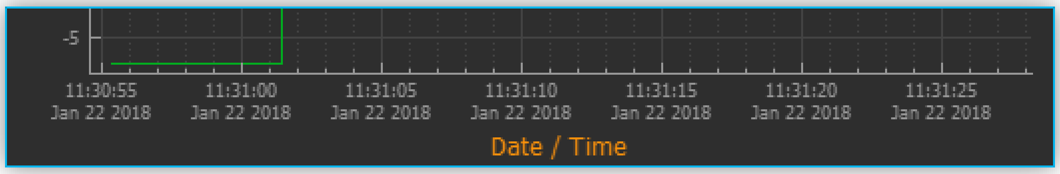
You may clear the plot area and delete all data recorded since the start of the present recording using the context menu item *Clear plot data*. Recording will resume from this point.



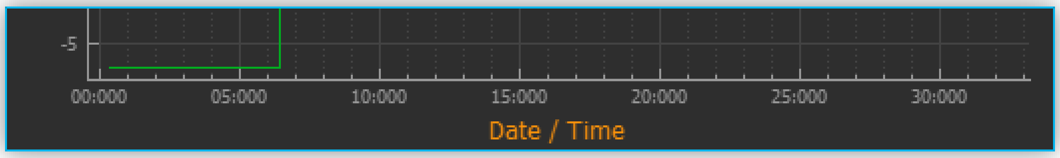
19.6.12 Switching the scaling of the X-axis



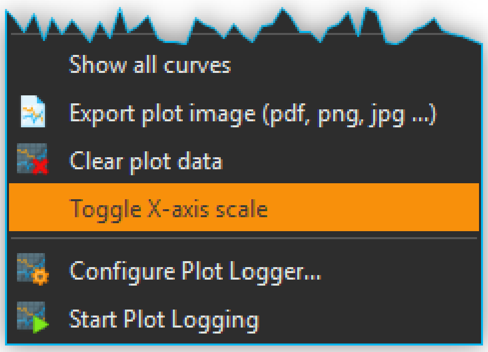
You can switch the scaling of the X-axis between two different modes. By default, the X axis displays an absolute date/time stamp.



You can switch the X-axis to display the relative time in seconds and milliseconds. This means that the event t_0 marks the point in time at which the recording was started.



To toggle the axis, right-click in the diagram and select *Toggle X-axis scale* from the context menu.



19.6.13 Saving plot data

If you click the *Save Plot Data* button, all plot data will be saved to a file (*.dat*) that can be loaded back into the plot later.

19.6.14 Loading plot data

By clicking the *Load Plot Data* button, plot data that was previously saved with *Save Plot*

Data can be loaded back into the plot. Only the curves that are present in the current configuration of the logger are loaded. I.e. if you record data, save it with *Save Plot Data* and load it again later, the logger configuration should be identical when saving and loading. If you change the logger configuration between saving and loading, e.g. remove channels, not all curves may be loaded.

19.7 Script Functions

19.7.1 Introduction

To automate the capture of data or to synchronize data capture with other processes, the graphical plot logger can be started and stopped using QmixElements script functions. The corresponding functions can be found in the *Logging* category in the list of the available script functions.

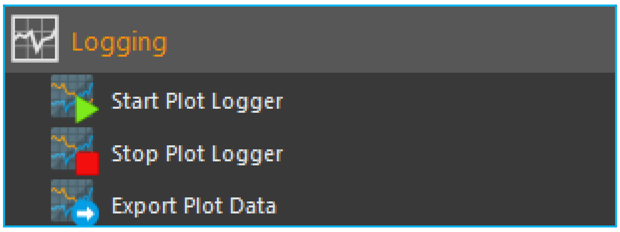
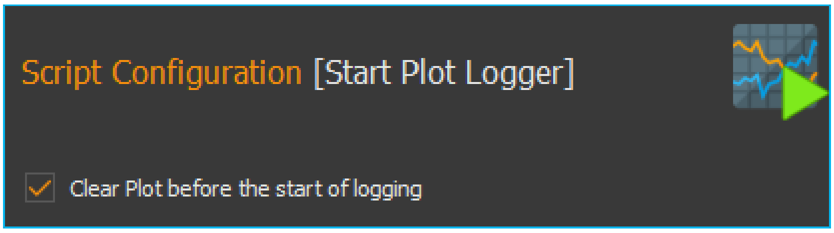


Figure 100: Logger script functions

19.7.2 Start Plot Logger

This function is used to start the graphical logger with the currently configured settings and channels. The current content of the plot is not deleted.

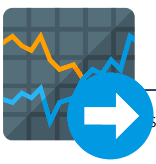


Check *Clear Plot before the start of logging* if you want to clear all plot data before logging. Starts.

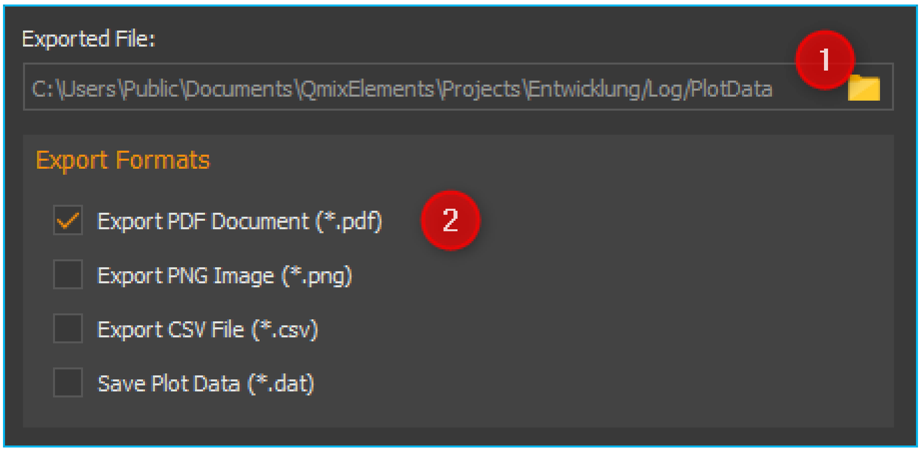
19.7.3 Stop Plot Logger

This function stops the current logging of process data into the process data plot.

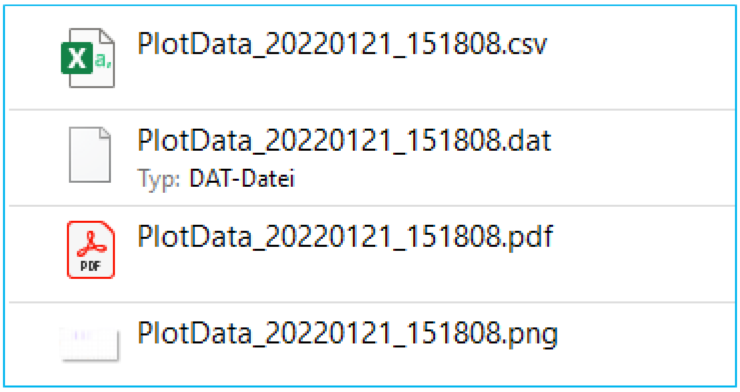
19.7.4 Export Plot Data



This function allows you to export the plot data to different formats. In the configuration area you can choose the file name and the saving location by clicking on the folder icon 1. For the saving location, you should keep the default location within the project folder.



In the Export Formats 2 area, select all formats you want the plot data to be exported in. The software saves the files with the selected file name + timestamp + the file extension of the export format (see example in figure below):



20 CANopen Tools Plugin

20.1 Introduction

This plugin contains tools for accessing and configuring CANopen devices. CANopen is a standardized application for industrial automation solutions, based on CAN. Originally, CANopen was developed to control machine networks. Today, CANopen is used in many fields, including medical equipment, vehicle production, shipping and public transport.

The CANopen profile family is based on a communication profile (communication profile for industrial systems DS-301), which specifies communication mechanisms and their description. The different device types used in automation equipment, such as drives and controllers, are described in device profiles. These device profiles determine the functionality and parameters of standard devices of the respective types. Such standardized profiles form the basis for uniform access to CANopen devices using the CAN bus. This makes it possible to be largely independent from particular manufacturers.

20.2 Object Dictionary

The object dictionary forms the central element of the CANopen standard. It describes the complete device functionality of a CANopen device. Each CANopen node implements a local object dictionary. It is basically a standardized and predefined grouping of objects that can be accessed through the network.

Each object in the dictionary is accessed through a 16-bit index and an 8-bit sub index. The entries in the object dictionary make it possible to access the “application objects” of a device, such as input and output signals, device parameters, device functions or network variables, through the network in a standardized fashion.

Index	Object
0000h	Reserved
0001h-009Fh	Data Types
00A0h-0FFFh	Reserved
1000h-1FFFh	Range for communication profile DS-301
2000h-5FFFh	Manufacturer-specific range
6000h-9FFFh	Standardized range of the implemented device profile
A000h-FFFFh	Reserved

Table 1: Layout of the CANopen Object Dictionary

Simple variables can be accessed directly through the 16-bit index. In case of data structures or arrays, the index addresses the entire data structure. The additional 8-bit sub index allows access to individual elements of a data structure or array.

Index	Sub Index	Data Type	Name
1000h	0	UNSIGNED32	Device type
1001h	0	UNSIGNED8	Error register
1018h		RECORD	Identity object
	0	UNSIGNED8	Number of entries
	1	UNSIGNED32	Vendor ID
	2	UNSIGNED32	Product code
	3	UNSIGNED32	Revision number
	4	UNSIGNED32	Serial number

Table 2: Excerpt from an Object Dictionary

The object dictionary is divided into two sections. The first section contains information regarding the device, such as device identification, manufacturer, etc. as well as communication parameters. The second section describes the specific device functionality.

20.3 Overview of CANopen Tools Workbench

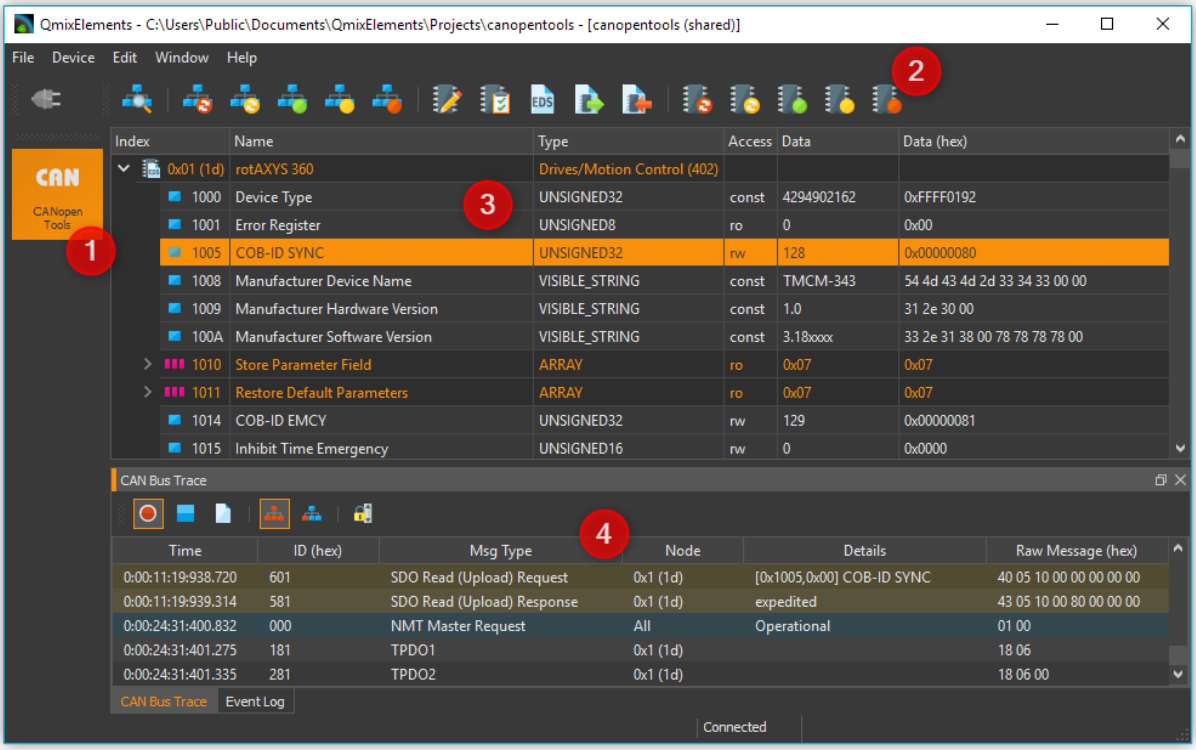


Figure 101: CANopen Tools Workbench

The Workbench of the *CANopen Tools* plugin can be displayed by pressing the *CANopen Tools* ❶ button in the sidebar. The toolbar ❷ contains important functions for accessing devices on the network.

The main part of the CANopen Tools Workbench is the Object Dictionary Editor ❸ for reading or writing individual entries to or from the object dictionary.

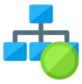
In addition, you can monitor the messages on the CAN bus in real time with the CAN Bus Trace Window ❹.

20.4 Toolbar

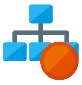
 Network scan – searches for connected devices

 Resets all devices

 Resets communication parameters of all devices

 Starts process data communication on all devices

 Sets all devices to pre-operational status

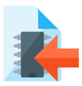
 Stops communication of all devices

 Saves parameters of selected devices (nodes) to non-volatile device memory

 Resets all parameters of selected device to factory values

Assigns EDS file (*Electronic Data Sheet*) to the selected device

 Export parameters of the selected CANopen node to a DCF file

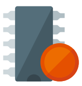
 Import DCF file into the selected CANopen node

 Resets the selected device

 Resets communication parameters on the selected device

 Starts process data communication on the selected device

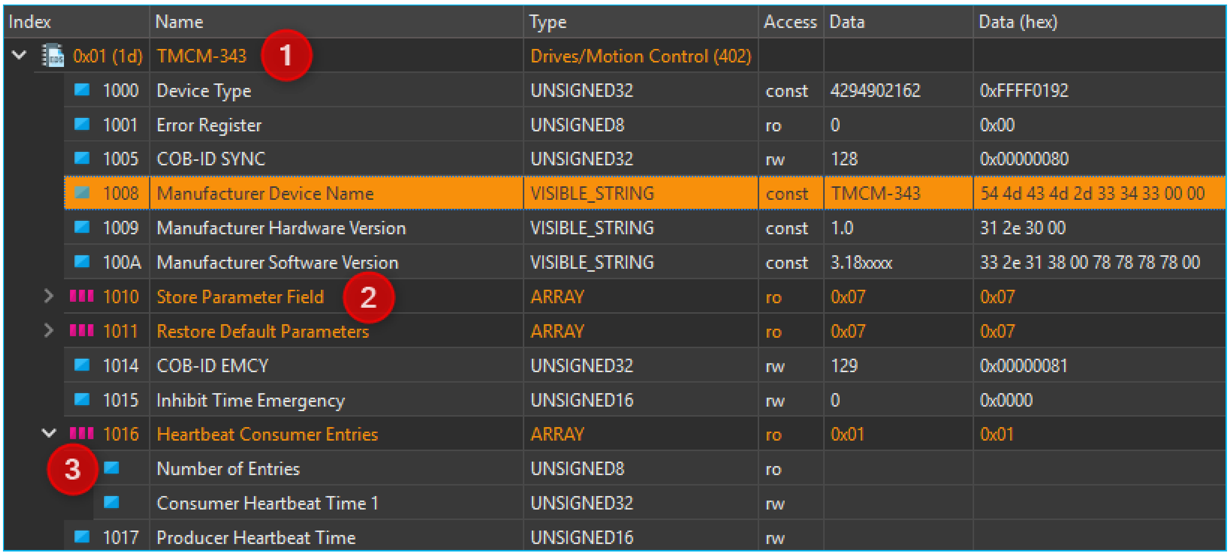
 Resets selected devices to pre-operational status

 Stops communication of the selected device

20.5 Object Dictionary Editor

20.5.1 Overview

The object dictionary editor gives you access to the object dictionaries of all connected nodes.



Index	Name	Type	Access	Data	Data (hex)
0x01 (1d)	TMCM-343 1	Drives/Motion Control (402)			
1000	Device Type	UNSIGNED32	const	4294902162	0xFFFF0192
1001	Error Register	UNSIGNED8	ro	0	0x00
1005	COB-ID SYNC	UNSIGNED32	rw	128	0x00000080
1008	Manufacturer Device Name	VISIBLE_STRING	const	TMCM-343	54 4d 43 4d 2d 33 34 33 00 00
1009	Manufacturer Hardware Version	VISIBLE_STRING	const	1.0	31 2e 30 00
100A	Manufacturer Software Version	VISIBLE_STRING	const	3.18xxxx	33 2e 31 38 00 78 78 78 78 00
1010	Store Parameter Field 2	ARRAY	ro	0x07	0x07
1011	Restore Default Parameters	ARRAY	ro	0x07	0x07
1014	COB-ID EMCY	UNSIGNED32	rw	129	0x00000081
1015	Inhibit Time Emergency	UNSIGNED16	rw	0	0x0000
1016	Heartbeat Consumer Entries	ARRAY	ro	0x01	0x01
3	Number of Entries	UNSIGNED8	ro		
	Consumer Heartbeat Time 1	UNSIGNED32	rw		
1017	Producer Heartbeat Time	UNSIGNED16	rw		

Figure 102: Object Dictionary Editor

The editor uses a tree-like structure with 3 levels. Level 1 shows all devices detected during a network scan. If you open a network node, you will see level 2 with the complete object dictionary of that node and all its dictionary entries. You can access simple variables in this level directly through the index. Complex data structures or arrays have an additional level 3. Here you access individual elements of an array or data structure, using the sub index. Entries of data structures with sub entries are highlighted in color 2.

The following overview shows you the structure once again:

Devices on CAN Bus (level 1)

Simple elements with direct index access (level 2)

Array elements – expand to access the individual elements

Data structure – expand to access the individual elements

Individual elements of arrays or data structures – access via index and sub-index (level 3)

20.5.2 Assigning Device Names

After a network scan, all devices are identified by their unique node number (index column). In the name column you can also assign a description of your choice in the name column to make identification easier.

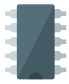
Figure 103: Assigning Device Names

Using the left mouse key, double-click on the node name field and enter a name of your choice.

20.5.3 Assigning an Electronic Data Sheet (EDS)

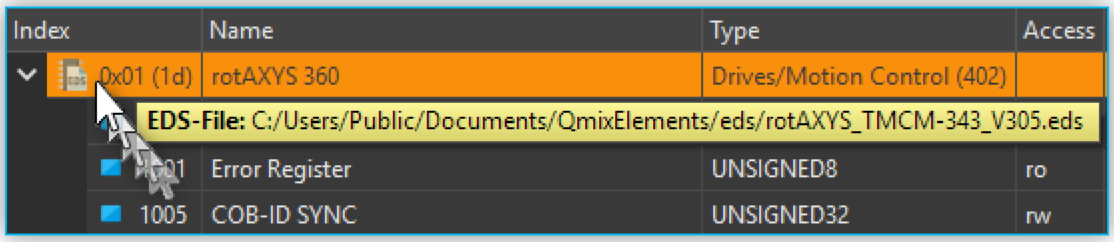
The object dictionaries of different nodes contain different entries. After a network scan you will only be shown the standard entries, that are specified according to the CANopen DS301 standard. Device profile specific or manufacturer-specific entries are omitted. In order to access those entries, you will need an EDS-file (electronic data sheet) for your device.

The symbol of the device in the *Index* column indicates whether a device has already been assigned an EDS file.

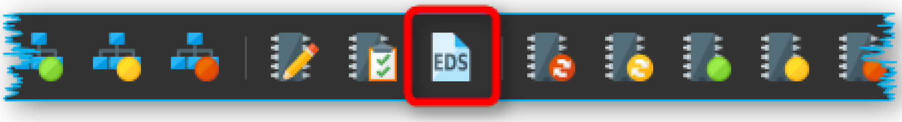

CAN node without EDS file

CAN node with EDS file assigned

If you want to know which EDS file has been assigned to a node, simply move the mouse over the node and wait until the help text appears. Here you can see the complete file path of the assigned EDS file.



To assign an EDS-file first select the device in the object dictionary editor by clicking on it. Then click the Assign EDS File button in the toolbar.



As an alternative, you can also open the context menu of the object dictionary editor by doing a right mouse click and selecting Assign EDS File from the menu.

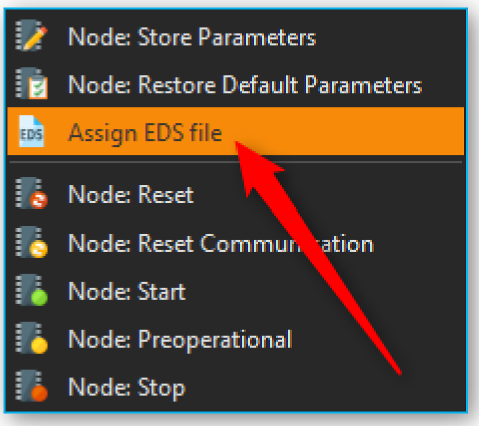


Figure 104: Object Dictionary Context menu

After assigning the EDS-file you will be able to access all device parameters.

IMPORTANT. Configuration parameters like device names or assigned EDS-files are saved automatically in the project settings of the current project and are restored upon relaunching the application or loading a project.

TIP. Create different projects (*File → Save Project*) to be able to change quickly between different network configurations.

20.5.4 Reading/Writing Object Dictionary Entries

When you click on an object dictionary entry the respective entry is imported from the device and the values filled into the Data and Data (hex) columns.

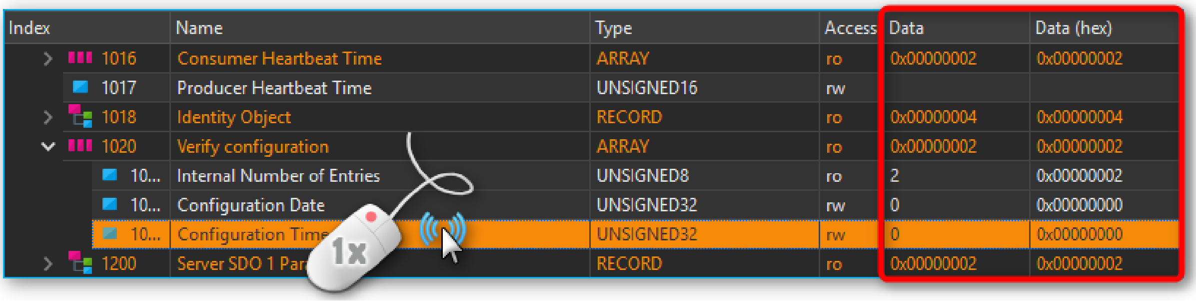


Figure 105: Reading an Object Dictionary Entry

To write device parameters simply perform a double-click with your left mouse button on the Data or Data (hex) column of the object dictionary entry you wish to modify.

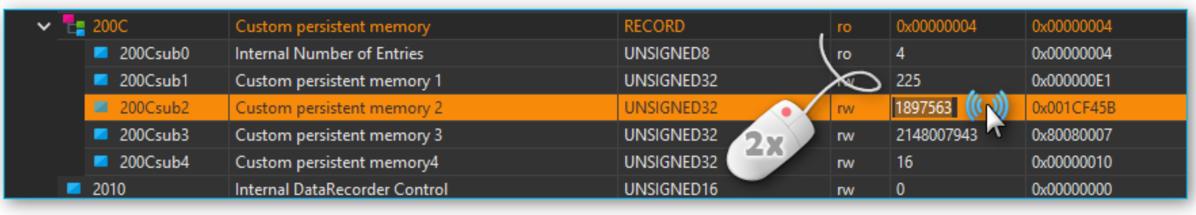


Figure 106: Writing an Object Dictionary Entry

After double-clicking, you can modify the values of the cell. When you press the enter key or click on a different cell, the value is accepted and transferred to the device.

TIP. You can only write object dictionary entries whose access type is set to read/write (rw) or write only (wo).

IMPORTANT. Unless you saved the modified data to the non-volatile device memory using the Save Parameters function, any parameter modifications will be lost when the device is turned off or reset.

20.5.5 Permanently saving / restoring Device Parameters

Any object dictionary entries you created will be lost when the device is turned off or reset. In order to permanently save device parameters to the non-volatile device memory, you have to click on Node: Store Parameters in the toolbar (see image below) after writing the entry.

If you want to restore a device's default parameters (factory settings), click on Node: Restore Default Parameters in the toolbar.

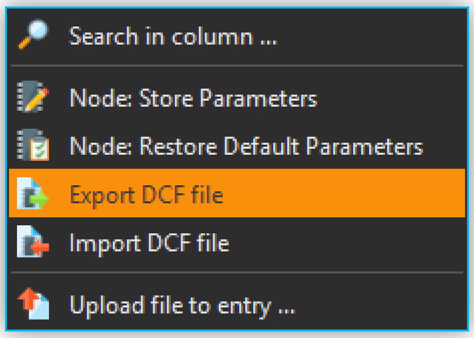
All parameter modifications saved on the device will be lost and replaced by the default parameters.

20.5.6 Exporting Device Parameters to DCF File

You can export the complete configuration of a CANopen node to a standardized file format as a DCF (**D**evice **C**onfiguration **F**ile) file.

IMPORTANT. A DCF file is an EDS file with the current values of each object. This means that the DCF export can only be performed for nodes with an assigned EDS file.

To start the parameter export, simply right-click on the node or on an object directory entry of the node and select *Export DCF file* from the context menu.

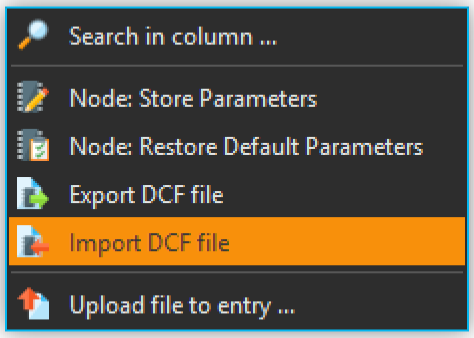


20.5.7 Importing Device Parameters from DCF File

You can import the complete configuration of a CANopen node from a DCF file.

IMPORTANT. A DCF file is an EDS file with the current values of each object. This means that the DCF import can only be performed for nodes with an assigned EDS file.

To start the parameter import, simply click with the right mouse button on the node or on an object directory entry of the node and then select the menu item *Import DCF file* in the context menu.

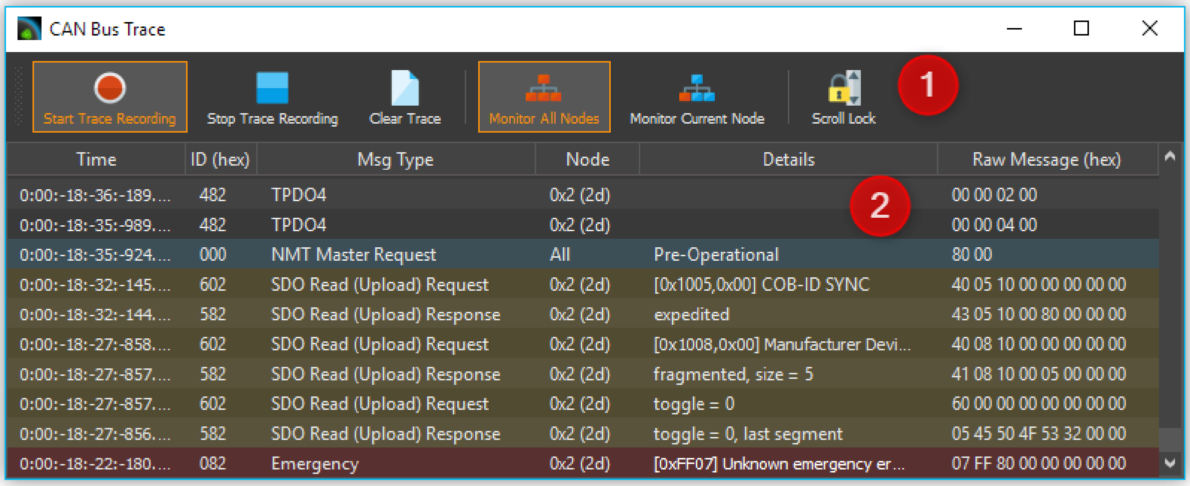


After the import, the imported parameters are not yet permanently stored in the non-volatile memory of the device. This means that you have to save the parameters explicitly using the menu item *Node: Store Parameters*.

20.6 CAN Bus Trace Window

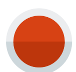
20.6.1 Overview

You can use the *CAN Bus Trace* to monitor and record messages on the CAN bus in real time. This tool is available as a freely positionable tool window. This means that you can move it to any position or detach it from the application and position it on a second screen.



The Trace window basically consists of the toolbar 1 and the list of recorded messages 2.

20.6.2 Toolbar

 Starts the recording of CAN messages

 Stops the recording of CAN messages

 Deletes all recorded messages from the trace list.

 Activates the recording of messages from all CAN nodes

 Only the messages of the currently selected node are recorded.

Enables / disables automatic scrolling to the last recorded message

20.6.3 The Trace-List

In the trace list, all messages are displayed in chronological order, i. e. older messages are at the top of the list and more current messages are at the bottom of the list. The last entry in the list is always the last recorded message.

If automatic scrolling is activated, the display is always scrolled to the last recorded message. If you want to view older messages during recording, deactivate automatic scrolling.

The different types of CANopen messages are displayed differently colored in the list to increase clarity. The following colors are used:

Not colored – PDO messages

Blue – Network management messages (NMT), Bootup messages and messages for node monitoring (Heartbeat & Node Guarding)

Yellow – SDO messages

Red – Emergency messages

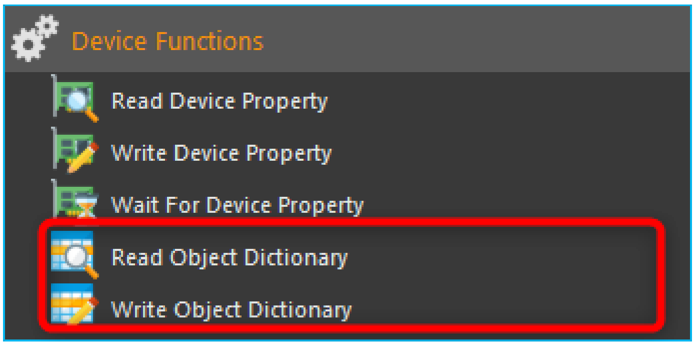
You can toggle whether you want to record the messages of all nodes or just the messages of a single node. If you activate recording for a single node, only the messages of the node selected in the Object Dictionary Editor are recorded.

To record messages from another node, you must select the other node in the Object Directory Editor. Then activate the recording for all nodes and then reactivate the recording of a single node.

20.7 CANopen Script Functions

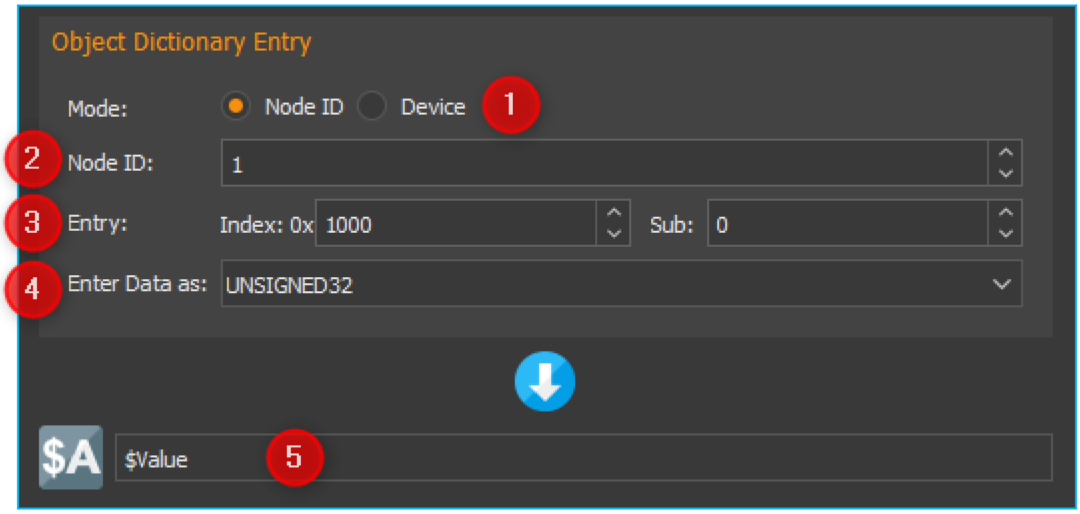
20.7.1 Introduction

The CANopen Tools Plugin provides functions for the Scripting System to read and write object dictionary entries of connected CANopen devices. The corresponding functions can be found in the *Device Functions* category:



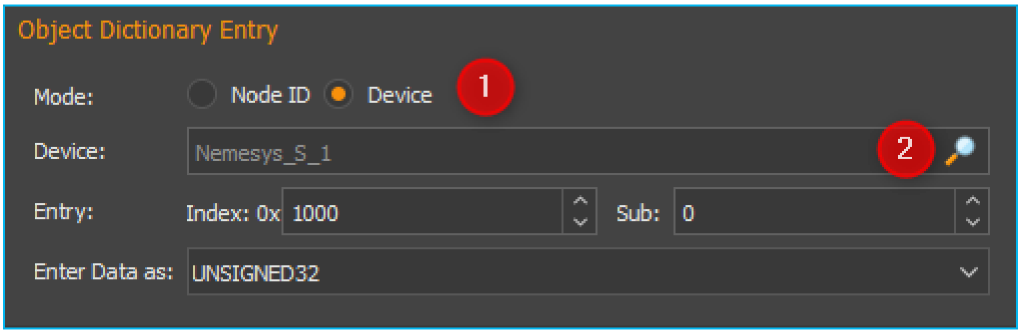
20.7.2 Read Object Dictionary

This function allows you to read an object dictionary entry and save the read value in a script variable. Use *Mode* ❶ to select whether you want to use the CANopen node ID (*Node ID*) to access the device or whether you want to access the device via its device name.



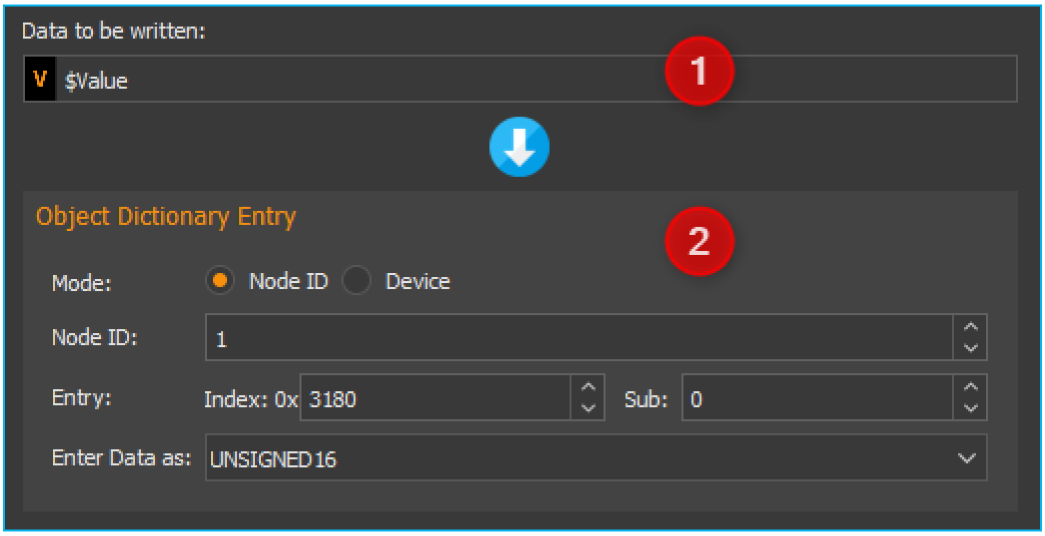
If you have selected *Node ID* for *Mode*, enter the node number of the CANopen device in the *Node ID* ❷ field. For *Entry* ❸, select the object dictionary entry to be read. This consists of *Index* (hexadecimal) and *Sub Index* (decimal). Then select the data type of the entry at 4. Finally, enter the name of the variable in which the read value is to be saved in the variable field 5.

If you have selected the option *Device* 1 in *Mode* (figure below), click on the magnifying glass icon 2 to select a device:



20.7.3 Write Object Dictionary

By using this function you can write a value into an object dictionary entry of a CANopen device. In the field *Data to be written* 1 enter the value to be written into the object dictionary entry. Alternatively, you can enter the name of a script variable to write its value into the object dictionary.



In the *Object Dictionary Entry* area, you specify the object dictionary entry of a particular device to be written. Details on this can be found in the documentation for the script function *Read Object Dictionary*.

21 Spectroscopy Add-on

21.1 Installation

The spectroscopy plugin is not part of the standard installation package and must be installed as add-on. The versions of the spectroscopy plugin and the installed CETONI Elements software should match. For example, if you want to install the spectroscopy add-on version 20180520, you should have CETONI Elements version 20180520 installed.

IMPORTANT. The version number of CETONI Elements and the spectroscopy plugin should match.

Please close all other programs before launching the installation.

IMPORTANT. Install the spectroscopy plugin and the device driver before connecting your spectrometer to your PC through USB for the first time.

Launch *CETONI_Elements_Spectroscopy_Setup_64bit.exe* to begin the installation. The installation assistant will guide you through the installation of the software and hardware drivers.

IMPORTANT. If you want to install the hardware drivers on a windows system, you need to be logged in as an administrator.

IMPORTANT. During the installation you can also install the optional Analyze IQ add-on, for chemometric (quantitative and qualitative) analysis of recorded spectrums. For this add-on you will need a paid runtime license.

21.2 Introduction

Clicking the Spectroscopy button 5 in the sidebar will take you to the Spectroscopy Plugin (see image below).

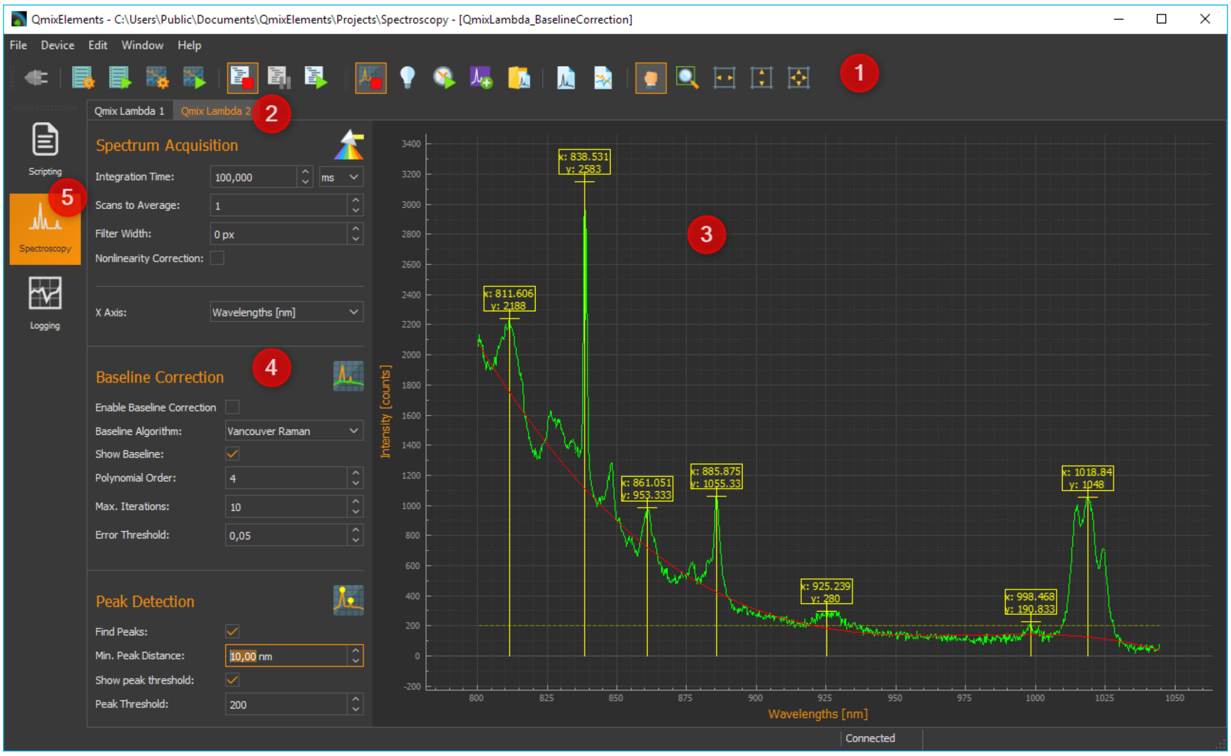


Figure 107: Overview of spectroscopy workbench

1 Toolbar

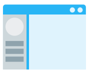
2 Tabs for the selection of a spectrometer display. Using the tabs you can switch between the displays and control elements of individual spectrometers. One tab is created for each spectrometer.

3 Spectrum live display

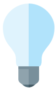
4 Parameter control elements

5 *Spectroscopy button*

21.3 Toolbar

 Show/hide the sidebar with the spectrometer settings

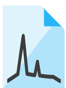
 Start / Stop spectral data live display

 Record / Delete a background spectrum

 Start / Stop determination of integration time

 Create a spectrum analysis channel

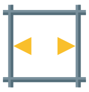
 Load spectrometer settings from an *.aiq file

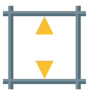
 Save spectral data in text format (*.txt) or as Analyze IQ file (*.aiq)

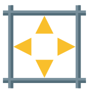
 Save spectral data in PNG image file or PDF document

 Panorama tool to move the spectrum live display

 Zoom area selection tool (enlargement)

 Scale x-axis to fit the visible screen area

 Scale y-axis to fit the visible screen area

 Scales both axes to fit the visible screen area

21.4 Spectrum Live Display

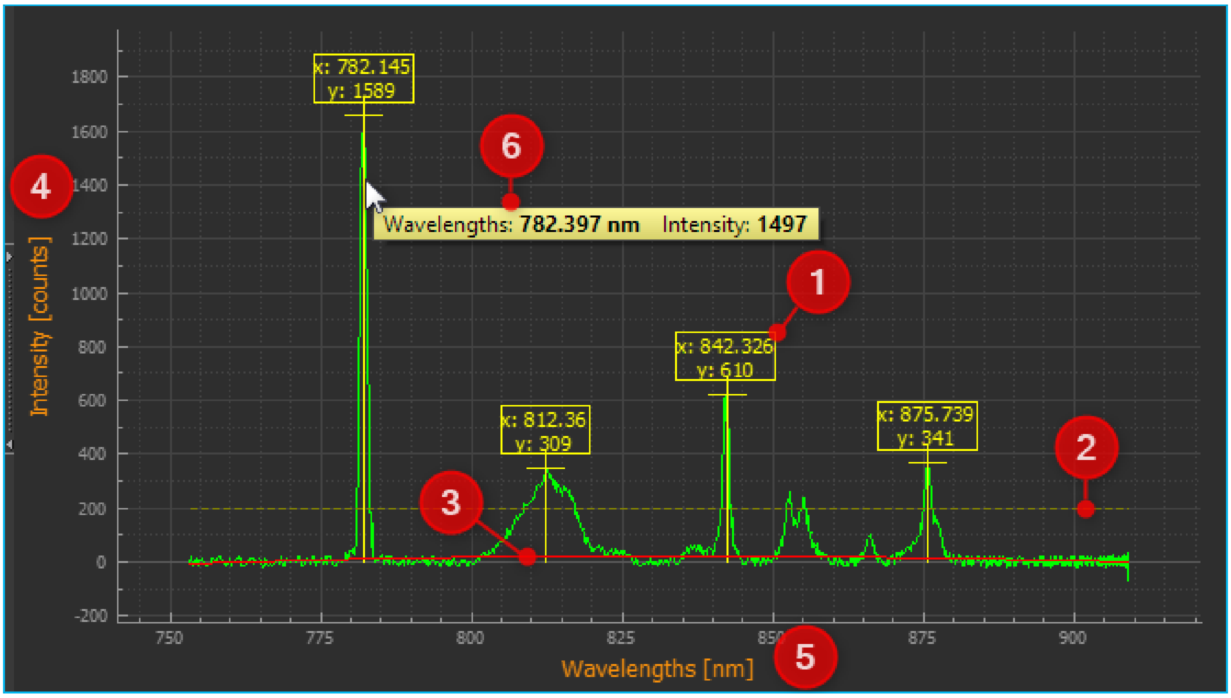


Figure 108: Spectrum Live Display

The spectrum live display shows the current, last recorded spectrum following all processing steps. The peak markers 1 show the x and y coordinates of a determined spectrum peak. The line for the minimum peak intensity is shown in yellow 2. Peaks below this line are not detected and marked. If you have activated the display of the calculated baseline in *Baseline Correction*, it is displayed as a red curve 3. The x-axis 5 shows the set unit (Wavelengths in this example). The y-axis 4 shows the measured intensity in counts.

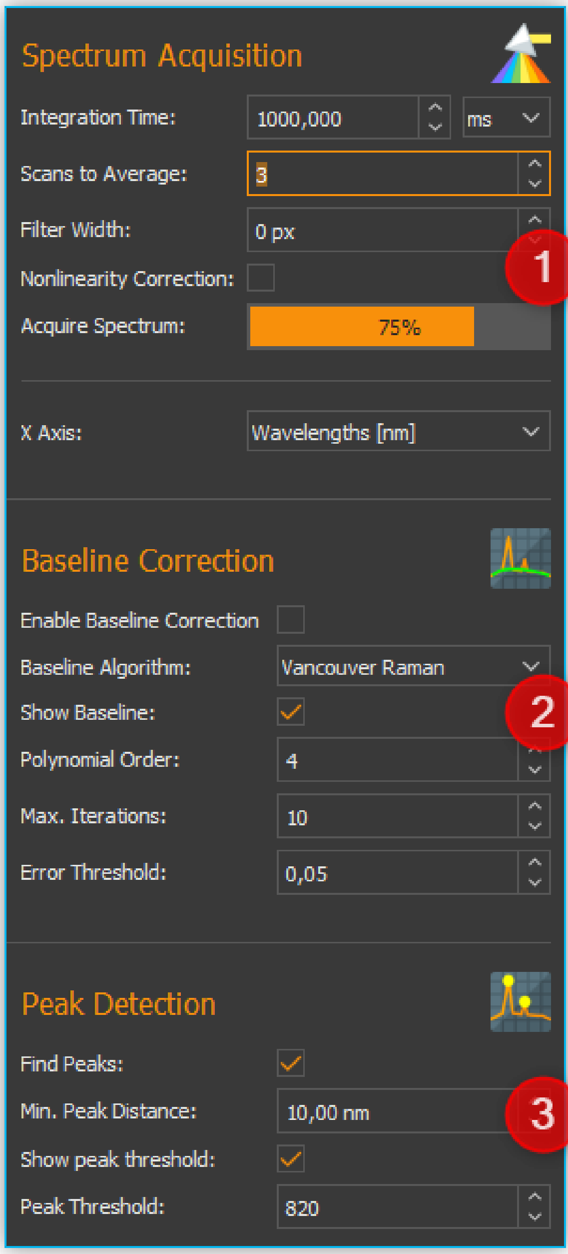
You can use the mouse wheel to zoom in and out of the spectrum display.

If the *Panorama tool* for moving the display area is activated, you can display the values of the spectrum at a certain position by moving the mouse pointer to the corresponding position over the curve 6. A window with the values will appear. If you move the mouse pointer over the curve, the window follows the mouse pointer and the values inside are continuously updated.

21.5 Parameter Control Elements

21.5.1 Overview

The controls are used to set the parameters for acquisition and pre-processing of spectra. They are arranged in the order in which processing is carried out from top to bottom.



1 **Spectrum Acquisition** - contains all settings of the spectrometer for the acquisition of a spectrum

2 **Baseline Correction** - here you can set the parameters for real-time baseline correction

3 **Peak Detection** - you configure the parameters for automatic peak detection here

The Spectrum Live Display lets you see and evaluate the effects of parameter changes right away.

21.6 Spectra Acquisition

21.6.1 Overview of controls for spectra acquisition

In the *Spectrum Acquisition* area you will find all controls for setting the acquisition parameters.

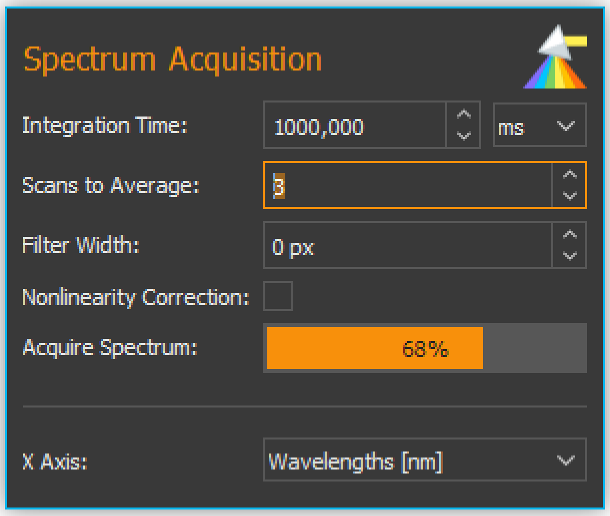


Figure 109: Spectrum acquisition controls

21.6.2 Setting the Integration Time

The integration time correlates with the exposure time of the spectrometer's CCD. It is the time period in which the detector can collect photons.

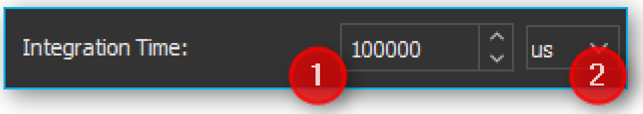


Figure 110: Setting the Integration Time

Enter the integration time into the input field 1. Set the desired time unit in the selection field 2. As the integration time increases, the spectrums intensity values go up as well, since the detector has more time to collect photons.

TIP. The recording duration of a spectrum increases along with the integration time.

If the recording duration is longer than 500 milliseconds, a progress bar is displayed below the control elements, showing you the progress of the current recording. If the product of integration time and scans to average is 10 seconds or greater, a warning message is displayed showing the future length of the integration time. You can then either confirm the current parameter settings or retain the original values.

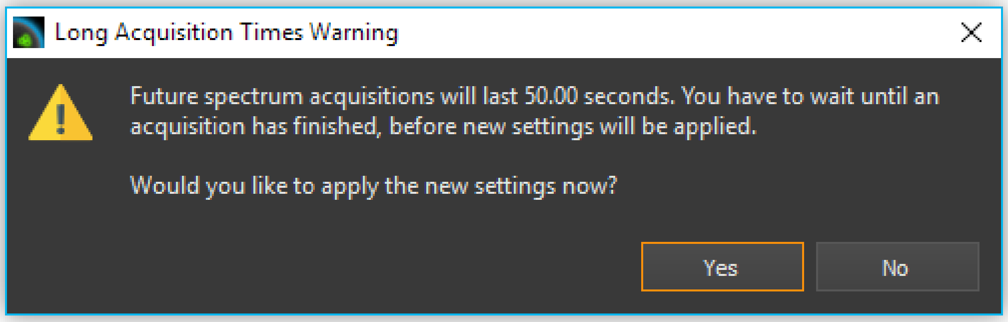


Figure 111: Information Window in Case of Excessive Integration Time

IMPORTANT. Any parameter changes are executed after ending the current spectrum recording operation. During long integration periods this may cause the impression that the software is no longer responsive.

21.6.3 Automatically determining Integration Time

If you click the button for automatic integration time determination in the toolbar, the software will attempt to control the integration time automatically in such way that the maximum intensity of the spectrum reaches 80% of the maximum sensor value (saturation).

You can stop the automatic integration time determination at any time by clicking the button again.

21.6.4 Pre-processing the Spectrum

Pre-processing live spectrums can contribute to an improved signal-to-noise ratio (SNR). This reduces interference, while improving spectrum quality. Spectral smoothing makes it possible to determine distinctive peaks. The removal of non-linearities and spectral baseline correction also helps improve quality.

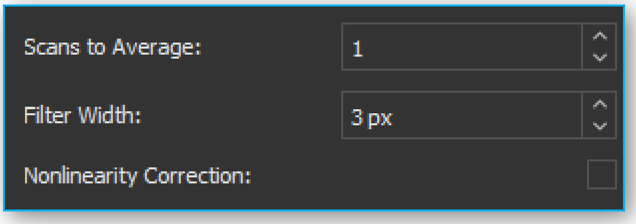


Figure 112: Spectrum Pre-Processing

21.6.4.1 SCANS TO AVERAGE

This is a time-based averaging feature. Time-based averaging increases the amount of spectral information captured by calculating the average outputs of individual pixels over multiple spectral scans. This process takes more time than viewing the results of one scan but produces a higher signal to noise ratio (SNR) and stabilizes fluctuating data. The SNR will increase by the square root of the number of time-based averages. For example, if 100 averages are used, the SNR will increase by a factor of 10, but the measurement will take 100 times as long to acquire than a single scan.

21.6.4.2 FILTER WIDTH

This is a spatial averaging feature, visually smoothing the spectrum. Spatially based averaging will visually smooth the results of a single scan by averaging the values of adjacent pixels together. This process improves the SNR at the expense of optical resolution. Spatial averaging is useful when the spectrum is relatively flat and little variation is expected between adjacent pixels, as the resultant loss of resolution can make it difficult to distinguish sharp spectral features. When using spatial averaging, the SNR will increase by the square root of the number pixels averaged.

For each pixel an average value is determined using neighboring pixel. The entered value defines the width of the filter in pixels on the left and right side of the current pixel (for example: filter width = 3: the average is derived from each data point and its 3 neighboring values to the left and to the right).

21.6.4.3 NONLINEARITY CORRECTION

This feature corrects the non-linearity between the photon stream and the CCD sensor's output signal.

21.6.5 Selecting the X-Axis Unit

You can choose from various units for the x-axis:

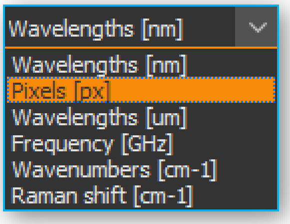


Figure 113: Selecting X-axis Unit

UNIT	MEANING
Wavelengths [nm]	Wavelength of x-axis in nanometers
Pixels [px]	Each pixel value corresponds with a CCD sensor element Starting at 0, ending at n-1 (n = maximum number of sensor elements)
Wavelengths [µm]	Wavelength of x-axis in micrometers
Frequency [GHz]	X-axis frequency in Gigahertz (GHz)
Wave numbers [1/cm]	X-axis in inverse centimeters (1/cm)
Raman shift [Raman]	X-axis in 1/cm – depending on the excitation wavelength of the laser

21.7 Baseline Correction

21.7.1 Overview of baseline correction controls

In the *Baseline Correction* section you will find all settings for real-time baseline correction:

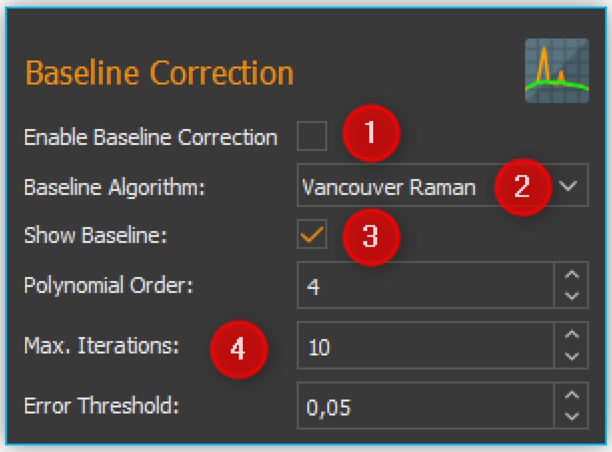


Figure 114: Controls for real-time baseline correction

Baseline variation is a problem that occurs with many types of spectral data. Typically, it is a linear or nonlinear addition to the spectra that results in expected zero measurements reaching a positive value. This can be caused, for example, by the fluorescence during the recording of Raman spectra. A baseline can be described as the slowly varying curve that runs through the lower part of the spectra without the jumps of the peaks.

During baseline correction, a baseline is calculated by the selected algorithm. This is then subtracted from the recorded spectrum to correct its baseline variation.

Use the *Enable Baseline Correction* ❶ check box to activate or deactivate baseline correction. In the *Baseline Algorithm* ❷ selection list, select the algorithm for calculating the baseline. Using the S*how Baseline* ❸ check box, you can select whether the calculated baseline is displayed in the *Live Spectrum Display* or not. Even if baseline correction is switched off, you can display the calculated baseline to visually check its quality (see figure below).

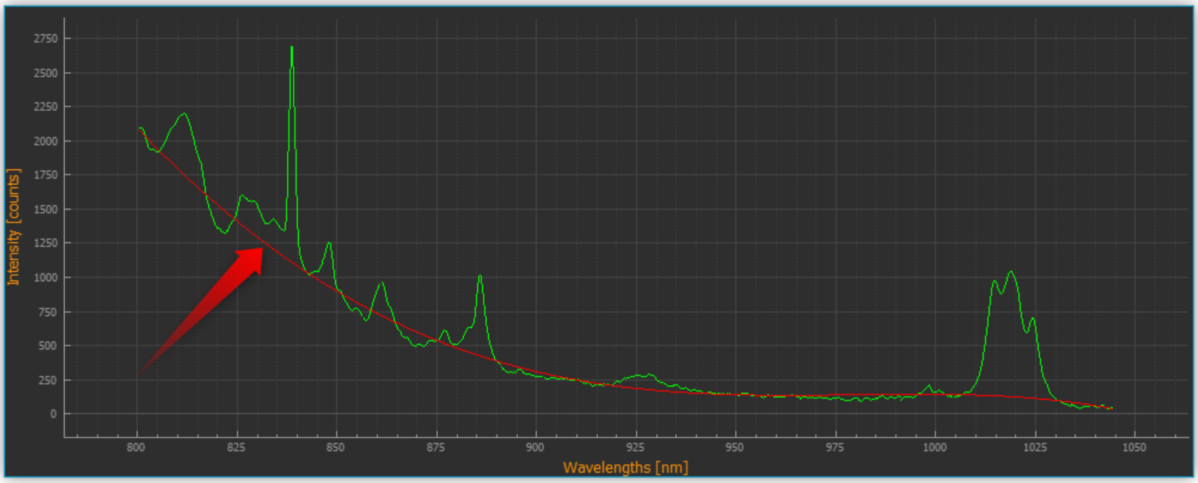


Figure 115: Display of the calculated baseline

In the area below ❹ you will find the settings for the selected baseline algorithm ❷.

21.7.2 Baseline Algorithms

21.7.2.1 VANCOUVER RAMAN ALGORITHM

The Vancouver Raman algorithm has been developed for automated background removal of autofluorescence in the field of biomedical Raman spectroscopy (see Zhao, J., Lui, H., McLean, D. I., & Zeng, H. (2007). Automated Autofluorescence Background Subtraction Algorithm for Biomedical Raman Spectroscopy. Applied Spectroscopy, 61(11), 1225-1232). It is based on a modified multi-polynomial fitting, with the addition of a peak-removal procedure during the first iteration, and a statistical method to account for signal noise effects.

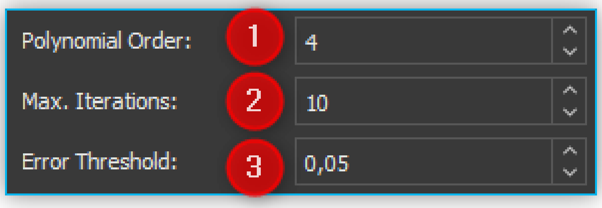


Figure 116: Parameters for Vancouver Raman Algorithm

You can set the following parameters for this algorithm:

- ❶ **Polynomial Order** – Sets the order of the polynomial for the polynomial fit. Based on empirical experience, fourth to sixth-order polynomials provide the best fluorescence approximations.

- **2** **Max. Iterations** – Limits the maximum number of iterations for baseline calculation. This means that even if the error is still above the threshold value, the calculation is terminated after the maximum number of iterations.

- **3** **Error Threshold** – Defines the threshold value for the maximum error as a stop criterion. If the value falls below the threshold value, the calculation is terminated. For 0.95% confidence use 0.05.

21.7.2.2 ROLLING BALL ALGORITHM

Kneen and Annegarn (see M. Kneen and H. Annegarn, Nucl. Instrum. Methods Phys. Res. 82, 59 (1996)) have described a baseline removal algorithm where one imagines a large ball rolling on the underside of the spectrum. The baseline is simply the trace of the topmost point of the ball. The original algorithm was made for X-ray spectra. In three loops the algorithm finds minimum points in local windows, finds maximum points among the minimum points, and smooths by averaging over the maximum points.

Figure 117: Parameters for Rolling Ball Algorithm

You can set the following parameters for this algorithm:

- **1** **Min. / Max. Window** – Size of the local windows to calculate the minimum and maximum points to identify the baseline.

- **2** **Smoothing Window**– Width of local windows for smoothing

21.8 Peak-Detection

21.8.1 Overview of peak detection settings

The peaks of a spectrum can be viewed as its "unique fingerprint". A substance or mixture of substances can be identified by the horizontal position of its peaks. In addition, the intensity of main and side peaks can be used as an indication of the various concentrations of substances within a mixture. The following functions allow you to mark and select peaks.

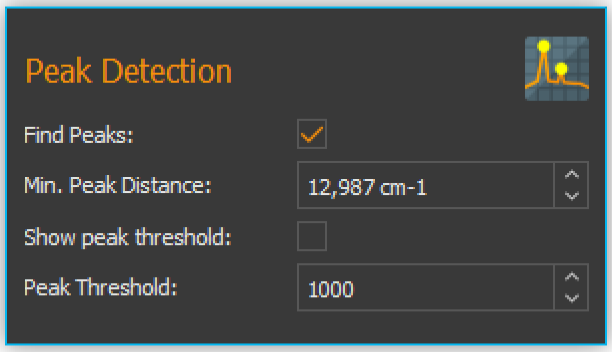


Figure 118: Peak Detection Settings

- **Find Peaks** – Toggles peak detection on or off
- **Min. Peak Distance** – Minimum distance (in x-direction) between two peak markers
- **Show peak threshold** - Shows the peak detection threshold as a yellow broken line in the spectrum display
- **Peak Threshold** – Peak detection threshold – peaks are only marked if they are above the set threshold

TIP. Since the entered minimal peak distance is not transferred to the plot at 100% accuracy, feel free to tweak the value to achieve an optimum result.

TIP. An excessive number of peak markings can affect your PC's computing power. Therefore, only a limited number of peaks is displayed.

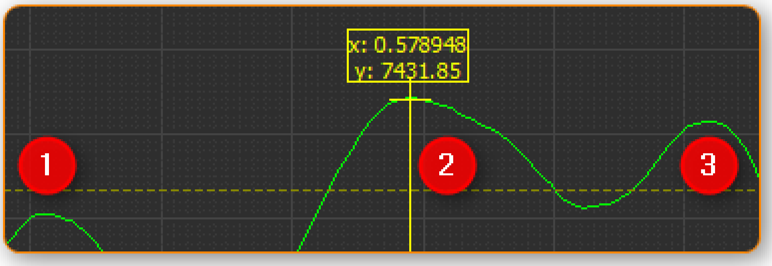


Figure 119: When Peaks are Marked

Application example: a peak is not marked if it is located below the minimum peak intensity line 1. If the peak is located above the line 2, the marking is displayed. The last peak remains unmarked 3, since the distance to its predecessor is smaller than the minimum peak distance.

HINT. If an excessive number of peak markings affects your PC's computing power, reduce the number of peak markings. Increase the minimum distance and minimum peak intensity or smoothen the spectrum.

21.9 Removing Dark Spectrum / Background Spectrum

Dark spectrum is the expected signal level in the absence of light. Background spectrum is the expected signal level in the absence of a sample. By deducting the dark / background spectrum from the currently recorded spectrum, stray light influences and sensor noise of the resulting spectrum are reduced.

Start by setting the parameters in the same way you would in your measurements. Deactivate your light or laser source and record the background spectrum by clicking on the bulb icon.

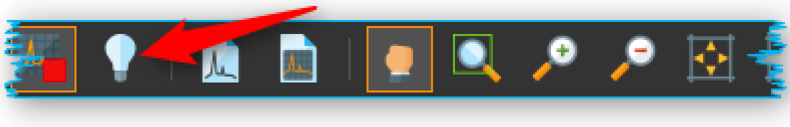


Figure 120: Apply Background Spectrum

Now, the same background spectrum is deducted from each newly recorded spectrum automatically.

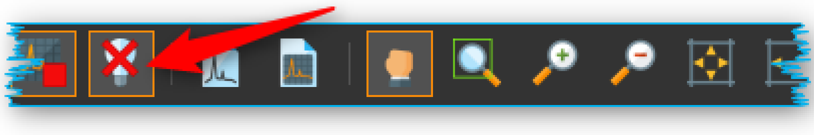
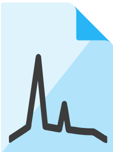


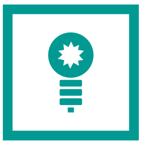
Figure 121: Removing Background Spectrum

If you are not satisfied with the result, click on the bulb icon with the red cross. This removes the background spectrum and lets you repeat the recording of the background spectrum.

21.10 Saving Spectrum Data and Spectrometer Settings



Save your spectrum data as a text file by clicking the respective symbol in the toolbar. You can save the recorded spectrum as a text file (*.txt) or as an Analyze IQ file (*.aiq). The advantage of text files is that they can be opened easily in spreadsheet applications like Excel. The advantage of Analyze IQ files is that they also contain meta data on the recorded spectral data, such as the creation date, units for x and y-axis, the user name and all current spectroscope settings. If you want to use the Analyze IQ add-on for qualitative and quantitative analysis at a later point in time, you should save your spectrums as an Analyze IQ file.



HINT. If you want to use the *Analyze IQ add*-on later for qualitative and quantitative analysis, you should save your spectra as Analyze IQ file (aiq).

A dialog window will open in which you can select the file name and file format.

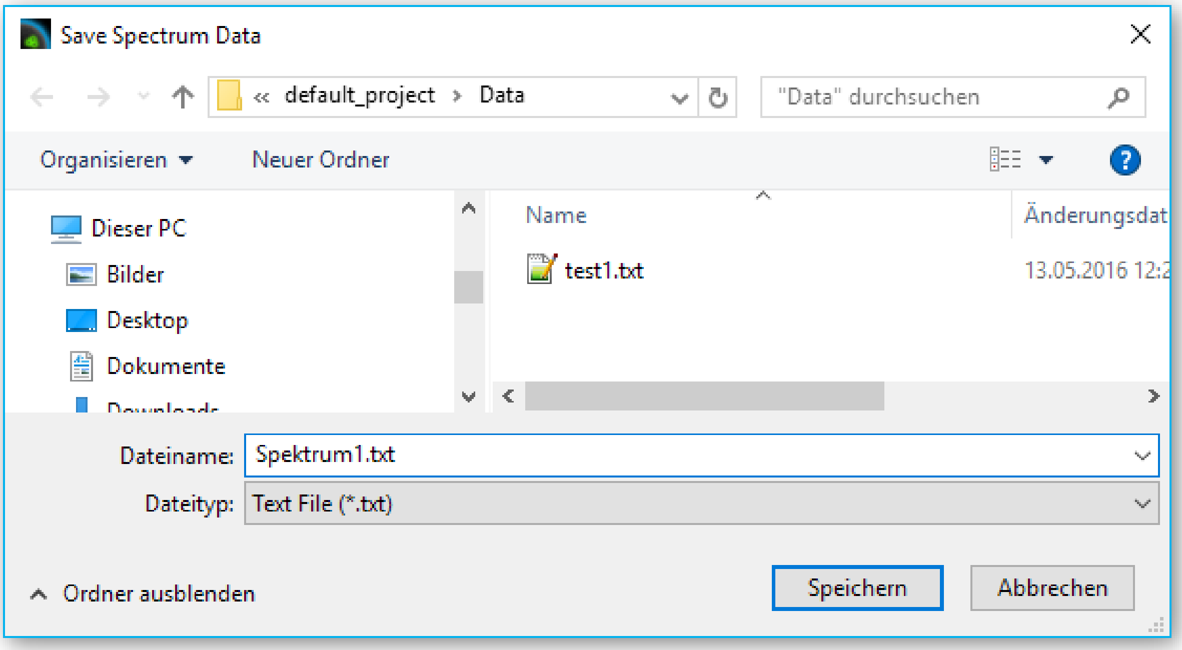


Figure 122: Saving Spectral Data As a Text File

The directory of the current project is used as a target directory. However, you can also select a different directory in the dialog window.

If you have selected the *.aiq format as file format, a dialog for adding metadata is now displayed.

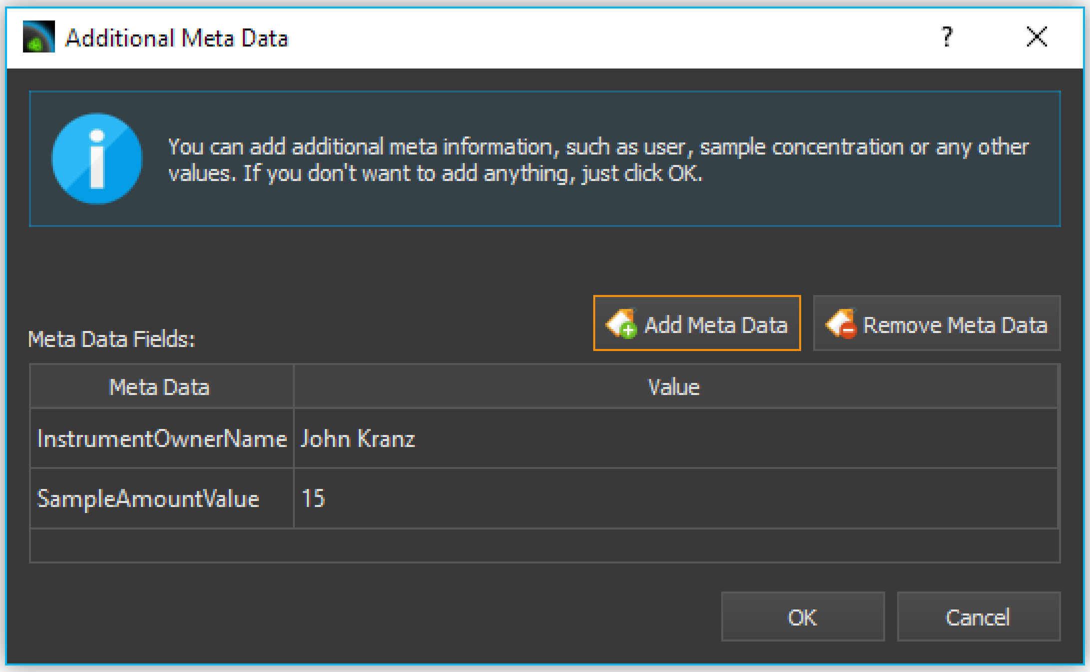


Figure 123: Metadata dialog

Here you can add additional information such as sample name or sample quantity to the *.aiq file by clicking *Add Meta Data*. After clicking *Add Meta Data*, the input window for selecting the metadata name and entering the value appears.

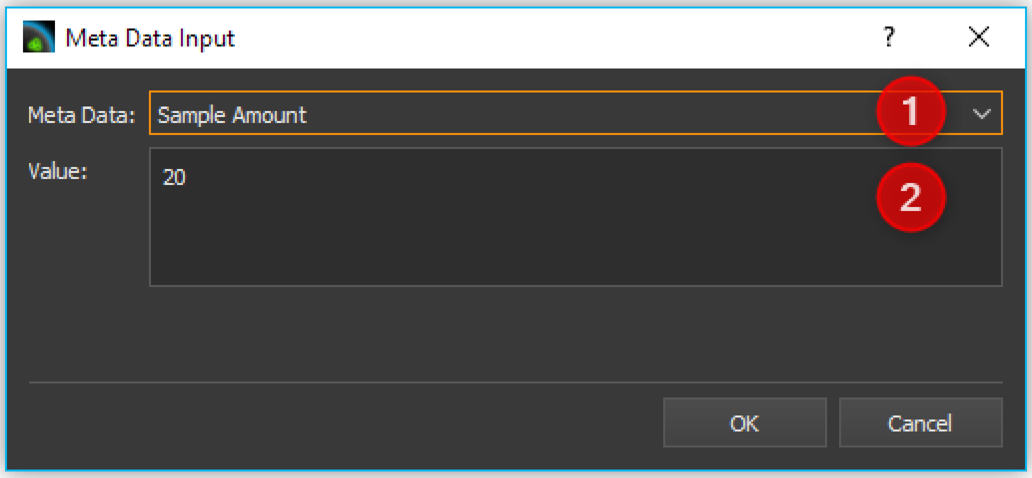


Figure 124: Metadata input dialog

Use the *Meta Data* 1 check box to either select a predefined metadata field or to define a new custom metadata field. You can define a new field by entering a name that is not in the list, for example, Concentration. The predefined metadata fields are defined in the file format for *.aiq files and are partially displayed in the external Analyze IQ software.

Once you have selected the metadata field, enter the value for the metadata in the *Value* 2 field and click *OK* to complete the input.

If you want to delete a metadata field from the list, select it and click the *Remove Meta Data* button. As soon as you click *OK*, the last recorded spectrum together with the current spectrometer settings and the metadata is saved in a file with the extension aiq.

HINT If you save files in AIQ format, you can convert them to CSV format at any time later using the Convert AIQ to CSV file feature.

21.11 Loading Spectrometer Settings

All spectrometer settings can be loaded from previously saved .aiq files. Click on the icon for loading spectrometer settings (*.aiq) and select an *.aiq file.

21.12 Saving a Spectrum Image

By clicking the respective button in the toolbar you can save the current image of the spectrum live display as a PNG file or a vector graphic in the form of a PDF-document. In the file dialog coming up you can select whether you want to save a PNG-image or a PDF-document 1. The image directory of the current project is used as a target directory. However, you can also select a different directory in the dialog.

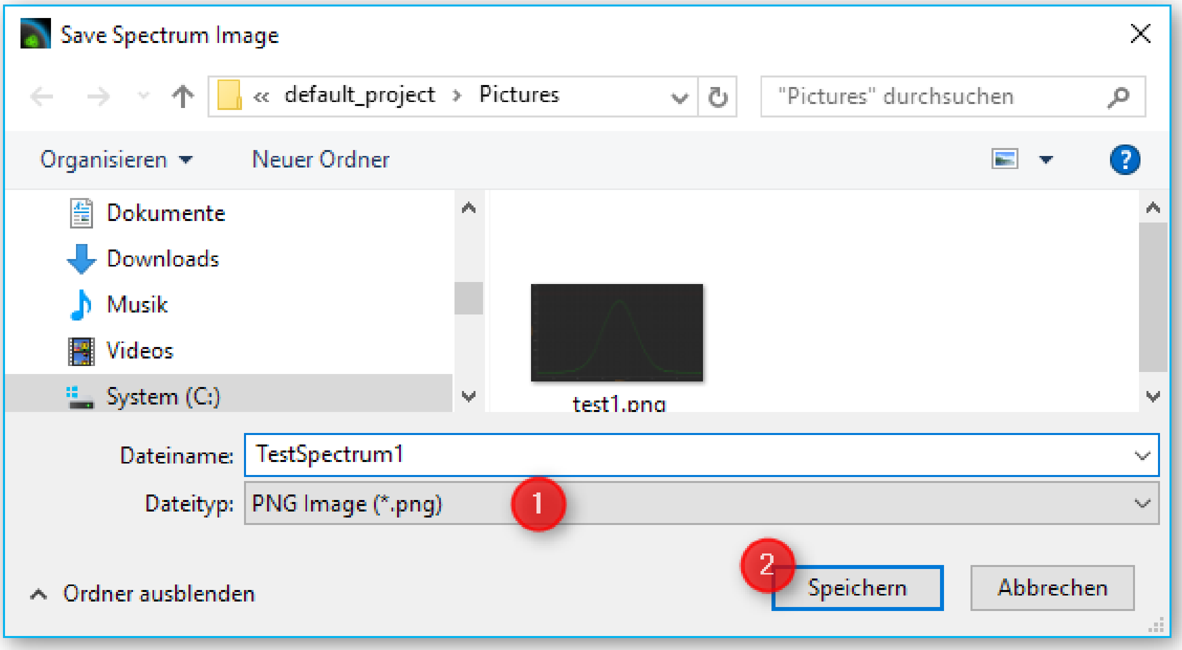
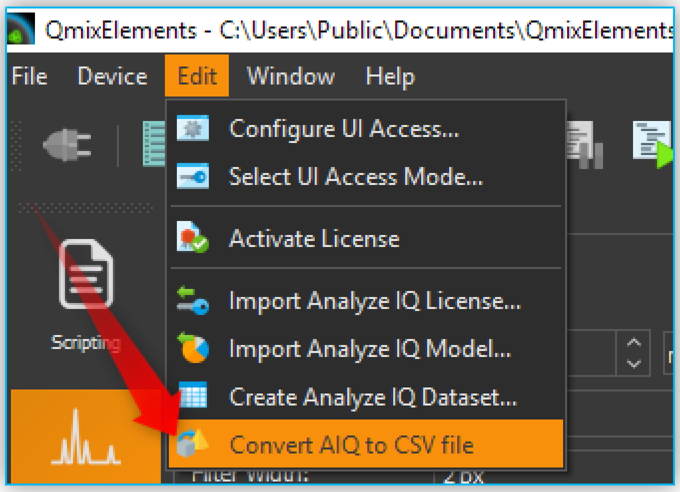


Figure 125: Saving an Image of the current Spectrum Display

Click on the save 2 button to save the image.

21.13 Converting Analyze IQ files to CSV

If you have saved your spectral data in Analyze IQ format (aiq), you can convert them to CVS format at any time later. In the main menu you will find the menu item *Edit → Convert AIQ to CSV file*. Click on this menu item, select the Analyze IQ file and the software saves the file as a CSV file with the same file name and the file extension *.txt.



21.14 Loading and displaying spectral data with the Spectra Viewer

The *Spectra Viewer* provides you with a tool for opening and viewing stored spectral data in Analyze IQ format (**.aiq*) or in CSV format (**.txt*). To open the Spectra Viewer, select *Window* → *Show View* → *Spectroscopy* → *Spectra Viewer* from the main menu. Alternatively, you can open the Spectra Viewer via the *Spectroscopy* button in the sidebar.

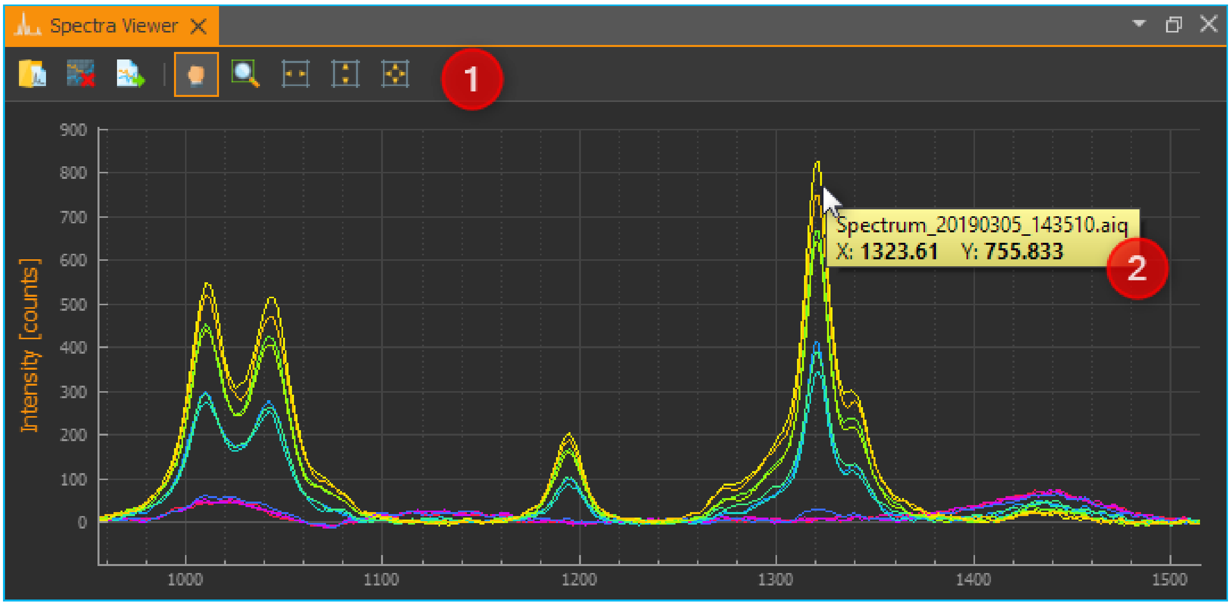


Figure 126: Spectra Viewer for loading and viewing stored spectra

In the toolbar 1 at the top you will find the individual functions of the Spectra Viewer. The functions are essentially the same as in the [Spectrum Live Display](). By right-clicking with the mouse into the Viewer, you can open the context menu with all functions.

Click the *Load Spectra Data* button to open spectrum files. In the file selection dialog that appears, you can select one or multiple files. The selected files will then be displayed in the Spectra Viewer.

Just like in the Spectrum Live Display, you can also move the mouse over a curve to display the value of the curve at the corresponding position 2.

If you want to load additional spectra, simply click the *Load Spectra Data* button again. The new spectra will then be added to the existing spectra in the viewer.

Click the *Clear Viewer* button to delete all curves in the Spectra Viewer.

The *Export Plot Image* button allows you to export the current image in the Spectra Viewer as a PDF or image file.

21.15 Realtime Analysis using Analysis Channels

21.15.1 Introduction

Analysis channels let you analyze recorded spectral data online and integrate the results into QmixElements script programing. This gives you a powerful tool to analyze spectral data online and use the results to control other devices or trigger specific events.

For each analysis an “analog” input channel is created in the I/O channel list (see image below). Like any other analog channel, these input channels can be evaluated and integrated into the QmixElements script system.

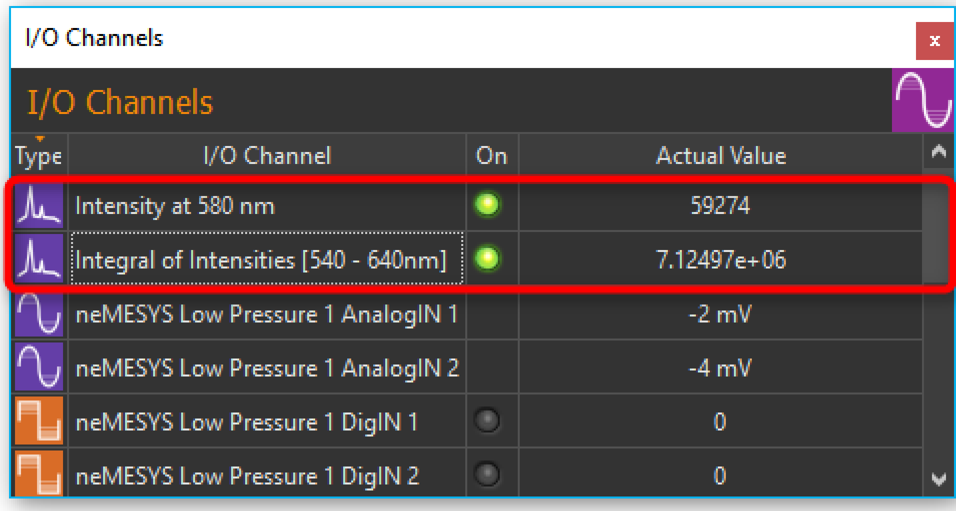
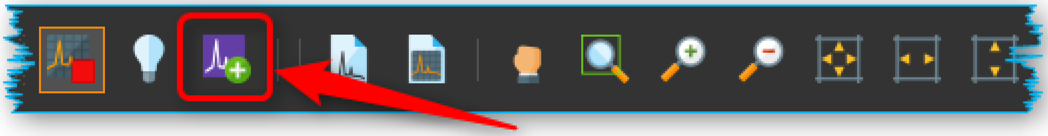


Figure 127: Analysis Channels in the I/O Channel List

To create an analysis channel click on the *Create Spectrum Analysis Channel button in the toolbar.*



A dialog will be displayed that lets you select the analysis function:

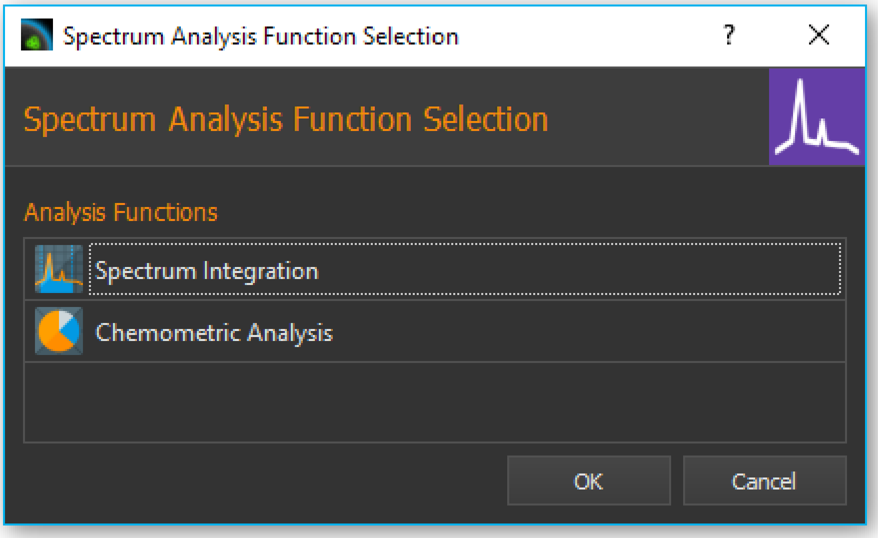


Figure 128: Selection Dialog for the Analysis Function

After you configured the analysis function an analysis channel is inserted into the I/O channel list. You can find details regarding the available analysis functions in the following sections.

TIP. You can create any number of analysis channels for a spectrometer. This allows you to measure the intensity at a certain wavelength, for example, or determine the intensity of the overall signal by integrating all values.

Clicking the green LED in the ON-column (see image below) lets you activate or deactivate the respective online analysis at any time. If you want to change the configuration of an analysis channel or delete the channel, perform a right mouse click on the channel to display the context menu (see image below). Then select the desired action from the context menu.

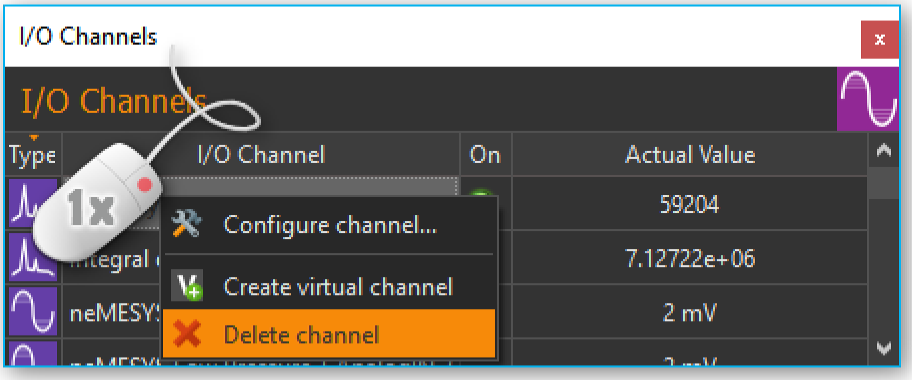


Figure 129: Context Menu: Deleting or Configuring Analysis Channels

21.15.2 Spectrum Integration

The integration function lets you measure the intensity of spectral data within a certain bandwidth or the intensity of the signal at a certain wavelength. Begin by setting the bandwidth in the configuration dialog 1 in which the signal should be integrated. If you want to measure the intensity at a specific wavelength, enter the same values for *Start Range* and *End Range*.

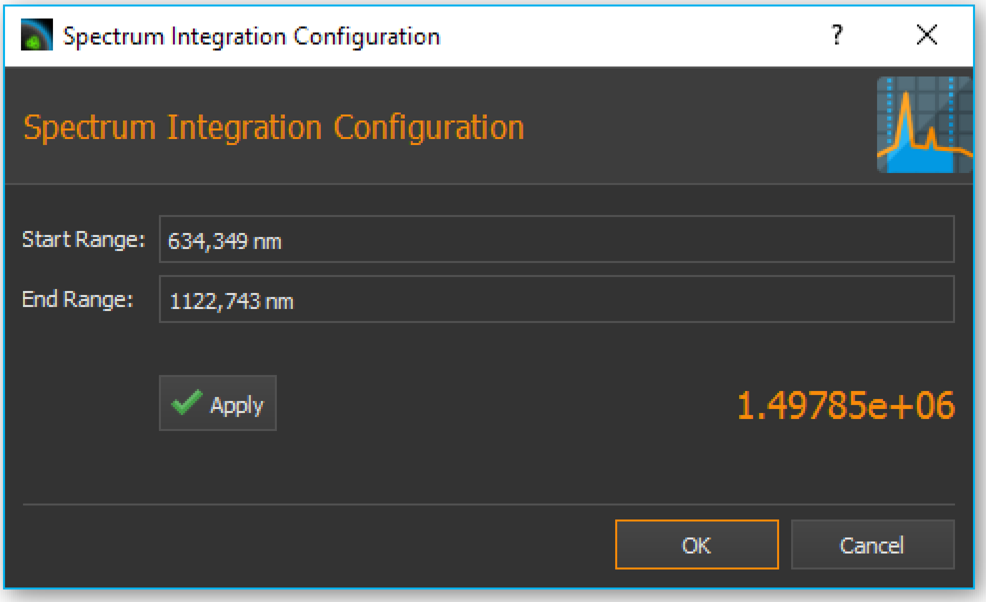


Figure 130: Configuration of Integration Function

Click on the Apply button to accept the values. The software will correct the entered values to the nearest values supported by the spectrometer. In the current analysis display 3 you can see the effects caused by the modified values immediately. If you click on *OK*, the set values are applied and a new analysis channel is created or the currently selected channel updated.

21.15.3 Chemometric Analysis

Chemometric analysis allows the quantitative and qualitative determination of concentrations of individual substances in a mixture of substances. You will need a paid license for the Analyze IQ realtime add-on to be able to use this function.

In this function's configuration dialog you simply have to select an existing Analyze IQ model, which you want to use for spectral analysis (see image below).

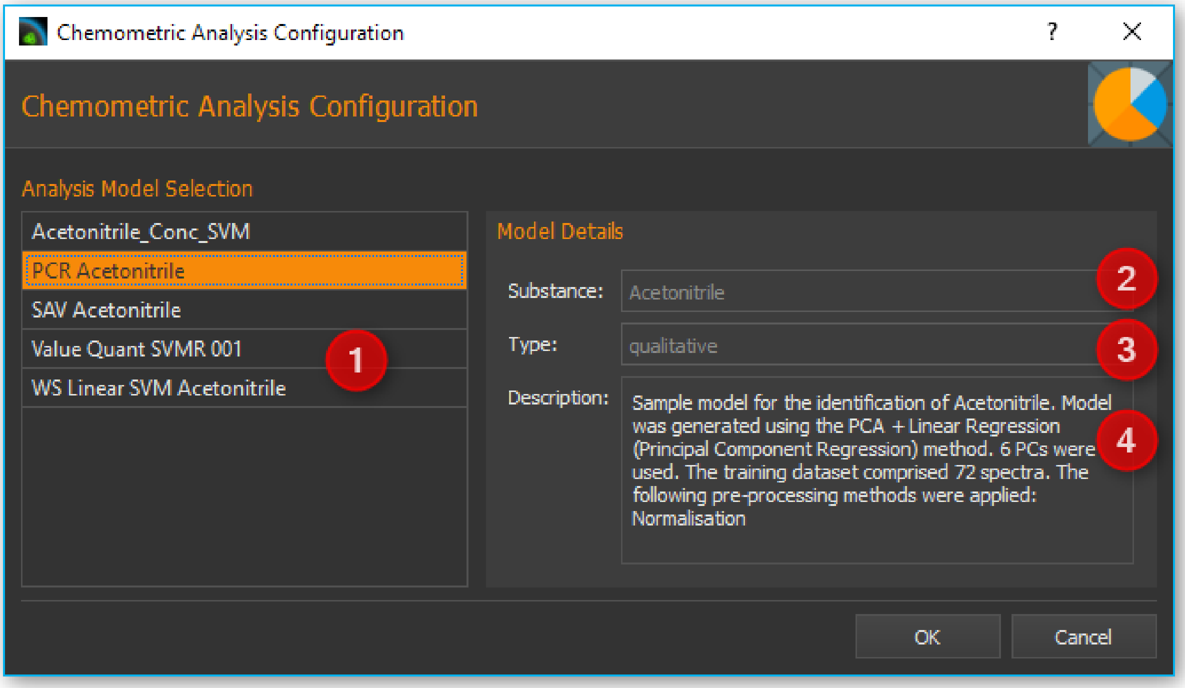


Figure 131: Selection Dialog for the Analyze IQ Model

On the left side you will find a list of all existing models 1. On the right side you will see additional details for the model you selected on the left. For example, you are shown which substance 2 is being analyzed, whether it's a quantitative or qualitative analysis 3 as well as a short description of the selected model 4

Once you selected a model, an analysis channel is created. In case of a quantitative analysis the channel shows you the substance concentration in a range from 0 – 100% (see image below). If you run a qualitative analysis the channel shows you the presence or absence of a substance in a substance mixture using the value 0 (absent) or 1 (present).

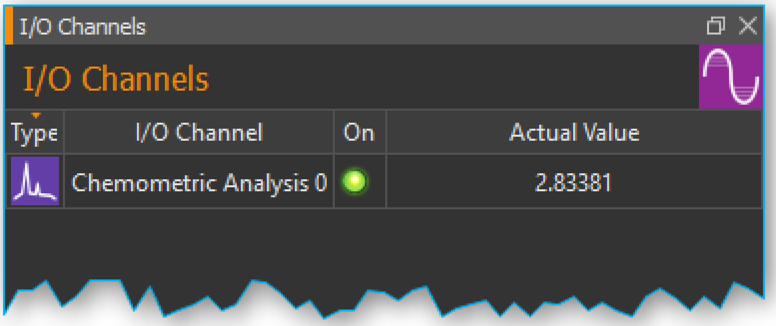


Figure 132: Analysis Channel for quantitative chemometric Analysis

21.16 Spectroscopy Script Functions

21.16.1 Introduction

The spectroscopy plugin contains various script functions for script-controlled spectrum recording.

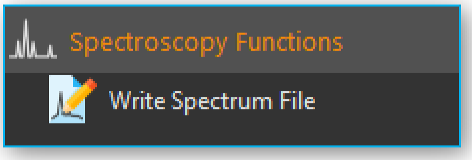


Figure 133: Spectroscopy Script Functions

21.16.2 Saving Spectral Data – *Write Spectrum File*

This function lets you write the current spectrum to a text file (*.txt) or Analyze IQ file (*.aiq).

First, select the spectrometer you want to use in the configuration section 1.

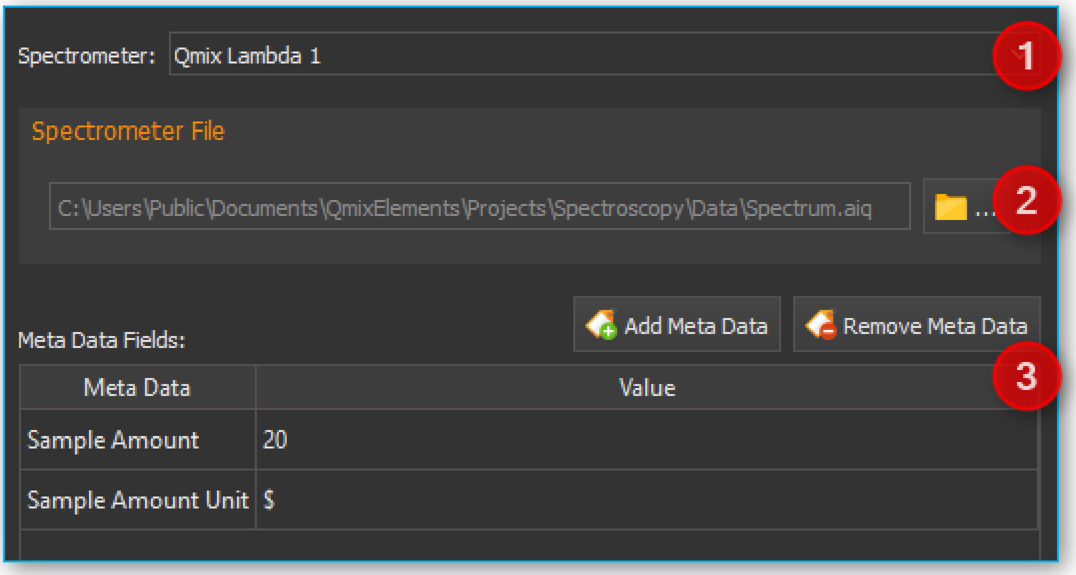


Figure 134: Configuration of Write Spectrum File Function

Then select the file name and file type (*.txt or *.aiq) to save the files under. Click on the file folder icon 2 and select the target directory as well as the file name and file type.

When you start the function the current time stamp will be added to the file name. In this way a file with a new and unique file name is created every time the function in launched. For example, a file called *Spectrum.aiq* will be saved under the following file name:

Spectrum_20161223_135552_545.aiq

In addition to spectral data you can also save metadata regarding the measurement, the sample or the device as part of the file 3.

IMPORTANT. Metadata are saved only if you select the Analyze IQ format (*.aiq).

Click on the *Add Meta Data* button to add a meta data field. Now select the meta data field in the dialog box 1 and then go to the entry field 2 and enter the value you want to assign to the meta data field.

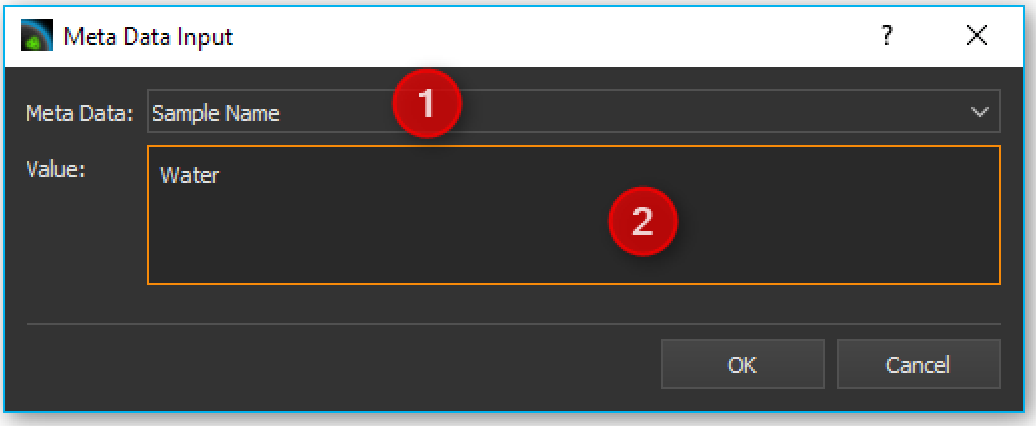


Figure 135: Entering Meta Data

You can also use simple script variables as a value for a meta data field. They will be evaluated when the script runs and the value of the script variables stored in the respective meta data field (see image below).

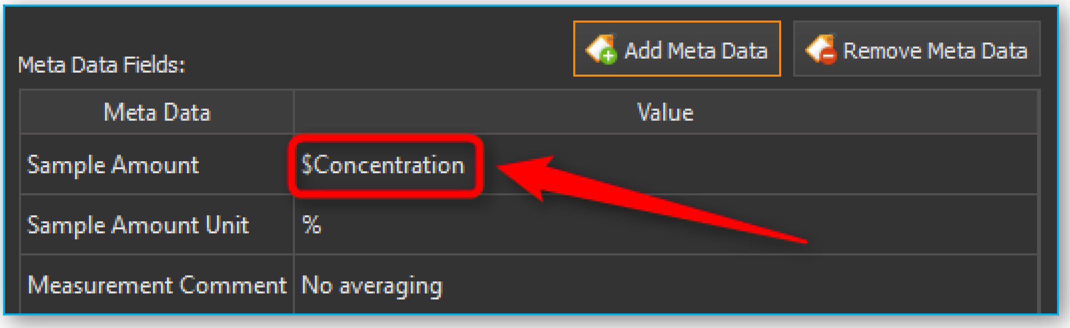


Figure 136: Using Variables in Meta Data

21.17 Analyze IQ Add-on

21.17.1 Introduction

The Analyze IQ add-on allows you to integrate powerful chemometric analysis functions of Analyze IQ into the QmixElements software.

In practical applications the analysis of substance mixtures or mixed liquids leads to spectral data peaks which can overlap. This may lead to non-linear relationships between spectral responses of the various substances.

Analyze IQ offers a new, model-based paradigm for spectral analysis:

- Create a series of known substance mixtures / material compositions
- Record spectral data of these substance mixtures
- Choose from a broad spectrum of chemometric analysis methods to build analytical models that summarize all spectral data in a compact format.
- Unknown mixtures can be analyzed quickly and precisely using these analytical models.

This paradigm offers a number of advantages:

- it separates model building from model use
- expert knowledge for the analysis of spectral data can be packaged in models and passed on to users

Chemometric models can be created conveniently using *Analyze IQ Lab* (see image below).

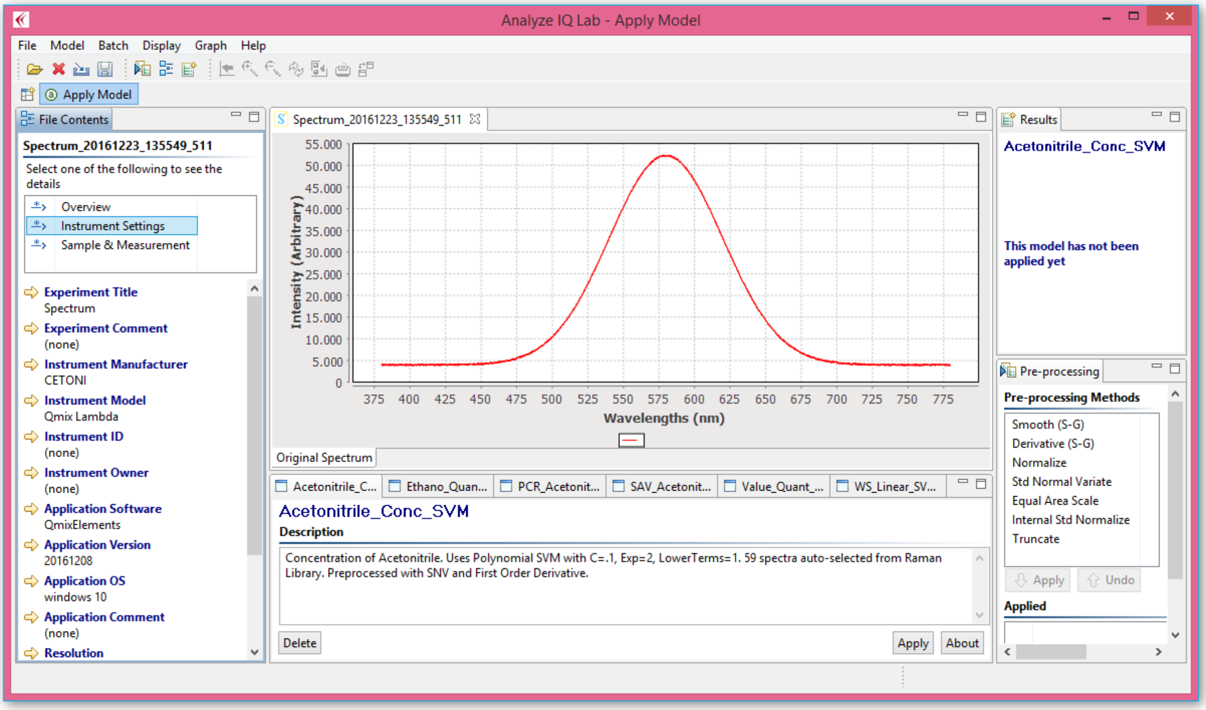


Figure 137: Analyze IQ Lab Software for the Creation of Analysis Models

The models can then also be used by non-experts to turn data into results and decisions quickly. Through an interface to Analyze IQ RealTime this analysis function is integrated into QmixElements. Through analysis channels it can be used for realtime analysis of substance mixtures.

21.17.2 Importing License Data

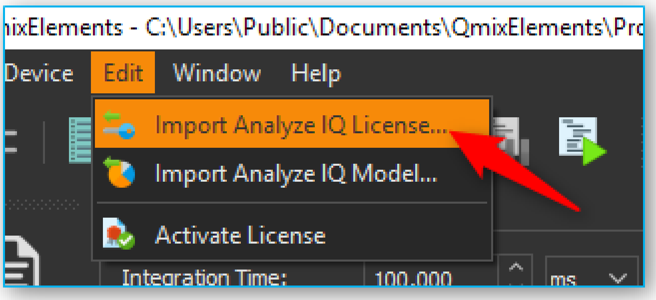


Figure 138: Importing Analyze IQ License

To be able to use the Analyze IQ add-on you will need a valid license file. You will receive this license file (*.ail) from CETONI when you order the Analyze IQ add-on. To import the license file select the menu item *Edit → Import Analyze IQ License*.

After you imported the license file the Analyze IQ RealTime server is launched and the QmixElements software connects itself to the analysis server.

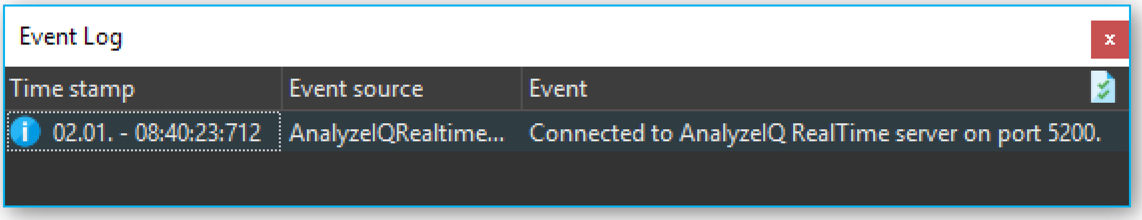


Figure 139: Status Message Following the successful Connection to the Analyze IQ RealTime Server

21.17.3 Generating Spectral Data for Model Building

The QmixElements software can store spectral data in aiq file format, greatly simplifying model building in Analyze IQ software. For the modelling of classification models (substance present yes/no) or quantification models (concentration of a substance in percent), the information whether a certain substance is contained or in which concentration it is contained must be assigned to each spectrum recorded. This task can be solved perfectly by using the metadata fields of the aiq file format.

You can save the concentration of a substance in the metadata both for manual storage and for script-controlled storage of spectral data. Use the same metadata field, e.g. concentration, for all spectra from which you want to create an analysis model. If you create script-controlled mixtures, you can automate the storage of the spectral data together with the corresponding concentration value by using variables (see figure below).

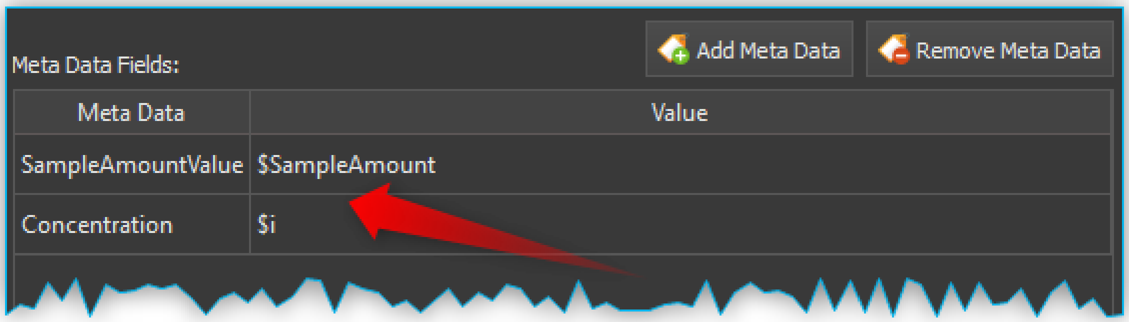


Figure 140: Using variables when saving metadata

All spectra that you want to use for building a certain model should be saved in the same directory.

IMPORTANT. Create a separate directory for each analysis model and then use it to store all spectra that you want to use for building this specific model.

21.17.4 Preparing the Dataset

In the Analyze IQ User Manuel you will find the section *Preparing the Dataset*. The subheading *Import from Multiple Spectrum Files* describes how you can create a model from several individual spectrum files in a folder. Analyze IQ creates a CSV file in which the spectrum is assigned to the concentration value.

	A	B
1	File name	Value
2	Acetonitrile-Methanol (50-50).aiq	50
3	Acetonitrile-Methanol-Water (10-70-20).aiq	10
4	Acetonitrile-Methanol-Water (10-80-10).aiq	10
5	Acetonitrile-Methanol-Water (20-20-60).aiq	20
6	Acetonitrile-Water (50-50).aiq	50
7	Methanol-Water (80-20).aiq	0
8	Methanol-Water (99-1).aiq	0
9		

Figure 141: Example of a spectra list with target values

For each spectrum in the folder, a line is created in the CSV file in which the user manually enters the concentration value in percent (quantification model) or the presence of a substance (classification model) as "Yes" or "No" selection (see figure).

This time-consuming process can be done in QmixElements with just a few mouse clicks. To create a data set, select *Edit* → *Create Analyze IQ Dataset* from the main menu.

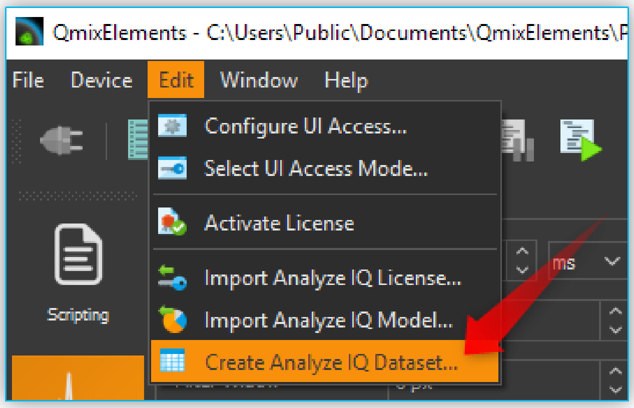


Figure 142: Open the dataset preparation dialog

A dialog is now displayed for preparation of the data set for an analysis model. In the *Folder Containing Spectra* 1 field, select the folder containing the spectra. In the *Value Metadata* 2 field, enter or select the name of the metadata field in which you wrote the concentration value when saving the spectra.

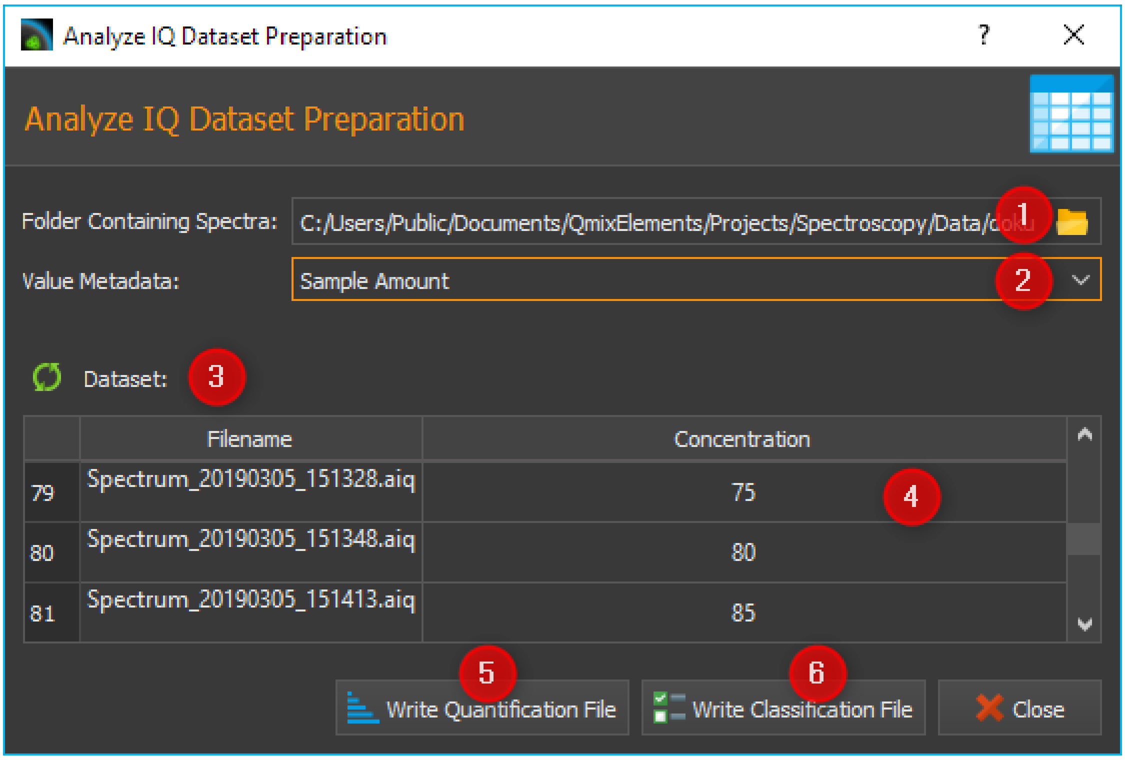


Figure 143: Dialog for the configuration of an Analyze IQ data set for model building

Then click on the *Refresh* button 3 with the green arrow symbol to automatically read in the concentration values from all spectra and to update the column *Concentration* 4 with these values.

HINT. If the expected values are not read in but all fields in the *Concentration* column remain at the value 0, then please check whether you have selected the correct metadata field or whether you have a typing error in the name of the metadata field.

Whenever you change the entry in the metadata field, you should re-read the values using the *Refresh* button.

Then click the *Write Quantification File* 5 button if you want to create a quantification model, or click the *Write Classification File* 6 button to create a classification model. When you click one of the buttons, a file dialog opens for you to enter the file name. The dialog opens in the folder containing the spectra. Enter the file name to save the dataset as a CSV file.

The CSV file for a classification model contains a line with two entries for each spectral file. The entry in the first column contains the file name of the spectral file. The second column contains the text “Yes” or “No”, indicating the presence or absence of the target substance in the sample corresponding to that row.

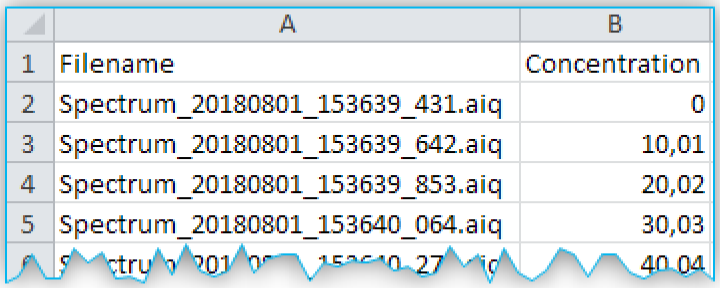
	A	B
1	Filename	Target Present
2	Spectrum_20180801_153639_431.aiq	No
3	Spectrum_20180801_153639_642.aiq	Yes
4	Spectrum_20180801_153639_853.aiq	Yes
5	Spectrum_20180801_153640_064.aiq	Yes
6	[illegible]	[illegible]

Figure 144: Example CSV file for classification model

The following rules are used to generate the entries:

- Value from metadata field = 0 → No
- Value from metadata field ≠ 0 → Yes

You should take this into account when saving the metadata. If you use a quantification model, the concentration values are written directly to the CSV file. That means, the second column contains a value from 0 to 100 (inclusive), which indicates the percentage concentration of the target substance in the sample corresponding to that row (see figure below).



	A	B
1	Filename	Concentration
2	Spectrum_20180801_153639_431.aiq	0
3	Spectrum_20180801_153639_642.aiq	10,01
4	Spectrum_20180801_153639_853.aiq	20,02
5	Spectrum_20180801_153640_064.aiq	30,03

Figure 145: Example CSV file for quantification model

21.17.5 Import spectra into Analyze IQ

When you create a new analysis model in Analyze IQ, all you need to do is to import the recorded spectra and the created CSV file, and you are ready to start modeling immediately:

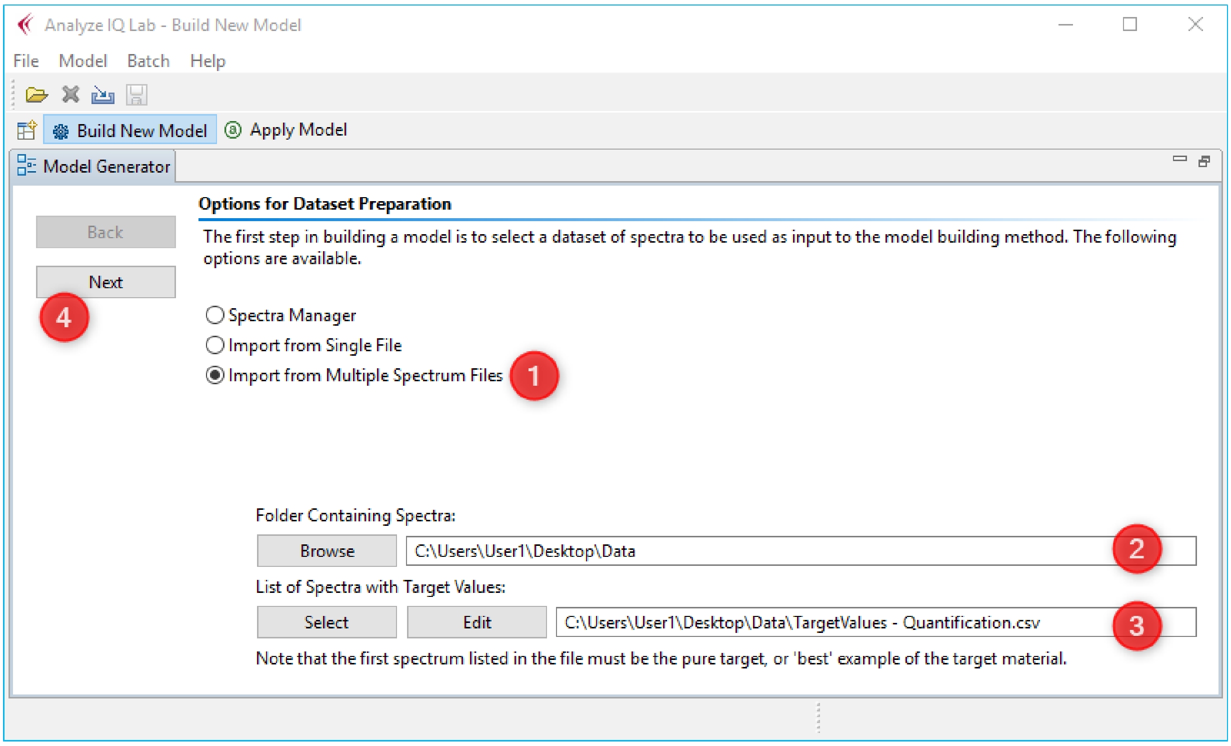


Figure 146: Analyze IQ Creation of an analysis model from imported spectra

To do this, click on the item Import from *Multiple Spectrum File* 1 in the Analyze IQ software. Then select the folder containing the spectra 2 and select the CSV dataset file 3 that you have previously created with the QmixElements software. Then click *Next* 4 to import all data.

After the data import, you can start modeling, which is described in detail in the Analyze IQ User Manual (see figure below).

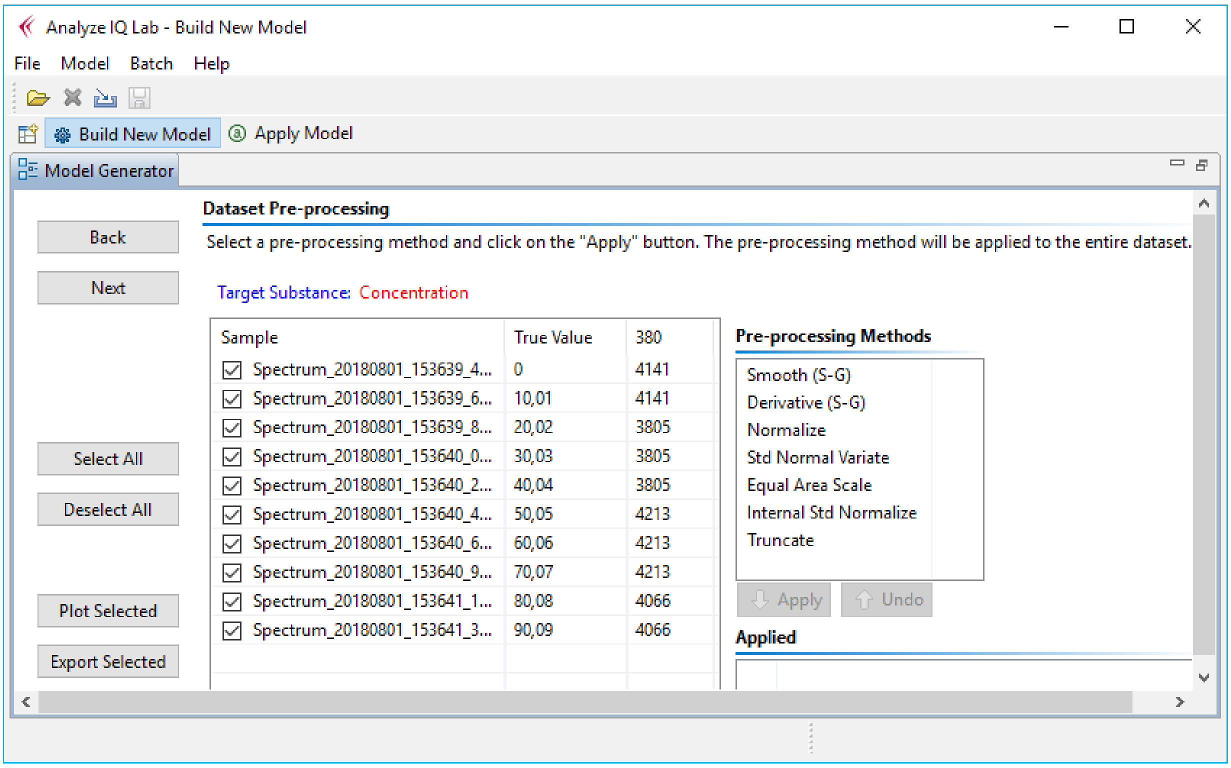


Figure 147: Example CSV file for quantification model

At the end of modeling, you will get a complete analysis model (*.aiqm), which you can then use again in the QmixElements software for automatic analyses. The following section explains how this works.

21.17.6 Importing Analysis Models

You will need completed analysis models to perform chemometric analyses. These models (*.aiqm files) must be imported through QmixElements in order to be added to the *Analyze IQ RealTime server*. To import these model files go to *Edit → Import Analyze IQ Model*.

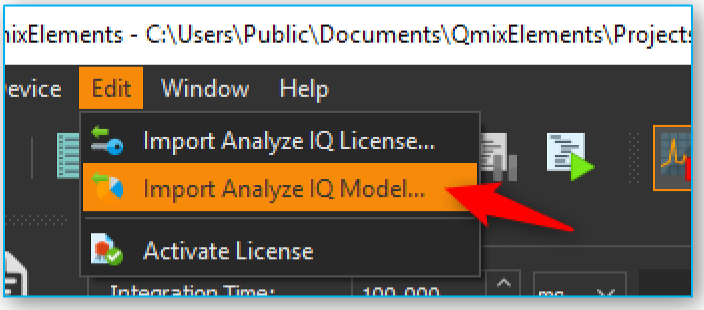


Figure 148: Importing Analyze IQ Models

After the importing process the *Analyze IQ RealTime Server* is restarted and the imported model is ready for your analyses.

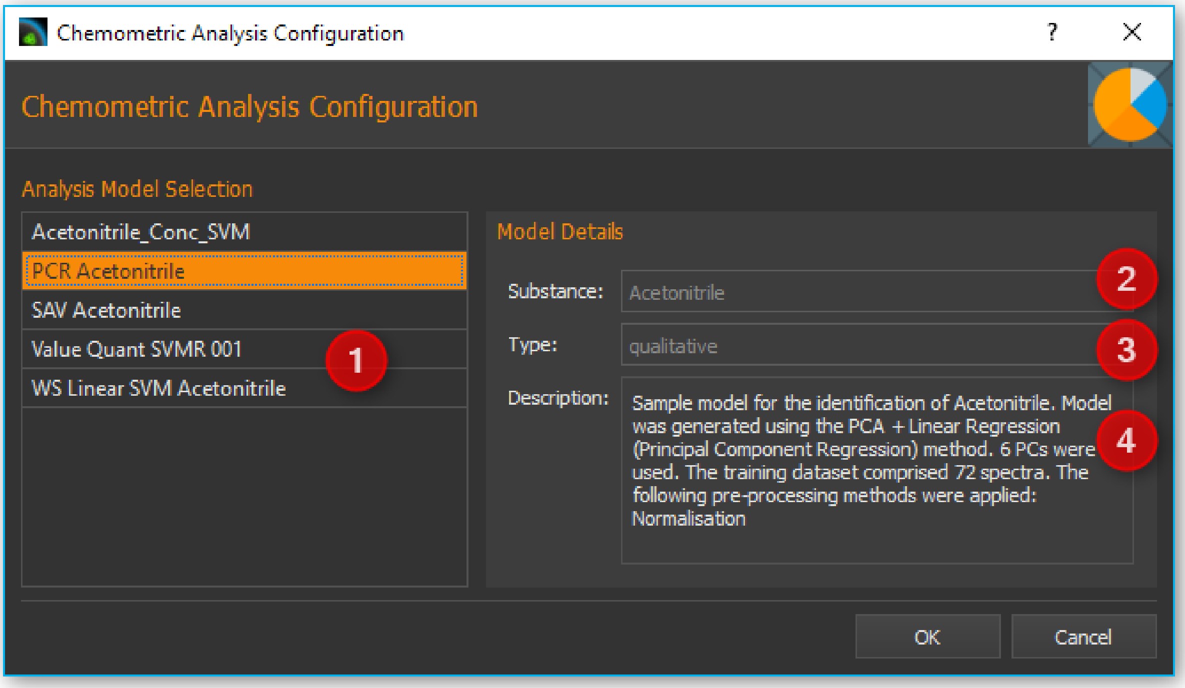


Figure 149: Selection Dialog for Analyze IQ Model

TIP. To create analysis models yourself you will need the Analyze IQ Lab software from Analyze IQ, which is not part of the Analyze IQ add-on.

IMPORTANT. The analysis models need to have been created for the respective spectrometer, i.e. the analysis models need to have been created with spectral data recorded by the same spectrometer or type of spectrometer as the spectrometer used for the current spectral data recording.

22 DAQ Add-on

22.1 Installation

The DAQ Add-on is not part of the standard installation package and must be installed as add-on. The versions of the DAQ Add-on and the installed CETONI Elements software should match. For example, if you want to install the DAQ Add-on version 20190721, you need to have CETONI Elements version 20190721 installed.

IMPORTANT. The version number of CETONI Elements and the DAQ Add-on should match.

Please close all other programs before launching the installation.

IMPORTANT. Install the DAQ plugin and the device driver before connecting your DAQ devices to your PC through USB for the first time.

Launch *CETONI_Elements_DAQ_Setup_64bit.exe* to begin the installation. The installation assistant will guide you through the installation of the software and hardware drivers.

IMPORTANT. If you want to install the hardware drivers on a windows system, you need to be logged in as an administrator.

22.2Introduction

The DAQ Plugin serves data logging and evaluation at high sample rates (> 1000 samples per second). Devices with a high data acquisition rate are supported (i.e. . National Instruments USB 600x multi-function I/O devices). But the DAQ Plugin supports usage of devices with low data acquisition rate, too. Data are written into a process graph and into a CSV-log-file simultaneously. There is no separate configuration required. Use the push button *DAQ* 1 (see figure below) in the sidebar to display the process data graphs.

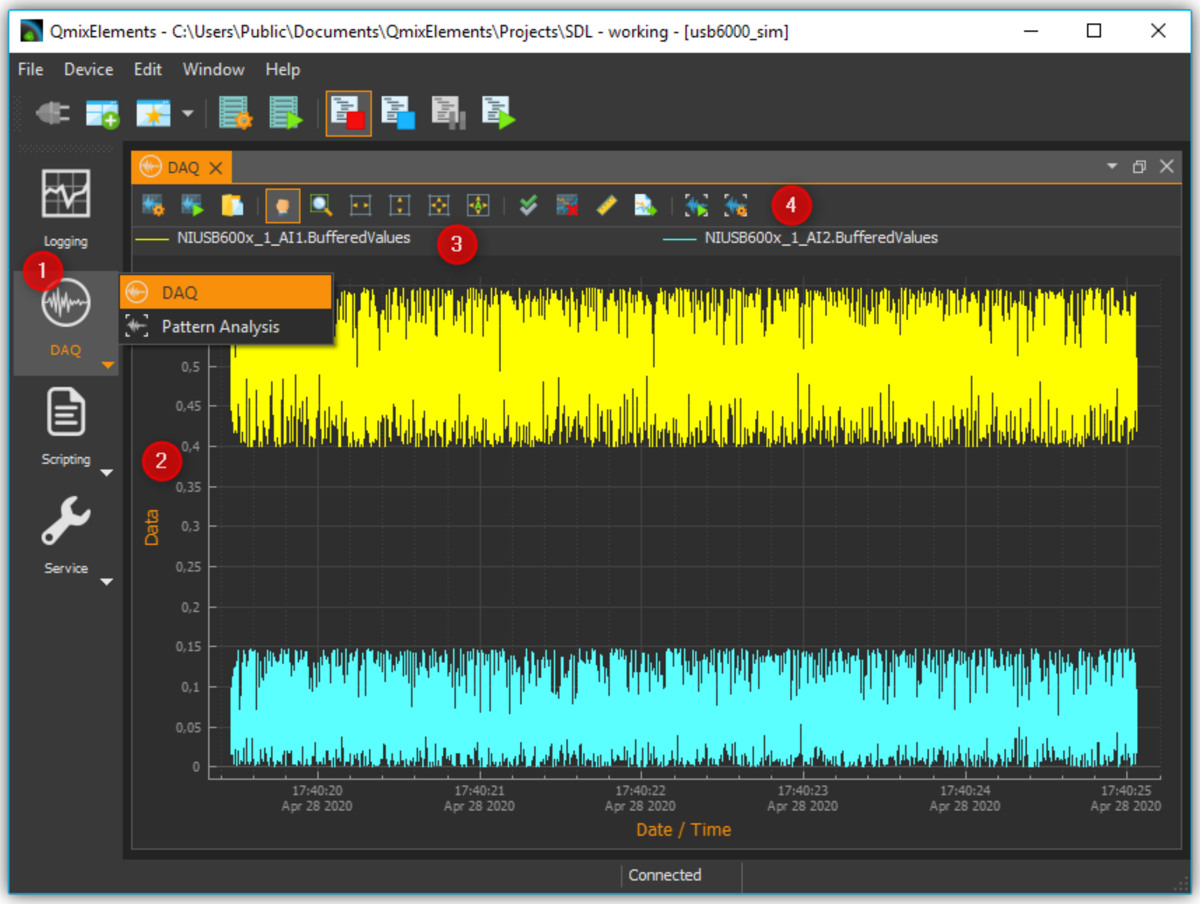


Figure 265: Overview DAQ

The main elements of the process data graph area are as follows:

1 DAQ *selector buttons* – Click this to show the process data graphs and the workbench for the results analysis of pattern recognition. The former is explained in detail in section [1.7.Diagram Navigation & Use](), the latter in section 1.10.4.Result Analysis for Pattern Recognition

2 *Graph canvas* – This displays the curves of all process data sets that are being recorded.

3 *Legend* – The legend lists all data sets that are displayed with their respective colors. Here you can toggle between whether or not a curve is being displayed.

4 *Toolbar* – Here you find buttons to configure the data logging, to start and stop the recording and to navigate the display.

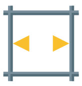
22.3 Toolbar

Opens the configuration dialog of the graphic process data logger.

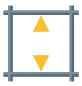
Toggles the recording of process data.

Panning tool to move the currently displayed section of the graph.

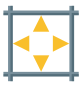
Draws a zoom-in frame to enlarge a desired area of the graph.



Auto-scales the X axis to fit all process data on the screen.



Auto-scales the Y axis to fit all process data on the screen.



Auto-scales both X and Y axes to fit all process data on the screen.

Activates auto-scaling: during a recording, both x- and y-axes are continuously rescaled to fit all process data on the screen.

Show all curves. If curves are hidden, they are displayed again.

Clear plot data. Deletes all data from the diagram.

Toggle X-axis scale. This switches the scaling of the X-axis between absolute date/time stamp and relative time in seconds and milliseconds since the start of recording.

Export plot image. Exports an image of the currently displayed section.

22.4 Configuration Dialog

22.4.1 Overview

Click on the button *Configure data acquisition* in the toolbar to open the configuration dialog.

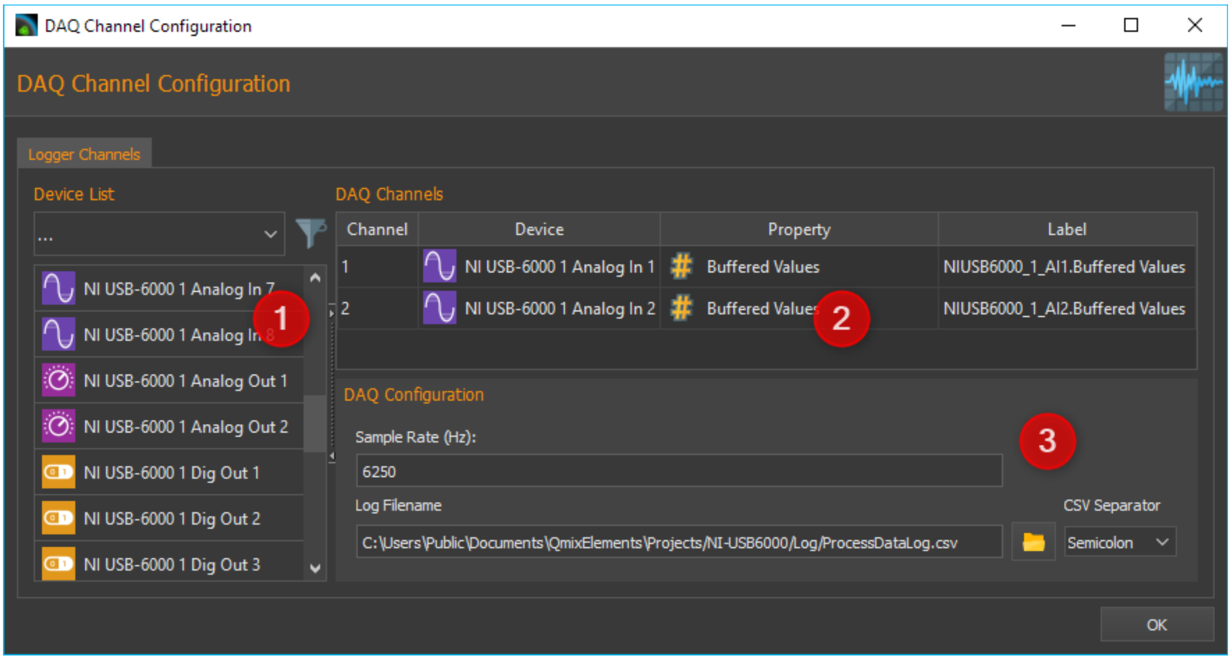


Figure 150: DAQ Configuration Dialog

The configuration dialog consists of the following sections:

1 *Device List* – shows all devices that return data that may be logged. The filter selection box allows to pre-select a specific device type (e.g., I/O channels).

2 *DAQ Channels* – lists all data series or curves that are being recorded and displayed in the diagram.

3 DAQ *Configuration* – in this section you can adapt the sample rate and enter the path of the log file that is written simultaneously to the process data graph.

22.4.2 DAQ Channels Table

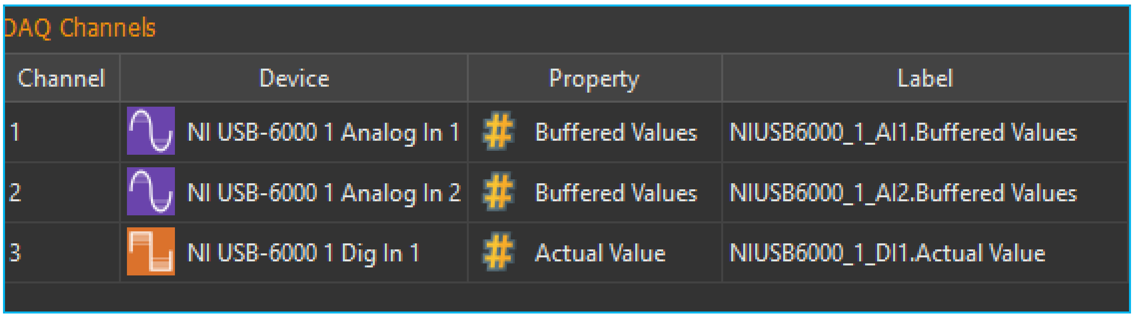


Figure 266: DAQ Logger Channels

The table *DAQ Channels* shows the configured data acquisition channels in tabular form. Each row in the table corresponds exactly to one curve in the graphical plot. The following columns are available:

- *Channel* – returns the channel number.
- *Device* – lists the device name for each respective channel including its icon.
- *Property* – shows the property of the respective device that is to be recorded. The data type is identified via a data-type specific icon.

 Numerical value

 Boolean value

Text value

- *Label* – allows you to define a user-specific name for each channel. This label will also be used in the legend of the plotted graph.

To add and configure channels, please proceed as detailed in the following sections.

22.5 Configure data acquistion

22.5.1 Step 1 – Adding Channels

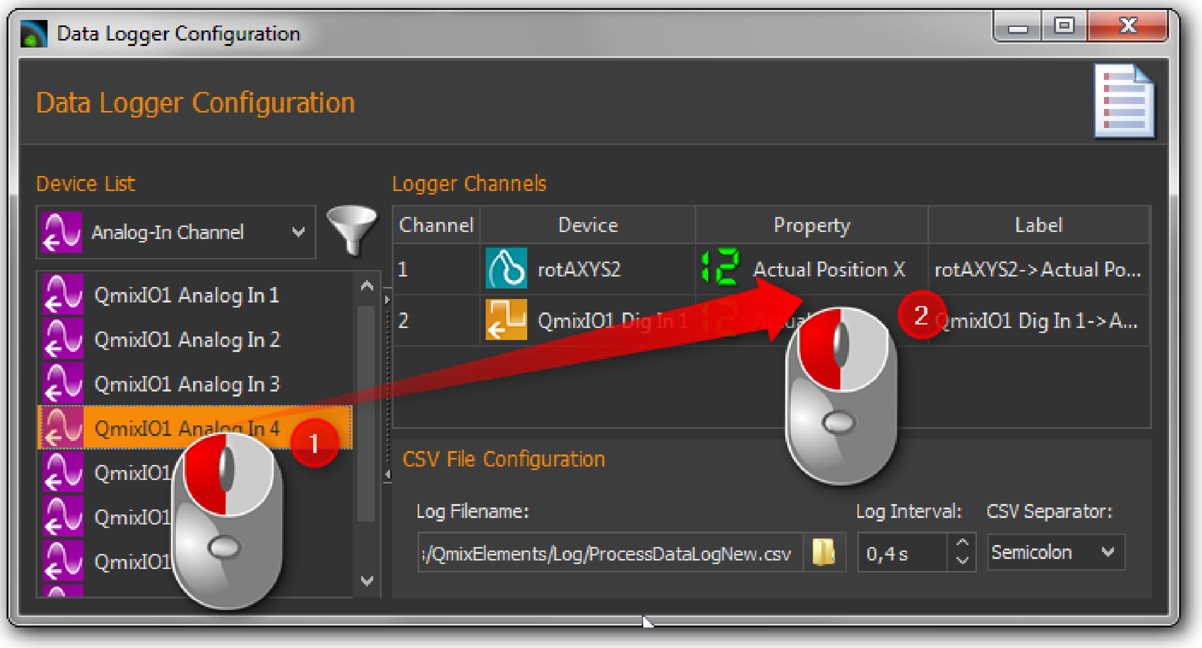


Figure 151: Adding a channel to the channel list

To add a channel you first have to add the relevant device to the *Device List* of the *Plot Logger Configuration*. To do this, move the relevant item from the device list to the *Plot Curves* table using Drag-&-Drop. The new channel will be added at the position where you release the mouse button (see figure below).

HINT. To simplify the device selection process, the device list may be filtered for a relevant device type.

22.5.2 Step 2 – Selecting the Device Property

Select the device property that you want to record by double clicking into the *Property* field of the respective channel from *DAQ Channels* table. This will display a drop-down list with all available device properties from which you may select the desired item (see figure below).

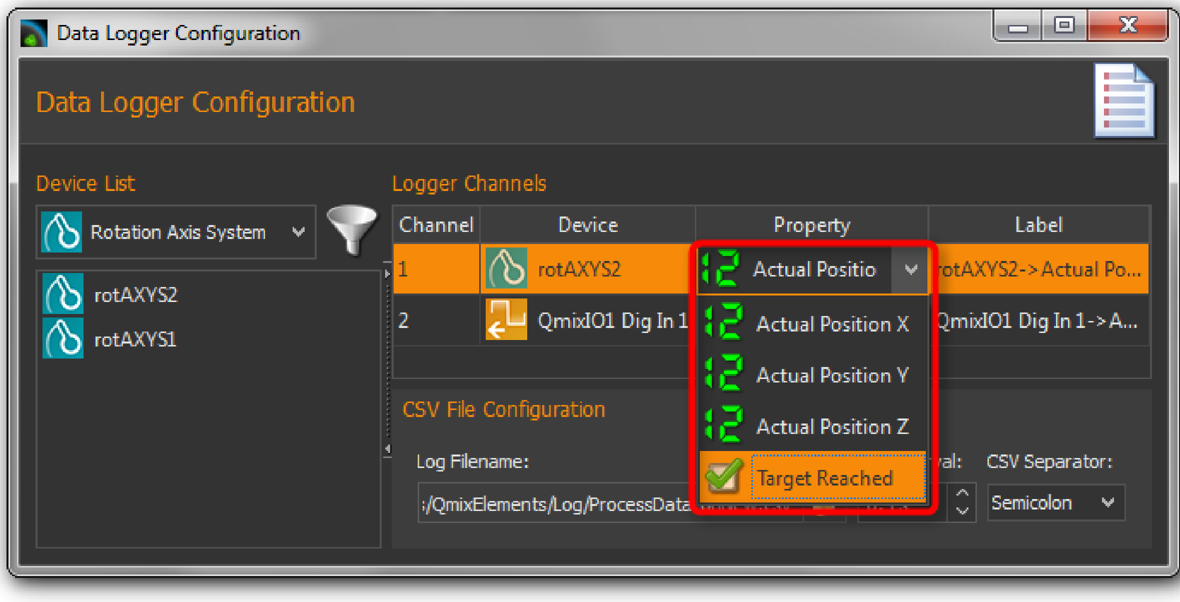


Figure 152: Selecting the device property to be recorded

22.5.3 Step 3 – Changing the Channel Label

In the Label column you can assign a label to each channel. This label will later be displayed in the legend of the graph as the curve label. To change the label, double-click in the table cell (see figure below) and then enter the new label.

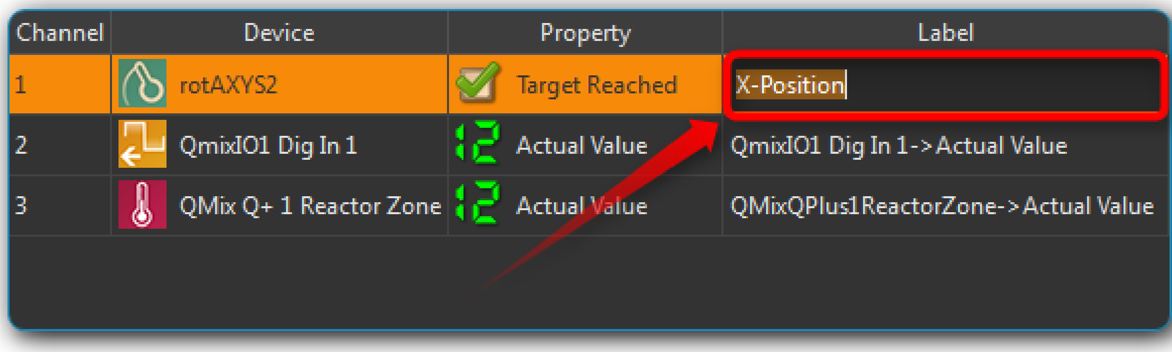
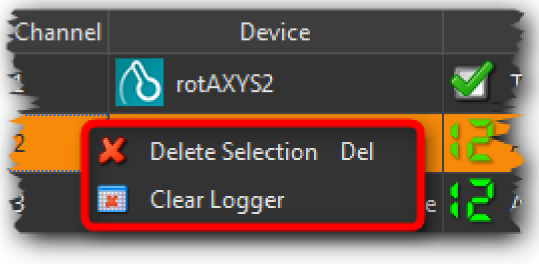


Figure 153: Changing the channel label

IMPORTANT. When a different device property is being selected, a new channel label will be assigned automatically. Therefore, the channel label should be changed after the device property has been selected.

22.5.4 Deleting Channels

In order to delete one or multiple channels from the *DAQ Channels* list, first you have to mark the respective channels using the computer mouse. Now you may use either the keyboard's *Delete* key or select the relevant item (*Delete Selection*) from the right-click context menu.



You may also delete the entire list in a single step by using the *Clear Logger* item of the context menu.

22.5.5 Step 4 – Defining the Sample Rate

In the field *Sample Rate (Hz)* you can define the frequency measurement values are written into the plot and the log file at. The maximum sample rate depends on the measurement frequency of the hardware in use.

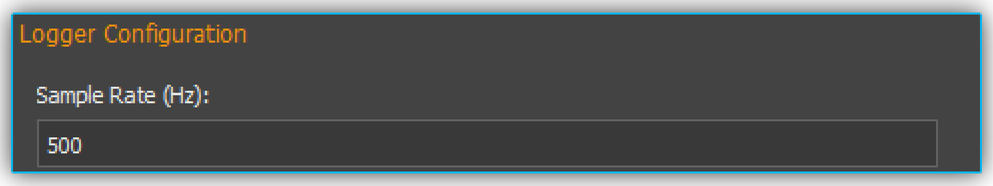


Figure 154: Configuring the sample rate

IMPORTANT. Always select a sample rate that is only as high as necessary. High sample rates produce a lot of data. Drawing large amounts of data requires more computer power and may slow down the usability of the application.

IMPORTANT. Use a low sample rate to record data over several days or use the graphical logger instead.

The configuration will be saved and reloaded automatically upon exiting the *Logger Configuration* dialog.

22.5.6 Step 5 – Configuring the Log File

In this section you can enter the log file 1 the measurement values are written into in addition to the graphical plot. Furthermore you can define the separator character the measurement values of the different channels are separated from each other with 2.

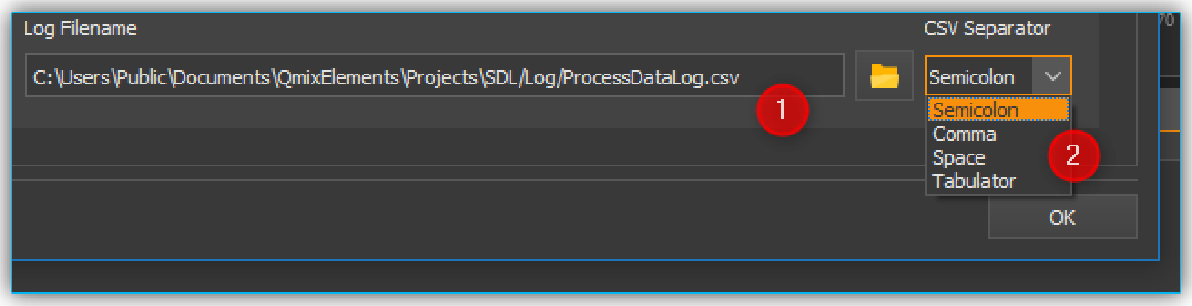


Figure 155: Configuring the log file

22.6 Start/Stop Data Logging

The data logging process may be started/stopped via the relevant button in the toolbar.

22.7 Diagram Navigation & Use

22.7.1 Overview

The DAQ plug-in offers a number of possibilities to customize the way data are displayed. This includes resizing parts of a curve and showing or hiding individual curves. The diagram consists of a plot area 1 plus both an X-axis (time) 3 and a Y-axis (process data) 4.

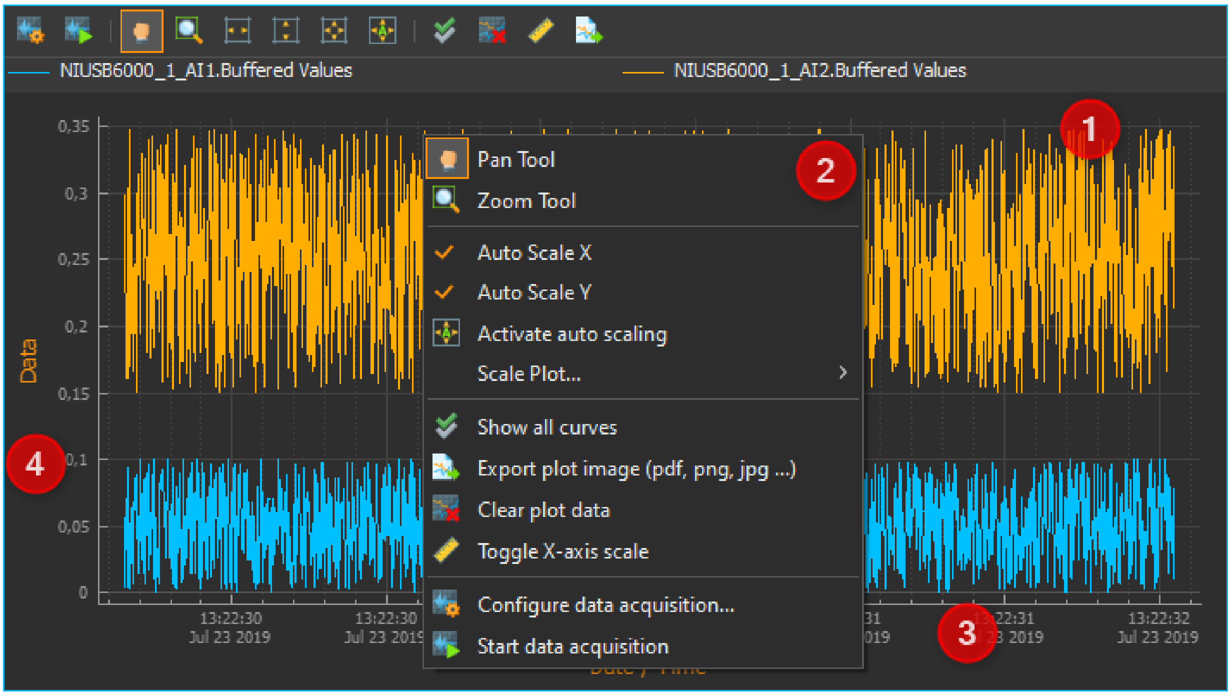


Figure 156: The process-data diagram section

The time axis shows date and time as absolute values. The process-data axis shows the respective measurement data; it is without units as it potentially represents a variety of very different values and measurement units.

A right mouse click within the plot area will open a context menu 2 with a number of additional functions.

22.7.2 Changing the Displayed Section

The *Pan Tool* provides you with a simple way to move the displayed section of the plot area. It may be activated via its toolbar button and the displayed section may then be moved around using the mouse whilst keeping the left button pressed.

IMPORTANT. Panning of the displayed plot section will deactivate the auto-scaling of the diagram axes.

22.7.3 Display Curve Values

When the *Pan Tool* is active, you can move the mouse pointer over a curve to display the value at that specific position.

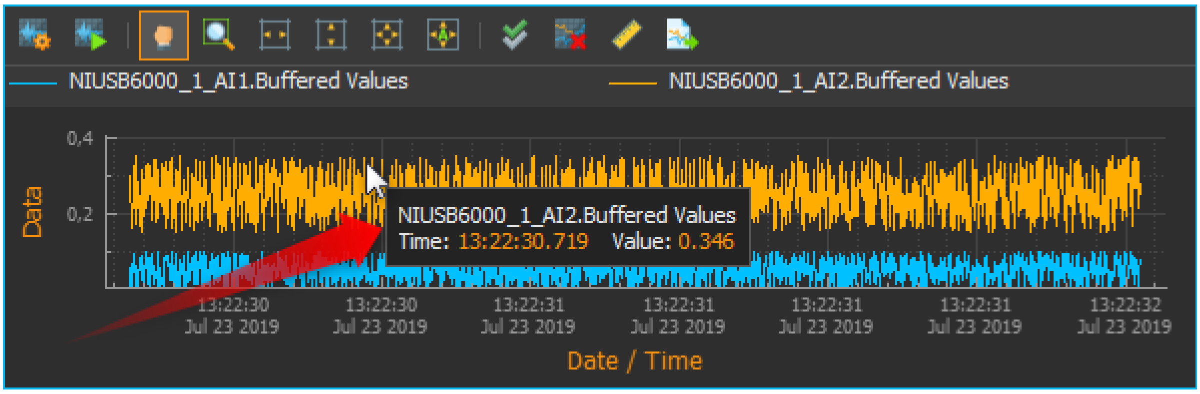
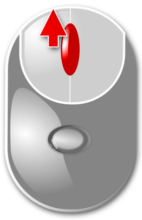


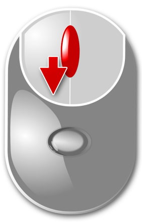
Figure 157: Display Curve Values

22.7.4 Zooming via the Mouse Wheel

Turning the mouse wheel whilst the pointer is within the plot area will allow you to adjust the displayed section of a graph by increasing (zooming in) or decreasing (zooming out) its zoom level.



Increase zoom level (zoom in)



Decrease zoom level (zoom out)

22.7.5 Defining a Display Section

The *Zoom Tool* allows you to directly select a specific area of the plot and increase its resolution. To do this, please proceed as follows (see figure below):

1 Using the mouse, left-click-and-hold into the plot area to set the first corner of the zoom frame.

2 Move the mouse pointer to define the size of the frame as desired.

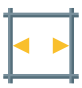
3 Releasing the mouse button will finalize the size of the frame. The selected area will be scaled to the current graph size (zoom in).

Figure 158: Setting a zoom frame

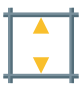
22.7.6 Auto-Fit & Auto-Scale

The toolbar and the context menu both contain a number of tools to adjust what is displayed in the diagram, in particular to ensure that all or specific data are visible.

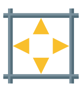
The following possibilities exist:



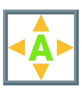
Rescales the X axis to display all current time data values for a given process data resolution.



Rescales the Y axis to display all current process data values within a given time period.



Rescales both X and Y axes to display all currently available data.



(Re-)activates auto-scaling: as long as data are being recorded, both X and Y axes will be adjusted dynamically to ensure all data are being displayed.

You may also activate auto-scaling for X and Y axes individually via the context menu:

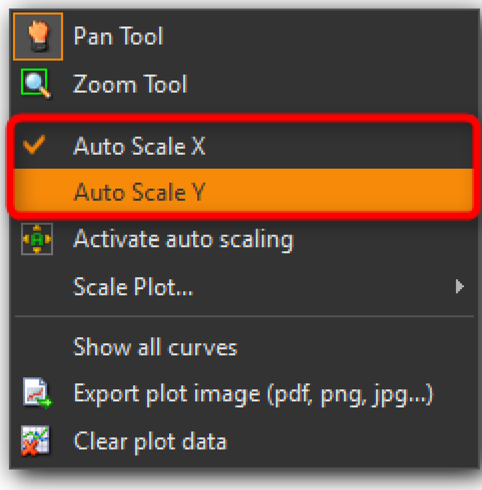


Figure 159: Auto-scaling toggle for X and Y axes

IMPORTANT. Zooming or panning within the displayed plot section will deactivate auto-scaling.

22.7.7 Show/Hide Individual Curves

To improve scaling and visibility, you may show or hide individual curves. To do this, right-click the desired item in the plot legend and select the desired function to either hide the corresponding curve only (*Hide Curve*) or all other but the corresponding curve (*Show only this curve*) as indicated in the figure below.

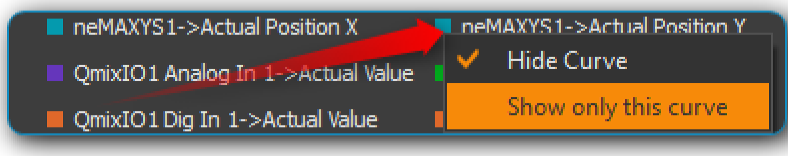


Figure 160: Context menu legend item

To revert to displaying all curves, activate the context menu from within the plot area and select the menu item *Show all curves* (see figure below).

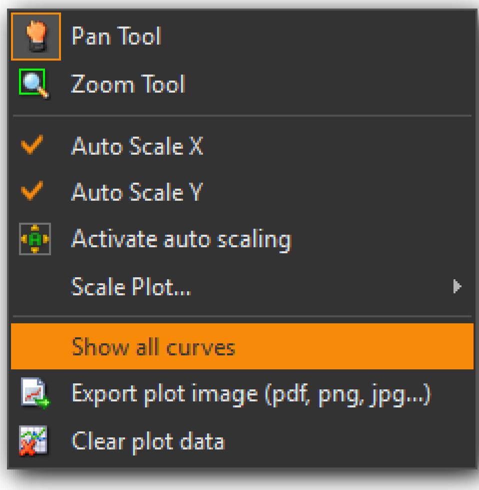


Figure 267: Context menu item - Show all curves

22.7.8 Select Curve Color

To choose a different curve color, right click an item in the plot legend. From the context menu select the menu item *Select Color* (see figure below).

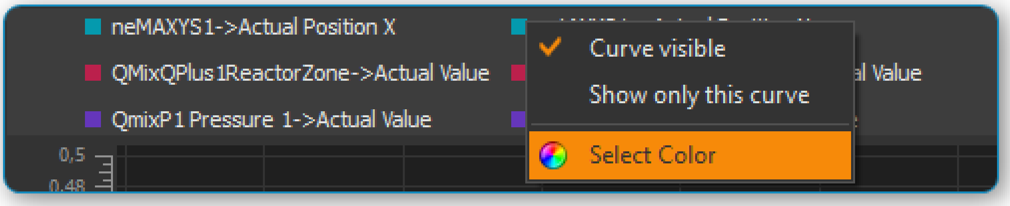


Figure 161: Context menu legend item – Color Selection

In the color selection dialog which is now shown (figure below), you can choose any color.

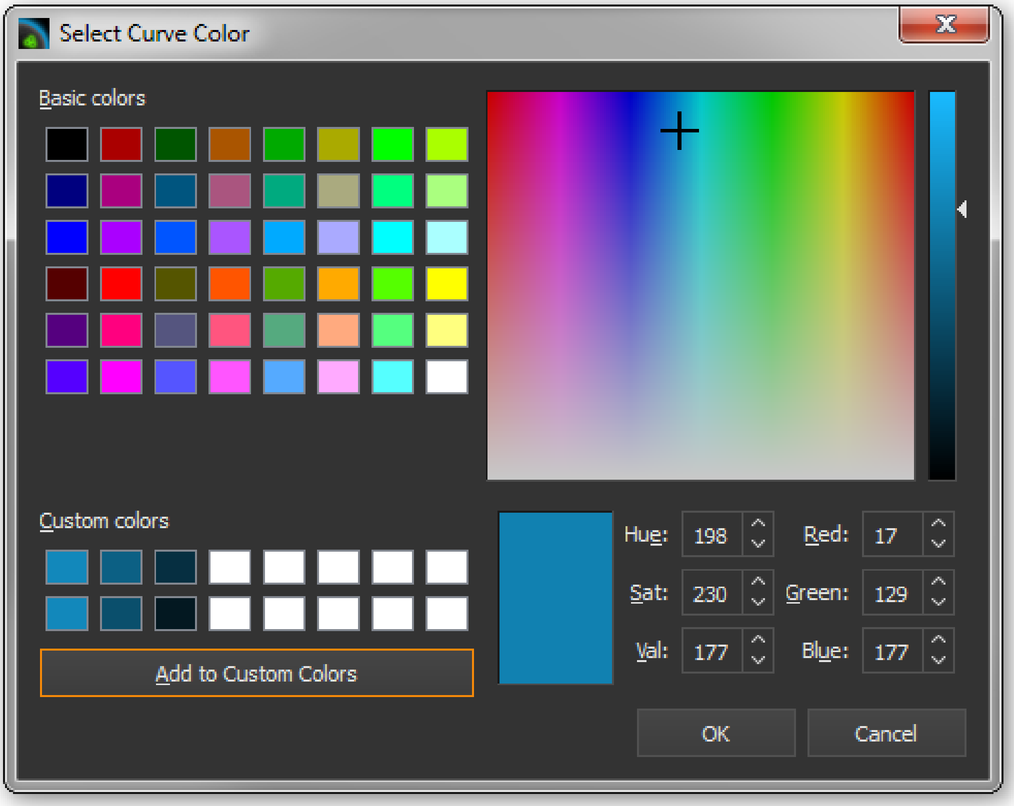


Figure 162: Color Selection Dialog

22.7.9 Exporting Plot Image

You may export a picture of the current diagram using the right-click context menu and selecting *Export plot image*. This will open a dialog box (see figure below) to define the location (folder) where the image is to be saved:

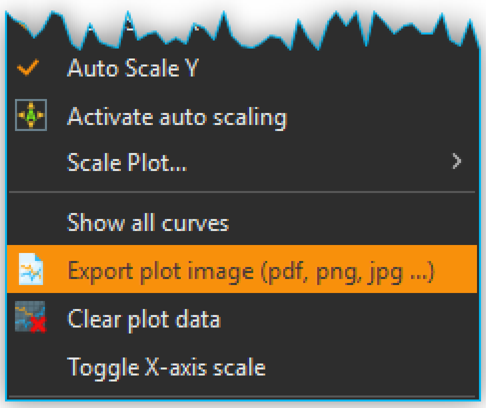


Figure 163: Diagram image export

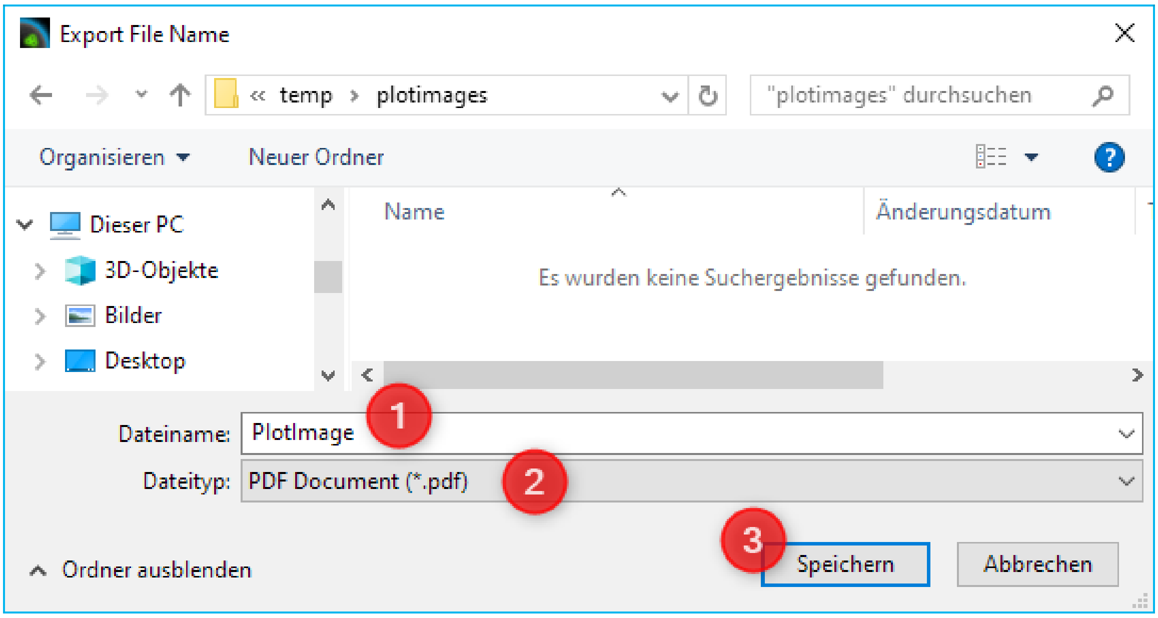


Figure 164: Diagram image export dialog

Please enter a name for the image file 1 and select the desired file type 2. The export function supports standard image file formats (*png, jpg...*) as well as scalable vector graphic formats (*pdf, svg...*).

To close the dialog and to start the image export, click *Save* 3.

22.7.10 Deletion of Diagram Data

You may clear the plot area and delete all data recorded since the start of the present recording using the context menu item *Clear plot data*. Recording will resume from this point.

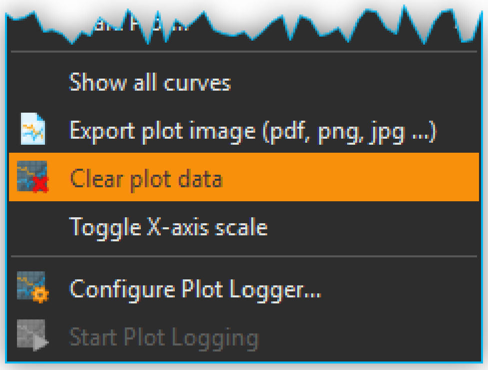
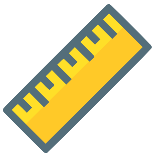


Figure 268: Context menu item - Clear plot data

22.7.11 Switching the scaling of the X-axis



You can switch the scaling of the X-axis between two different modes. By default, the X axis displays an absolute date/time stamp.

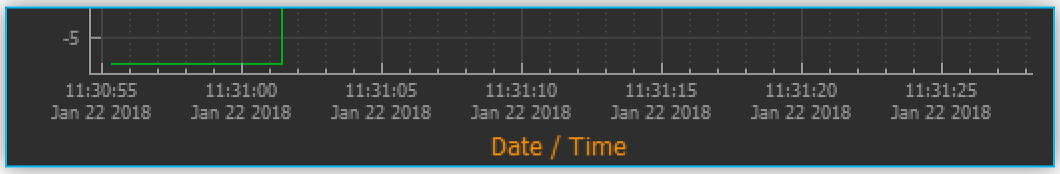


Figure 269: Plot time axis using absolute time values

You can switch the X-axis to display the relative time in seconds and milliseconds. This means that the event t_0 marks the point in time at which the recording was started.

1.

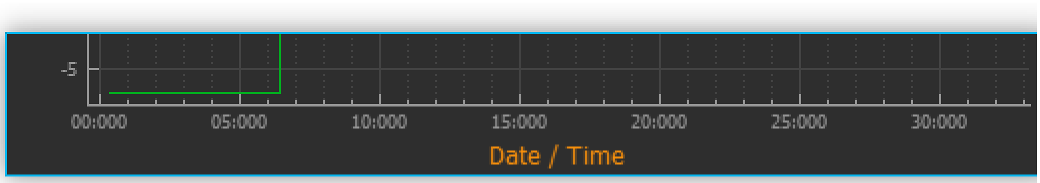


Figure 270: Plot time axis using relative time values

To toggle the axis, right-click in the diagram and select *Toggle X-axis scale* from the context menu.

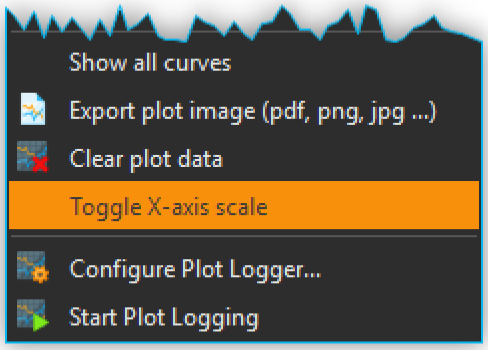


Figure 271: Context menu item - Toggle X-axis scale

22.8 Script Functions

22.8.1 Introduction

To automate the capture of data or to synchronize data capture with other processes, the data acquisition can be started and stopped using QmixElements script functions. The corresponding functions can be found in the *DAQ* category in the list of the available script functions.

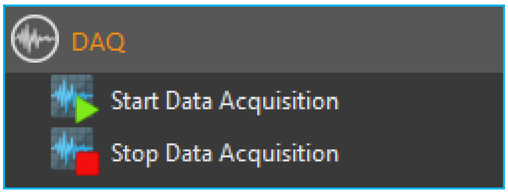


Figure 165: Logger script functions

22.8.2 Start Plot Logger

This function is used to start the data acquisition with the currently configured settings and channels. The current content of the plot is not deleted.

22.8.3 Stop Plot Logger

This function stops data acquisition.

22.9 Importing Log Files

The DAQ add-on offers the functionality to read recorded log-files into the plot. The tool for importing log files is able to recognize related log files of a long-term recording and import all related log files in chronological order. For large amounts of data, individual data points are omitted so that a maximum of one million data points per log curve are displayed in the plot.

IMPORTANT. The import tool recognizes log files that belong together and reads them in chronological order.

To read in a single log file or a series of log files proceed as follows:

(1) From the toolbar select the tool for importing log files.

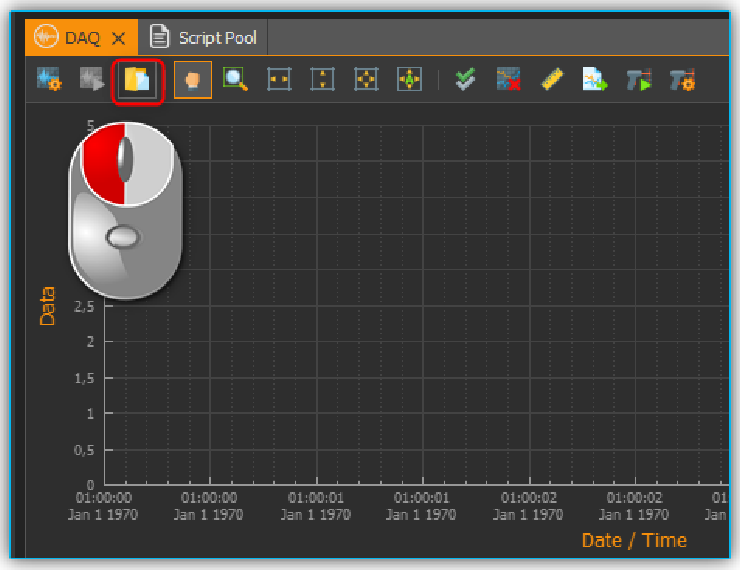


Figure 166: Opening the log file import tool

(2) Select an arbitrary file from the log file series you want to read in. In the example shown, the tool recognizes from the file numbering that the log files belong together.

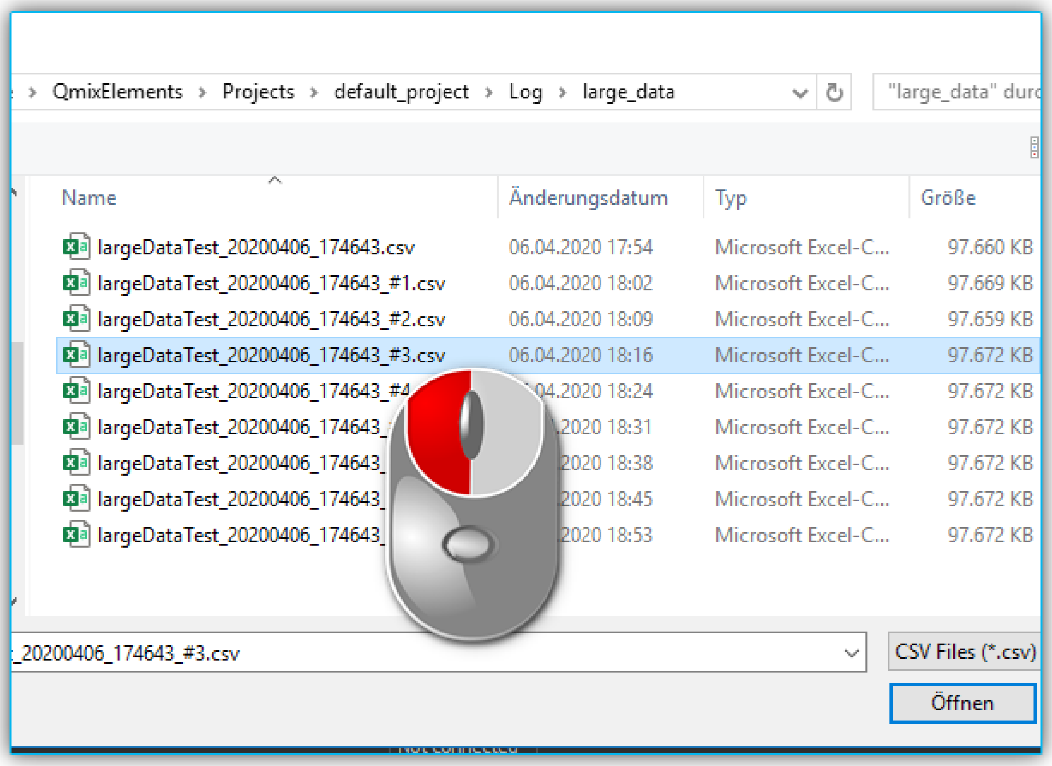


Figure 167: Selection of a log file series for reading into the graphical plot

(3) The log files are then read in. The progress window displays the progress of the import process. You can continue working with the software during the import.

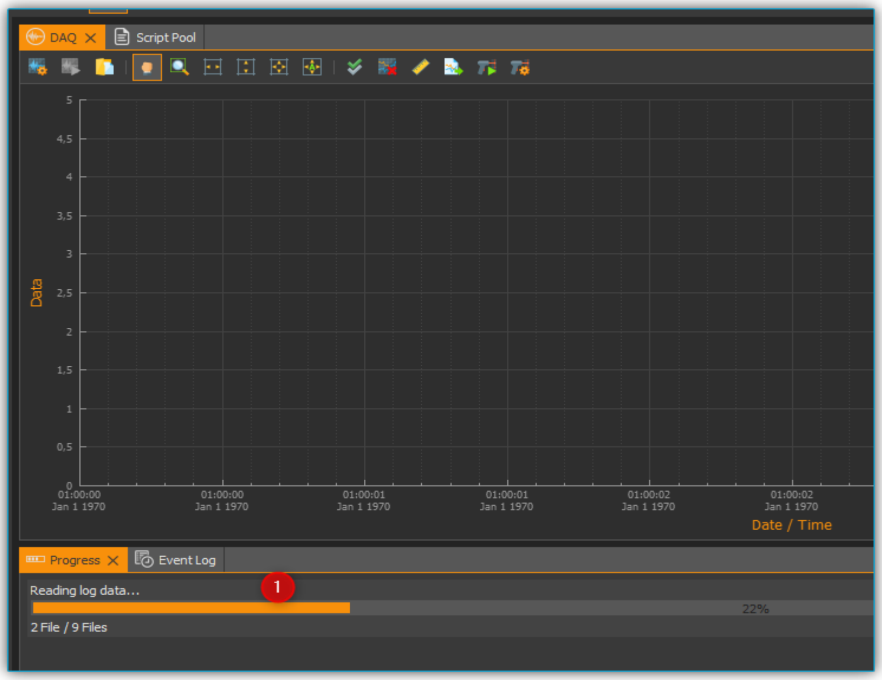


Figure 168: Import process with progress display

(4) After completion of the import process, the log curves are displayed in the graphical plot together with the file name.

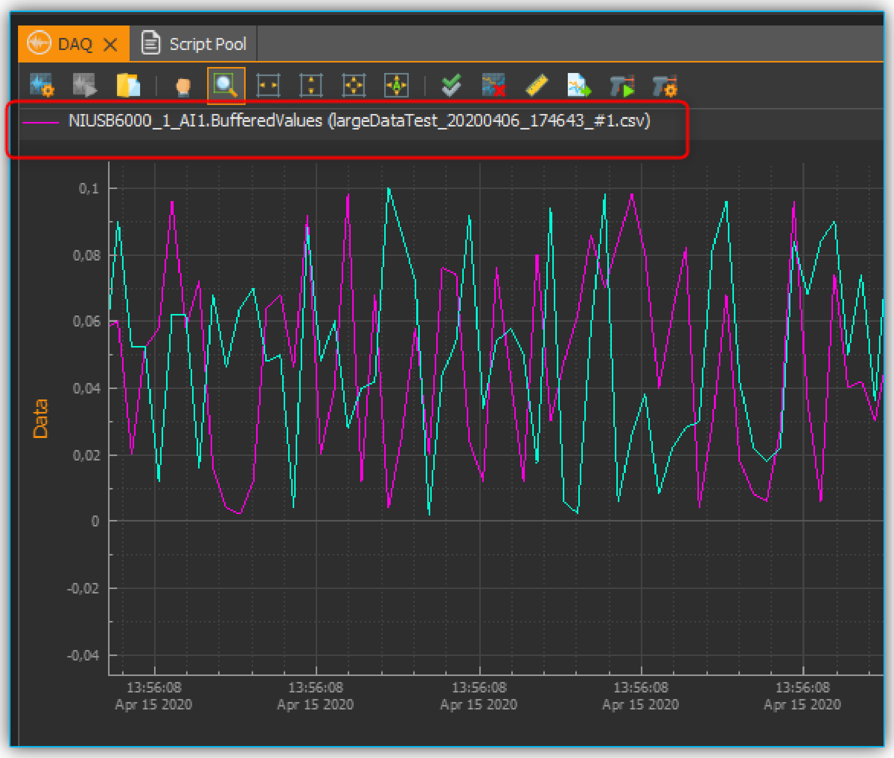


Figure 169: Imported log curves

22.10 Pattern Scan

The DAQ add-on offers the possibility to search plot curves for patterns and mark them in the graphical plot.

22.10.1 Configuring and Running the Pattern Scan

The following simple example explains pattern recognition based on the recognition of square wave signals.

(1) A plot curve is given, which contains square wave signals with a width between 5 and 8 seconds and a peak value of 5V.

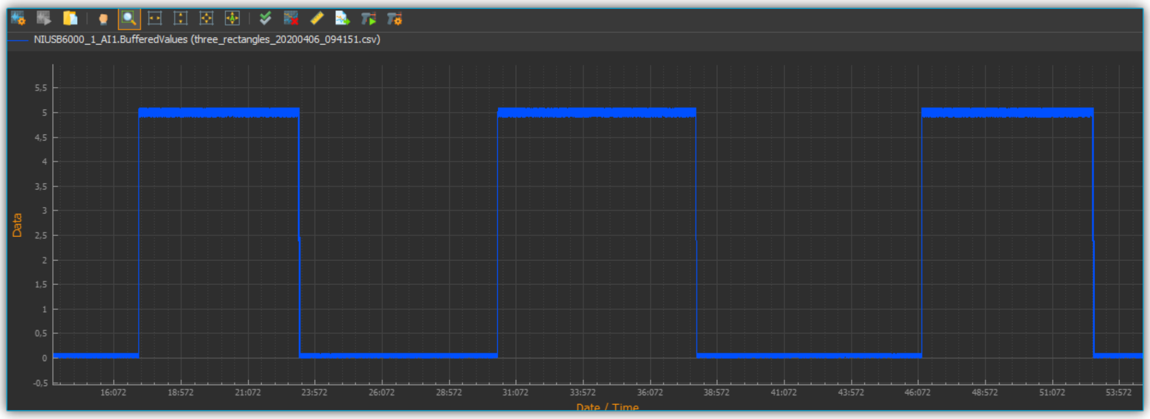


Figure 170: Plot curve containing square wave signals

(2) To configure the pattern scan, right-click the curve you want to scan 1 and select *Configure Pattern Scan* 2

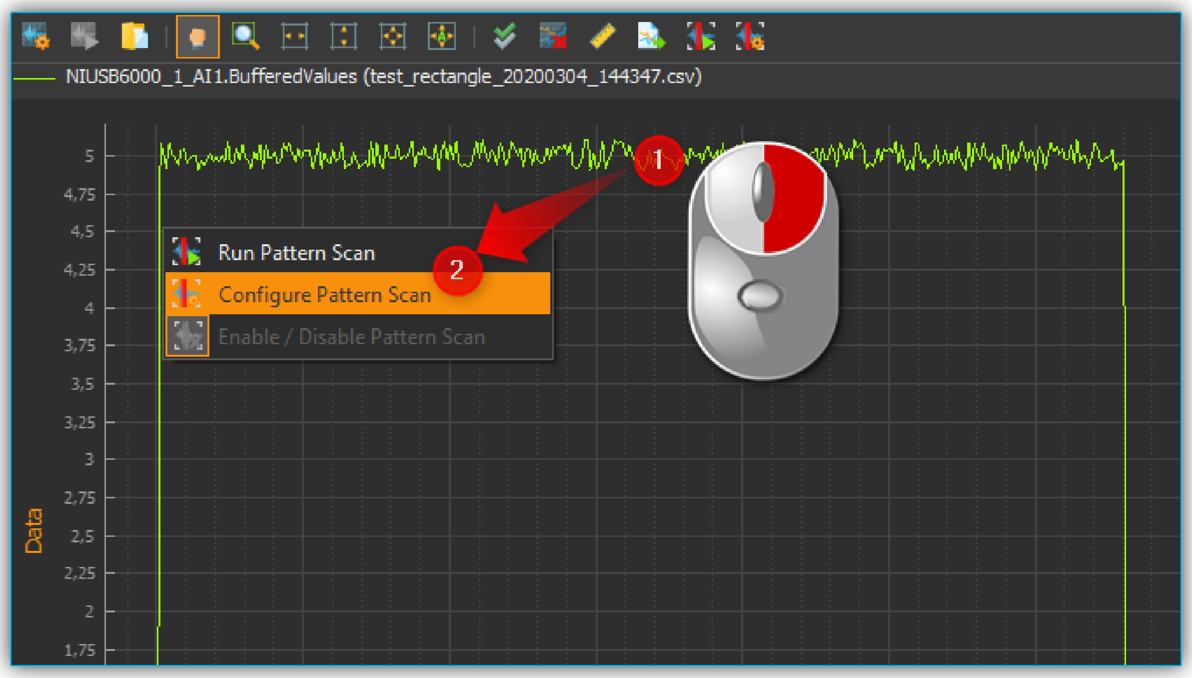


Abbildung 171: Opening pattern scan configuration

(3) In the dialog that opens afterwards, you must first create a new pattern scanner. To do this, click the *Add New Time Value Pattern Scanner* button ❶. Then you can give the pattern scanner a meaningful name ❷. In the example, "Rectangle Pattern Scanner" is selected. With the button ❸ you can delete the selected pattern scanner and with ❹ you can delete all pattern scanners.

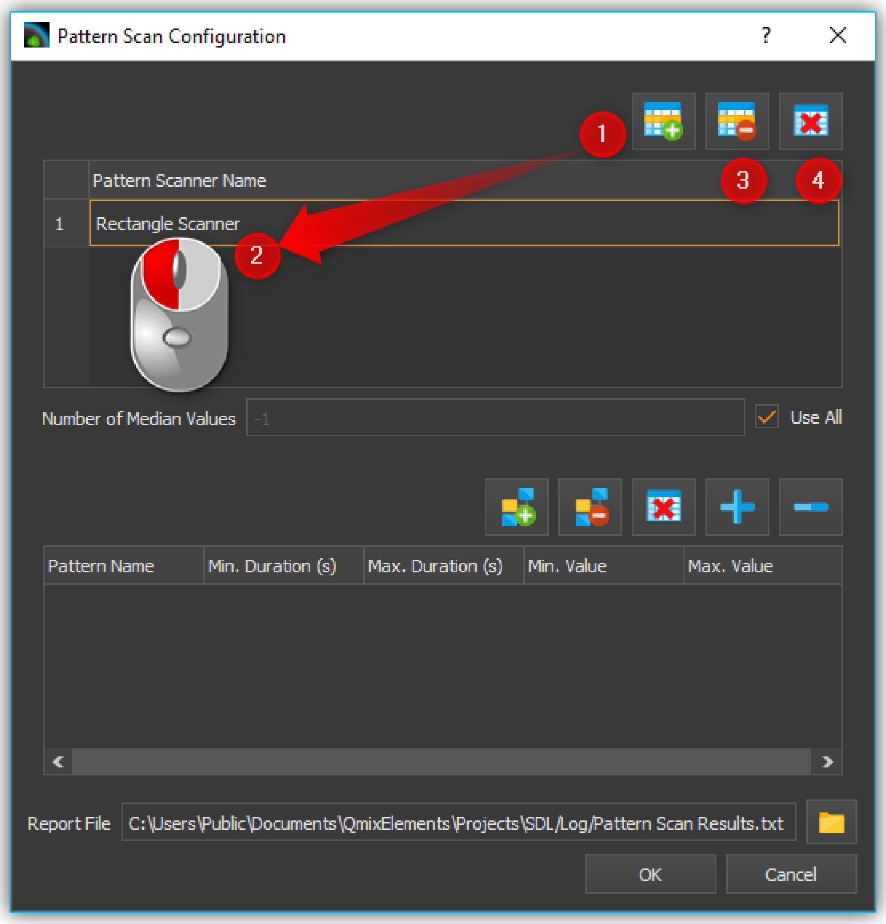


Figure 172: Creating a pattern scanner

(4) A report file is generated during pattern recognition. This file contains the measured values around the median of the recognized pattern. How many values around the median should be contained in the report file can be configured via the input field *Number of Median Values* 1. If you want all measurement values within the detected pattern to be included in the report file, select the check box *Use All* 2.

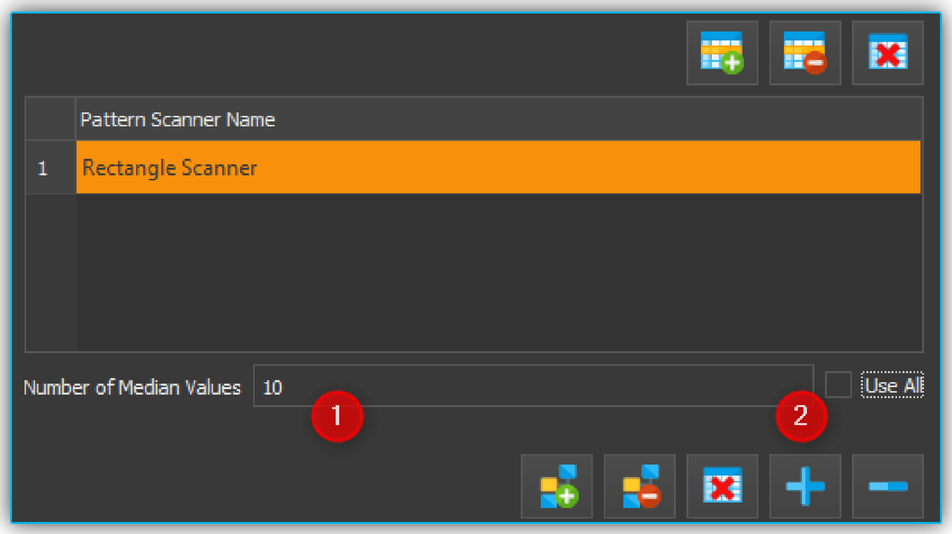


Figure 173: Configure the number of measured values around the median of the detected pattern

(5) The next step is to configure the patterns to be recognized by the pattern scanner. To do this, you must select the pattern scanner 1 whose pattern you want to configure. You can then create a new pattern using the *Create Item* button 2. You can now give the created pattern a meaningful name 3.

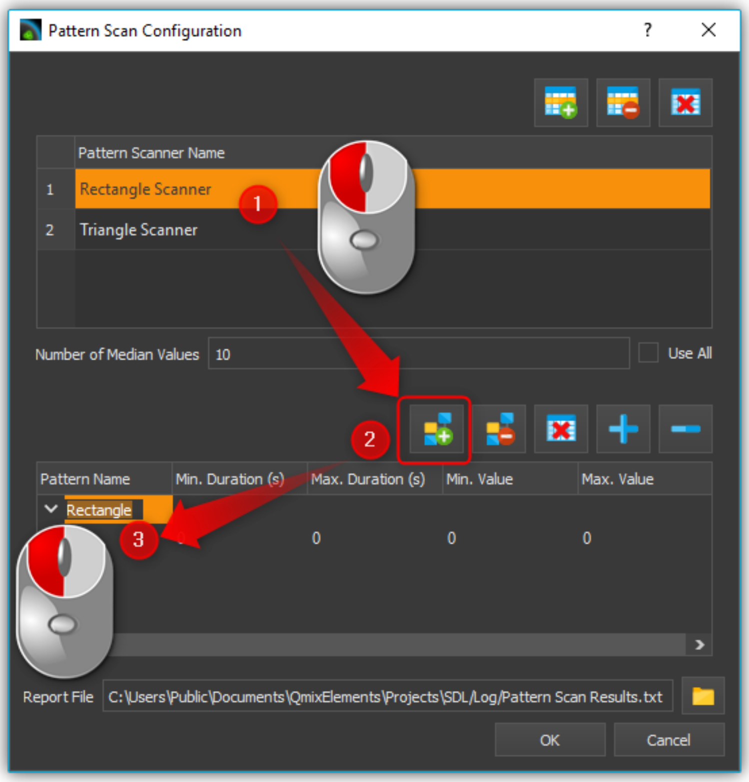


Figure 174: Creating a pattern

(6) The next step is to describe the pattern to be recognized. Since square wave signals with a duration of 5 - 8 seconds and a height of 5V are to be recognized, a section with a minimum duration of 5 seconds and a maximum duration of 8 seconds, as well as a minimum value of 4.8 and a maximum value of 5.2 is created. The value range is selected between 4.8 and 5.2 to tolerate a certain noise of the measurement signal. To do this, enter the above values in the first sample line that was automatically generated.

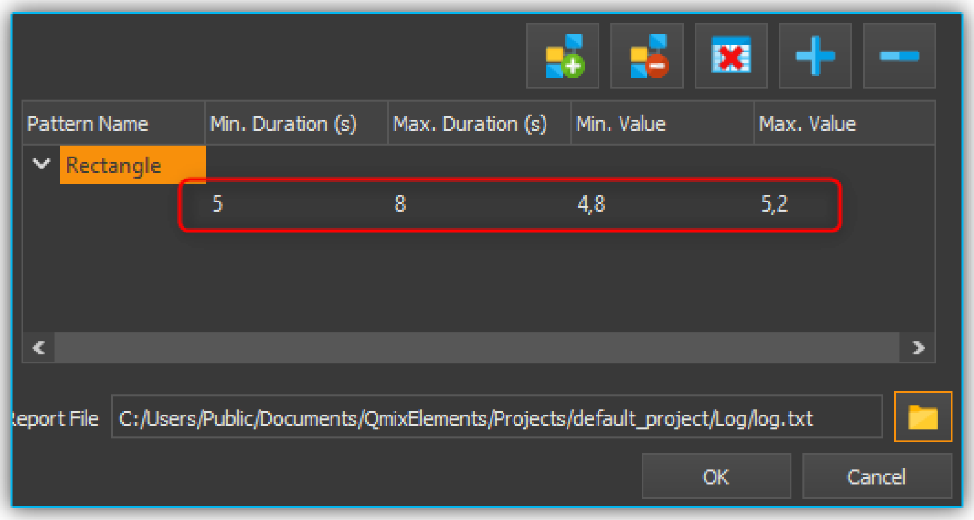


Figure 175: Creating the first pattern section

(7) Then the end of the rectangular pattern must be detected. The measuring signal must drop back to approximately 0V for a certain time. Thus a new section is inserted with a duration of 0.2 to 0.5 seconds, in which the measurement signal must remain between -0.2 and +0.2. To do this, click on the point after which the new section is to be inserted **1**, and then click on *Add Item* **2**. Then enter the specified values in the newly created line.

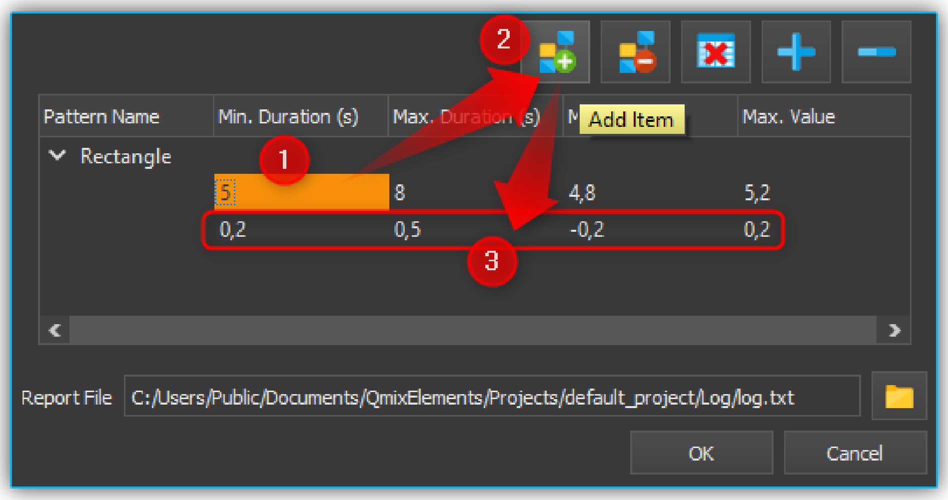


Figure 176: Creating the last pattern section

HINT. A single pattern scanner can detect multiple patterns. Repeat steps 5 - 7 to add another pattern to your pattern scanner.

(8) In the penultimate step, you can specify a file path where the report file is stored. To do this, click on the *Report File* button and enter the file path and name in the file dialog that opens. Finally, click on the *OK* button to complete the configuration of the pattern scanner.

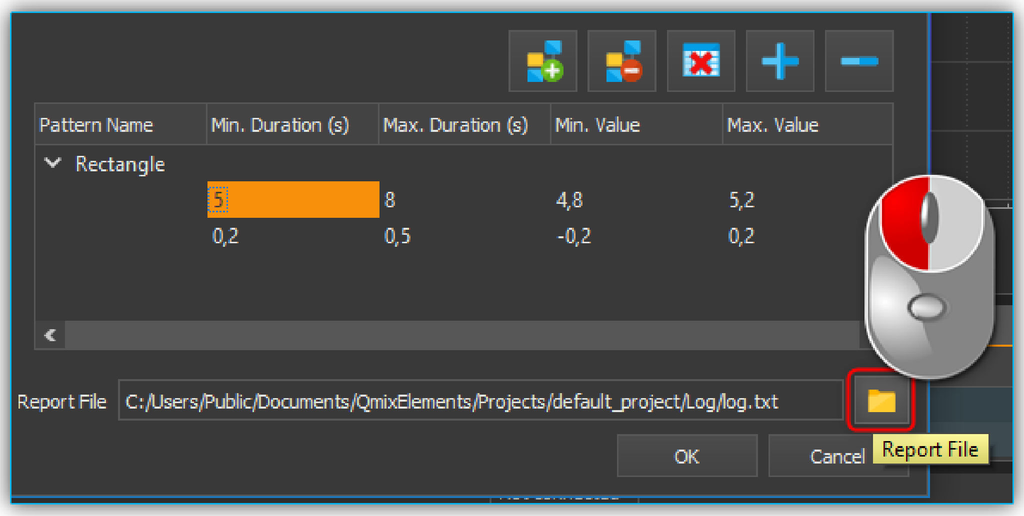


Figure 177: Specify the report file path

IMPORTANT. The pattern recognition settings for a plot curve are saved. When the plot curve is reloaded, the corresponding pattern scanner is automatically assigned to it.

(9) In the last step the pattern scan is executed. To do this, click the *Run Pattern Scan* button 1. The start 2 and end 3 points of the recognized patterns are then marked in the plot. A message 4 will show you how many patterns have been detected in the plot curves being examined. The report file is also written.

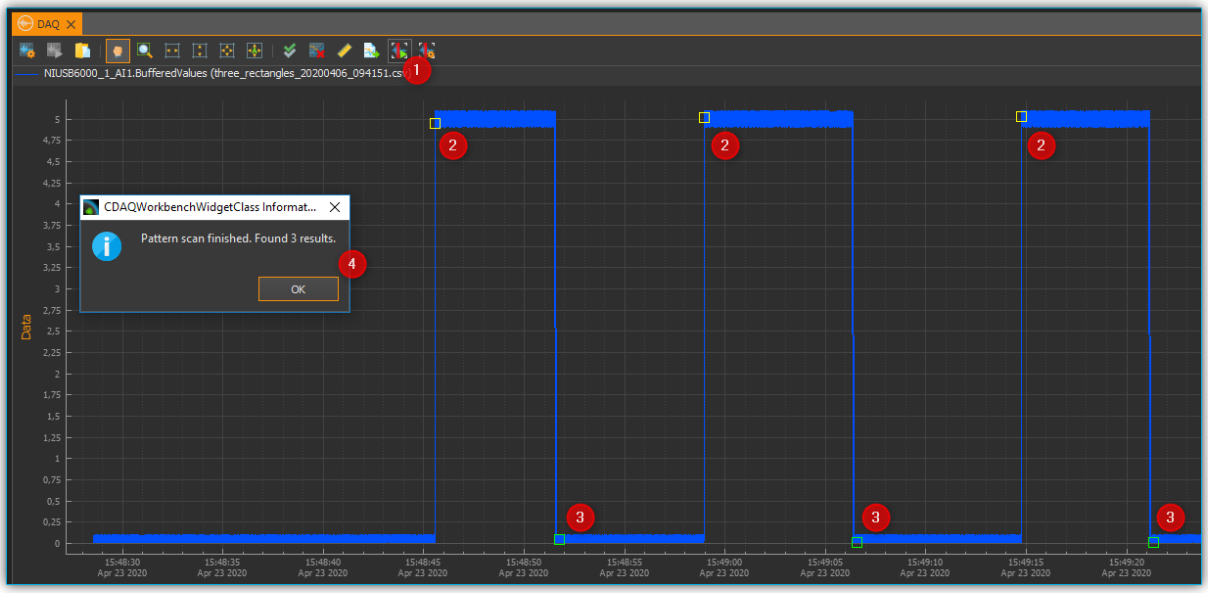


Abbildung 178: Result of the pattern scan

22.10.2 Using an Existing Pattern Scanner

Once you have configured a pattern scanner, you can assign it to a variety of other plot curves. Proceed as follows:

(1) Right-click on the curve you want to scan and select *Configure Pattern Scan.*

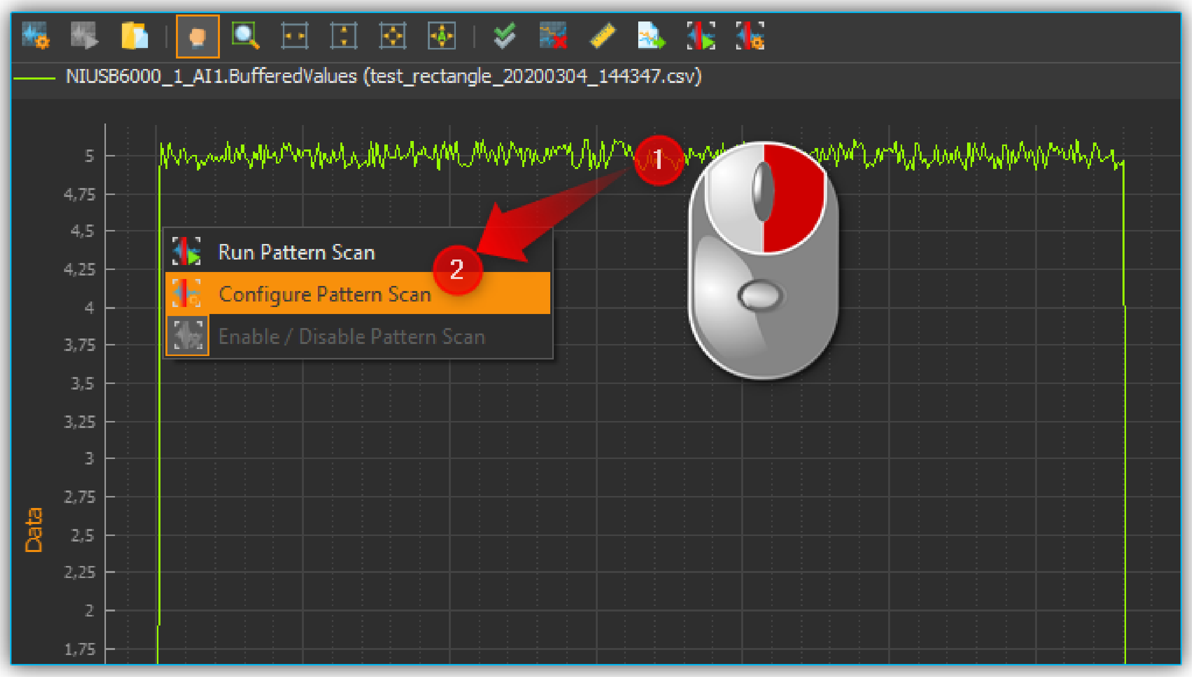


Abbildung 179: Opening pattern scan configuration

(2) In the dialog that opens, select the pattern scanner you want to use and then click *OK*.

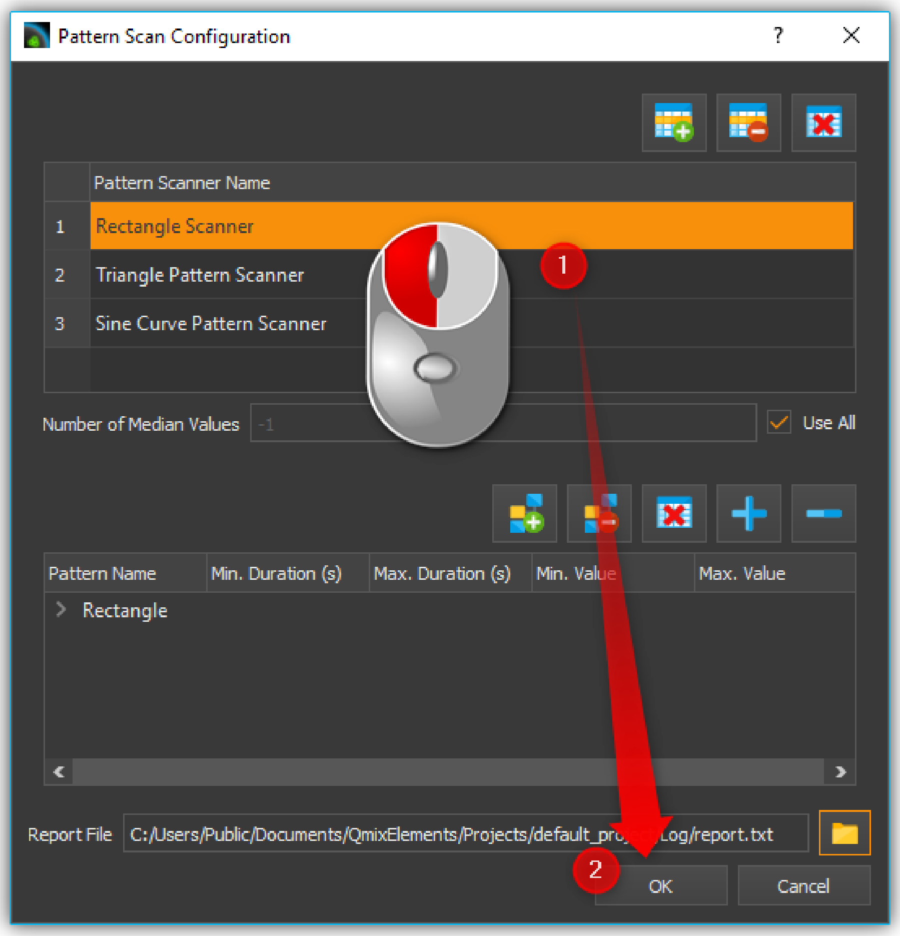


Figure 180: Selection of an existing pattern scanner

22.10.3 Enabling / Disabling Pattern Recognition for a Plot Curve

To enable or disable the pattern recognition for a plot curve, right-click on the plot curve 1 and select Enable / Disable Pattern Scan 2.

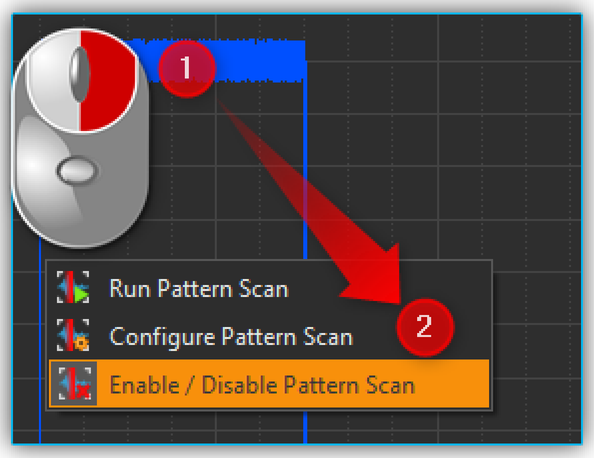


Abbildung 181: Enabling / Disabling Pattern Recognition

22.10.4 Result Analysis

22.10.4.1 OVERVIEW

Using the result analysis for pattern recognition, you can compare the individual results of a pattern recognition within a measured value curve. The individual results are superimposed in a separate analysis plot. This allows you to assess how they have changed during the course of the measurement. You can access the result analysis of pattern recognition by clicking the *Pattern Analysis* button 1 in the *DAQ* group (see figure below).

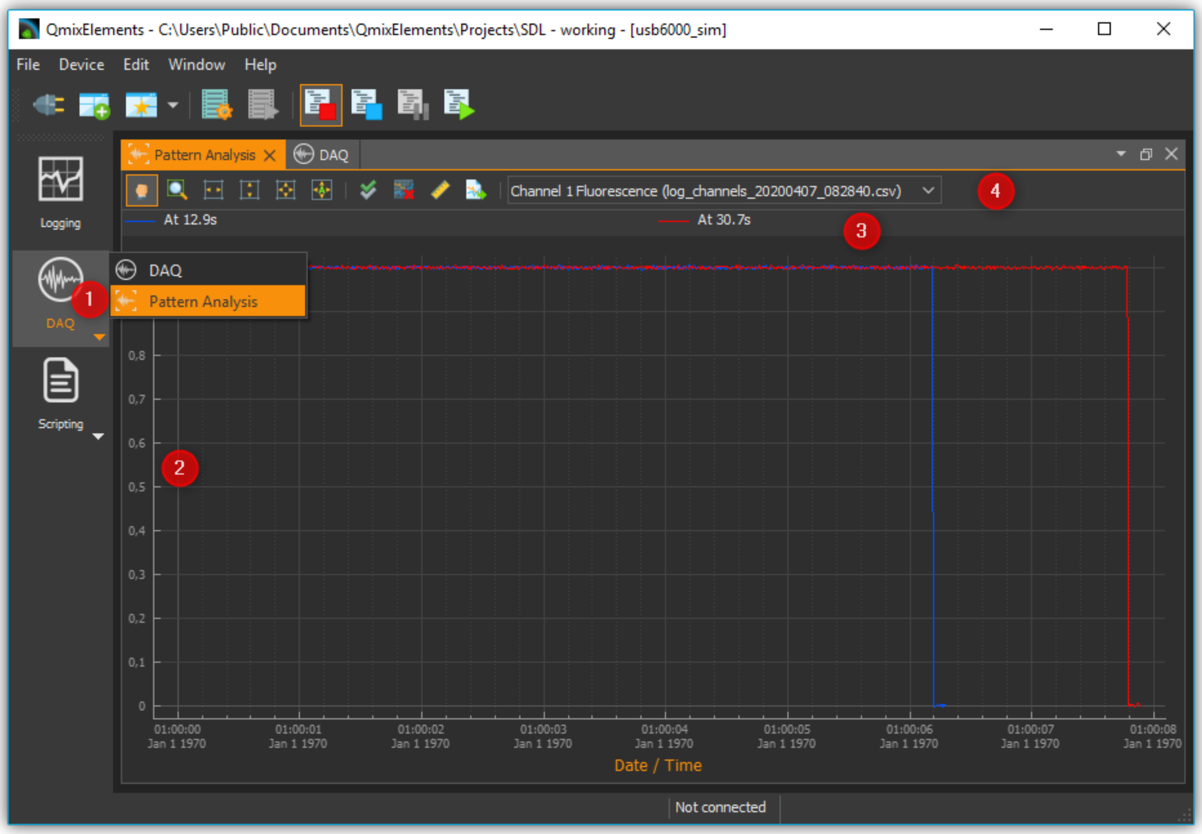


Figure 272: Pattern Analysis Overview

The most important controls are shown in the figure above:

1 *Pattern Analysis* selector button – This button takes you to the result analysis of the pattern recognition.

2 *Diagram* – In the diagram, the measured values of each individual pattern recognition result are displayed one above the other. Here you can compare the individual results with each other.

3 *Legend* – The legend contains the designation of all curves that are displayed in the diagram with the corresponding color. The curve designation contains the relative time at which the result occurred in the original trace.

4 *Toolbar* – Here you will find buttons for navigating within the display, as well as for selecting the plot curve in the process data diagram whose pattern recognition results are to be displayed in the result analysis. The operating elements of the navigation work in the same way as the operating elements in the process data diagram (see section 1.7.Diagram Navigation & Use) and are therefore not explained again.

22.10.4.2 CARRYING OUT A RESULTS ANALYSIS

Section 1.10.1.Configuring and Running the Pattern Scan describes how to perform pattern recognition. Assume that after performing pattern recognition there are two plot curves with results (see figure below).

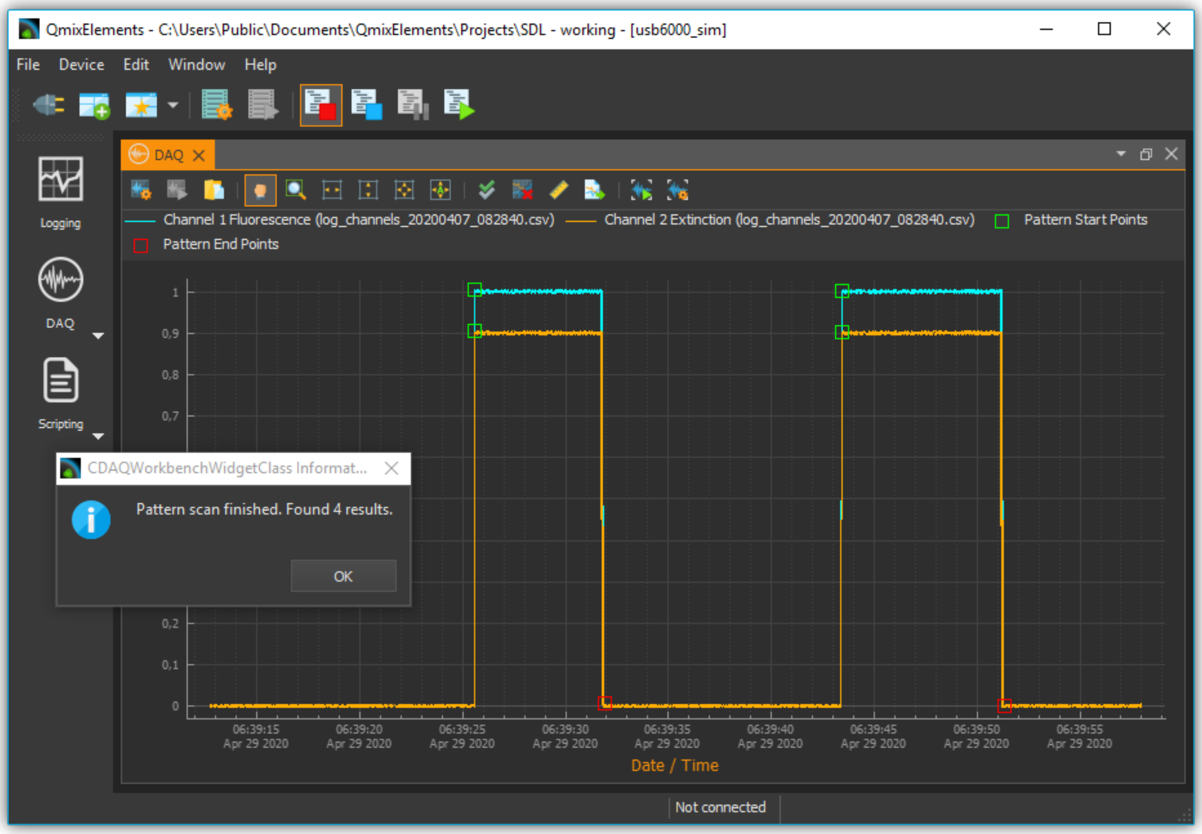


Figure 273: Pattern scan with two plot curves

In this case, the following steps take you to the results analysis:

(1) In the first step, select the *Pattern Analysis* button from the *DAQ* group to access the results analysis.

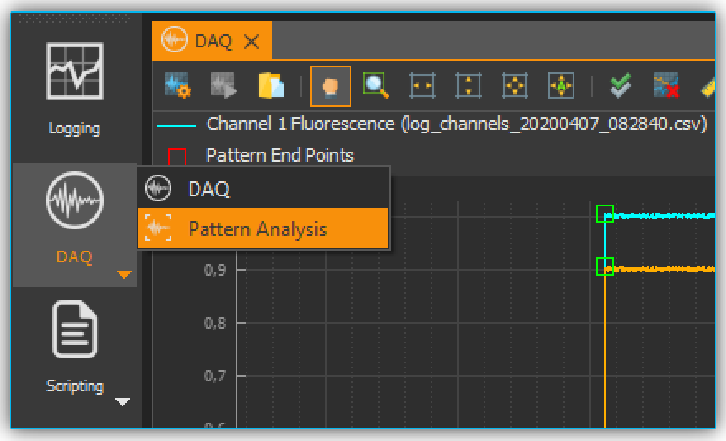


Figure 274: Button Pattern Analysis

(2) In the second step, select the plot curve whose results you would like to have displayed from the drop-down field in the toolbar. In the example shown, "Channel 2 Extinction" is selected.

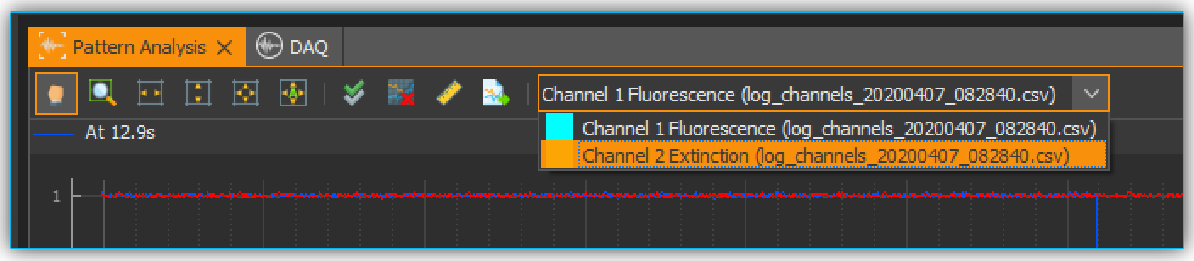


Figure 275: Selection of the plot curve for the result analysis

(3) The diagram now shows the individual pattern recognition results for "Channel 2 Extinction". For this plot curve, two matches were found with the configured pattern. You can see that both results have the same amplitude. However, the result which occurred after 30.7 seconds takes about 1.6 seconds longer than the result which occurred after 12.9 seconds.

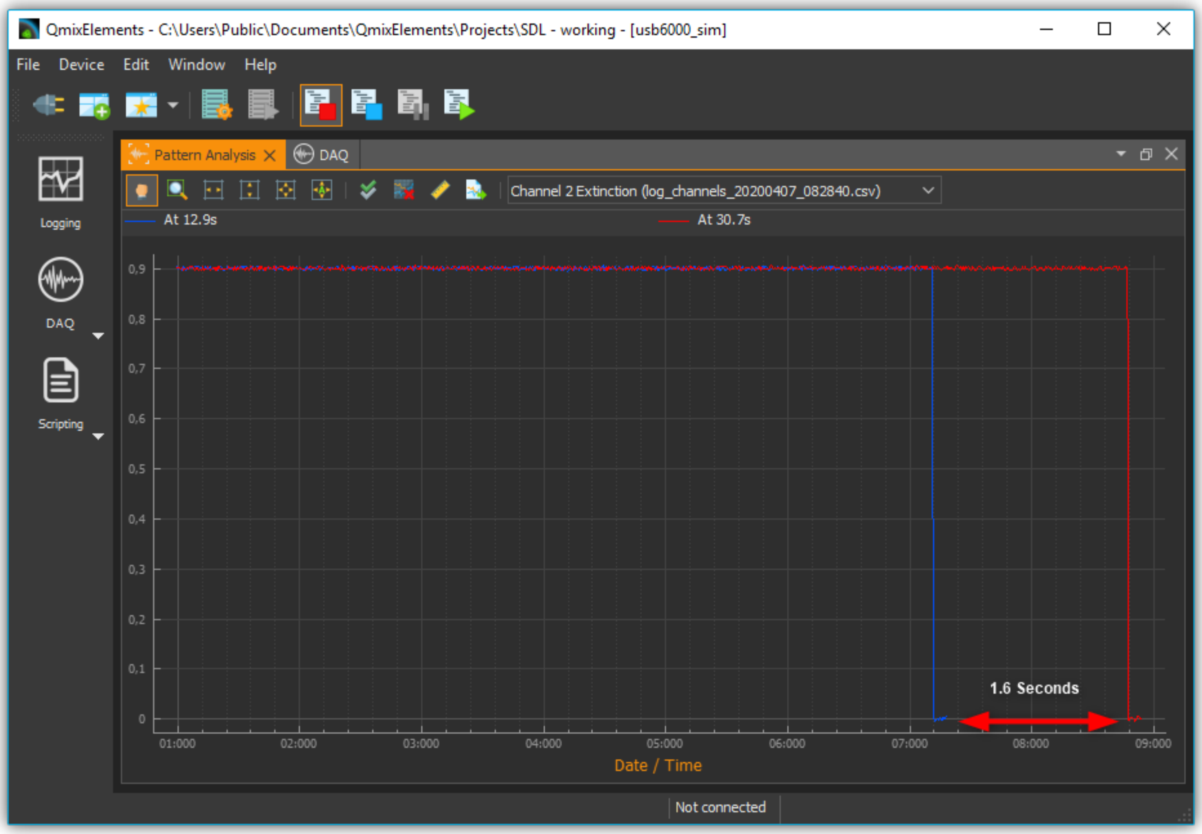


Figure 276: Result analysis pattern recognition

23 Python Add-on

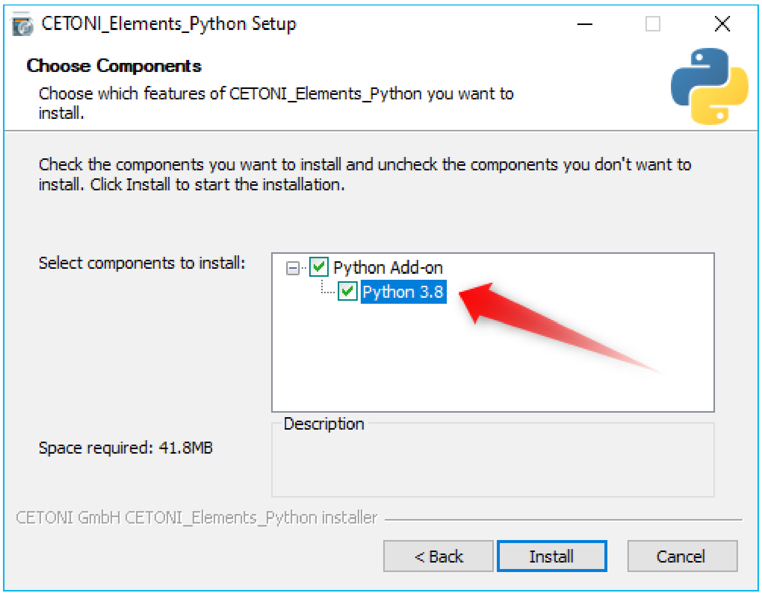
23.1 Installation

The Python add-on is not included in the standard installation package, but must be installed additionally as an add-on. The versions of the Python add-on and the installed CETONI Elements software should match. For example, if you want to install the Python add-on with version number 20220126, the CETONI Elements software 20220126 should be installed.

IMPORTANT. The version number of the CETONI Elements software and the Python add-on should match. The version number of the Python add-on must never be greater than that of the CETONI Elements software. The greater the difference between the version numbers, the greater the risk of problems and errors.

For installation, run the file *CETONI_Elements_Python_64bit_Setup.exe*. The installation wizard will then guide you through the installation of the software.

To use the Python add-on, a suitable Python 64-bit version must be installed on your computer. In the installation step *Choose Components* you can see the required version (here e.g. Python 3.8) and select it for installation if it is not yet available on your computer.



IMPORTANT. On Windows, you must be logged in with administrator privileges to install the add-on, as it is installed for all users.

IMPORTANT. The installed Python version **must** match the version specified in the installer. The plugin will not work correctly with other versions.

23.2 Introduction

The Python add-on allows you to execute Python code in CETONI Elements scripts via a corresponding script function. It thus provides an easy way to include the Python scripting language in your CETONI Elements scripts.

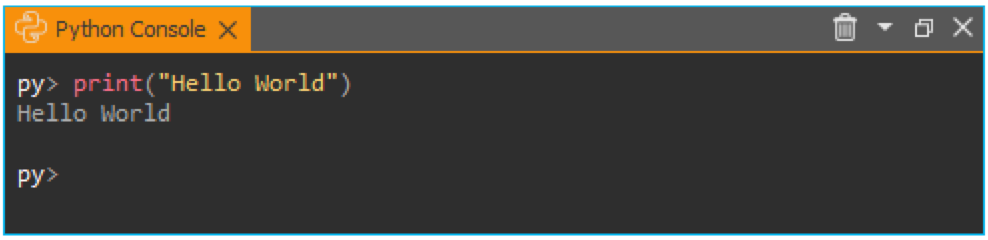
The embedded Python interpreter offers the same functionality as an execution of Python scripts via *python* command. I.e. if you install libraries via *pip* for the Python version used by the add-on, the functions of these Python libraries are also available in CETONI Elements.

This gives you many additional possibilities such as accessing and parsing files, accessing databases, TCP/IP network communication via Python sockets or using complex analysis functions implemented in Python.

23.3 Python Console

23.3.1 Overview

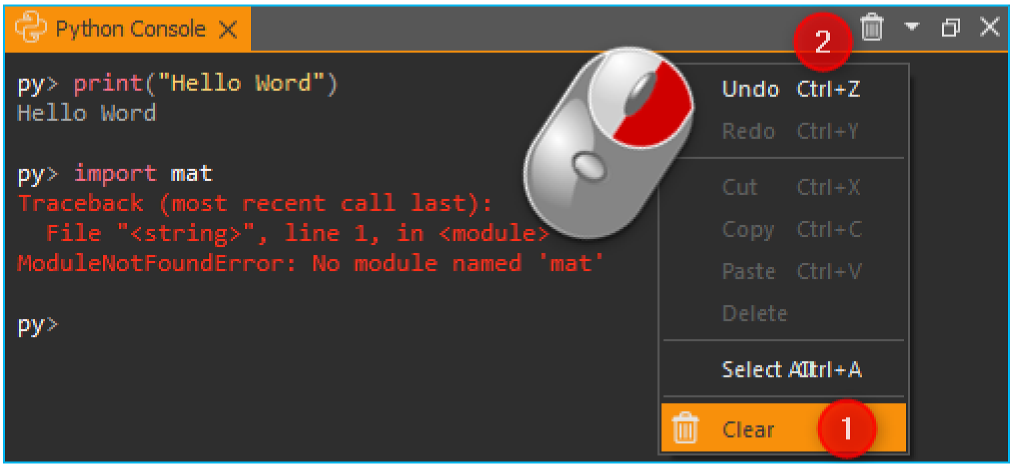
The Python add-on has a Python console that allows you to interactively enter and execute Python code similar to the standard Python console. You can show the Python console from the main menu (*Window → Show View → Python Console*):



In the Python console, you can try out commands, test access to objects, and it helps you debug problems. If function calls work in the console, you can use them in your script.

23.3.2 Context Menu

The context menu of the Python console contains the usual commands for text editing:

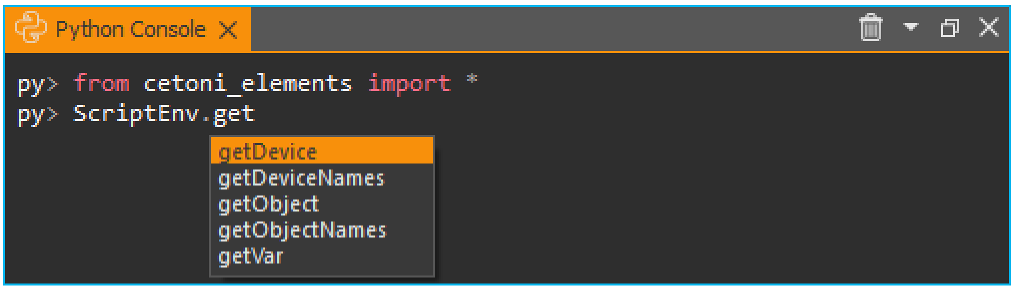


To delete the contents of the console, select *Clear* ❶ in the context menu or click the trash can icon ❷ in the title bar.

ATTENTION. Risk of malfunction or data loss !
Do not use the Python Console in a running experiment, process or during normal operation or productive use. Incorrect input or access to resources can possibly lead to a crash of the software.

23.3.3 Code completion

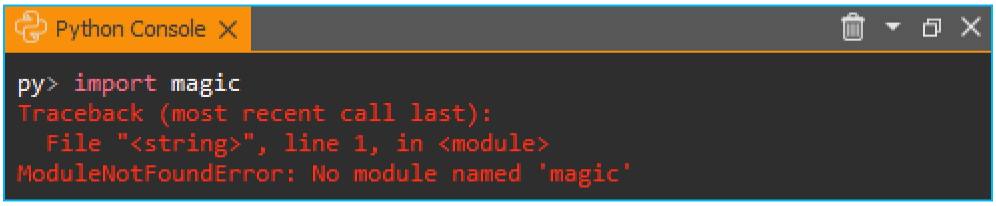
The Python console supports you during the typing with a simple code completion. I.e., if you have imported modules, then the code completion supports you with the source code editing by suitable suggestions, like you are used to it from other code editors.



In the example above all functions from the `cetoni_elements` module were imported. After entering `ScriptEnv.get`, matching functions for the `ScriptEnv` object are displayed that start with `get`.

23.3.4 Error messages

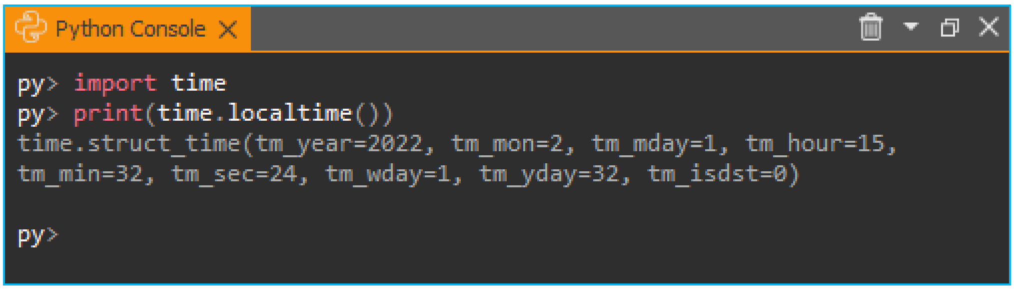
Errors that occur when executing code in the Python console, as well as errors that occur when executing Python code in the Python script function, are displayed to you as red text in the Python console.



23.4 Python Modules

23.4.1 Import standard modules

The plugin uses a standard Python installation and thus has access to all standard Python modules or to other modules installed via *pip*. You can import a module as usual using the `import` statement. In the following example the `time` module is imported and used:

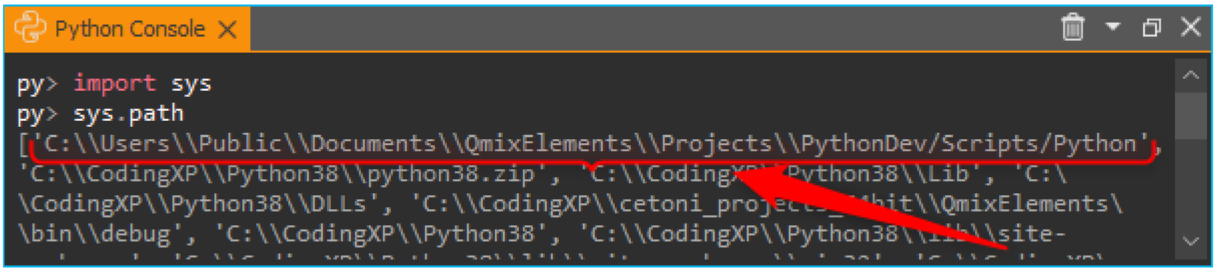


23.4.2 Import custom modules

If you use your custom modules that you want to distribute or ship with your project, you can store them in the *Scripts/Python* subfolder of your current project. This directory is added by the software to the `sys.path` of the Python interpreter. For example, if you are working in the *PythonDev* project, then the absolute path to this directory would be:

C:\Users\Public\Documents\QmixElements\Projects\PythonDev\Scripts\Python

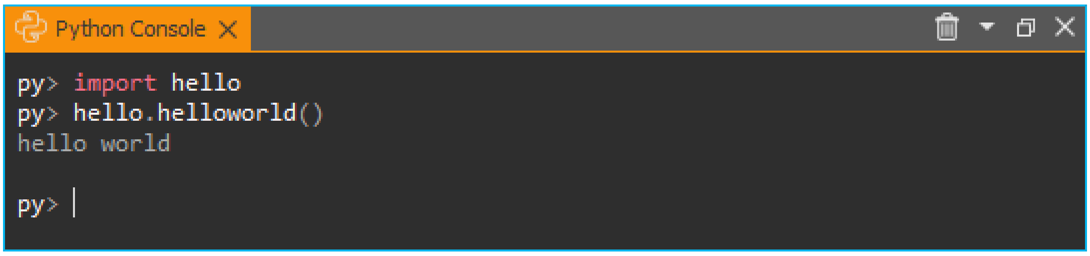
If you import the module `sys` in the Python console, then you can see via `sys.path` that the above directory is in the import path.



To test the import functionality, create the *hello.py* file in the above folder with the following content:

```
def helloworld():
    print("hello world")
```

You can now import this module and use the `helloworld` function:



23.5 Accessing application objects

23.5.1 The cetoni_elements module

The software adds a module `cetoni_elements` to the Python interpreter, which can be used to access application objects. After importing the module via `from cetoni_elements import *` you can access these objects.

The following objects are currently available in the module

`ScriptEnv`

The ScriptEnv object is the central object for accessing available devices and application objects.

`getDeviceNames()`

Returns a tuple with all device names that can be accessed from Python. Use the device name when calling the `getDevice()` function to get the corresponding device object.

getDevice(devicename)

Returns the device object for the given device name. The device object provides access to device-specific functions and properties of this device. The device names are the names that are also used in the CETONI Elements script system to access devices or device properties. Example:

```
pump = ScriptEnv.getDevice("Nemesys_S_1")
```

getObjectNames()

Returns a list of all registered application objects that are not devices and that can be accessed from Python, such as the graphical logger.

getObject(objectname)

Returns the object with the given object name. Example:

```
plot = ScriptEnv.getObject("ProcessDataGraph")
```

getVar(varname)

Returns the value of the script variable with the given name. Example:

```
flow = ScriptEnv.getVar("$FlowRate")
```

setVar(varname, value)

Sets the value of the script variable to the given value Example:

```
ScriptEnv.setVar("$TargetPos", 25)
```

setVars(dict)

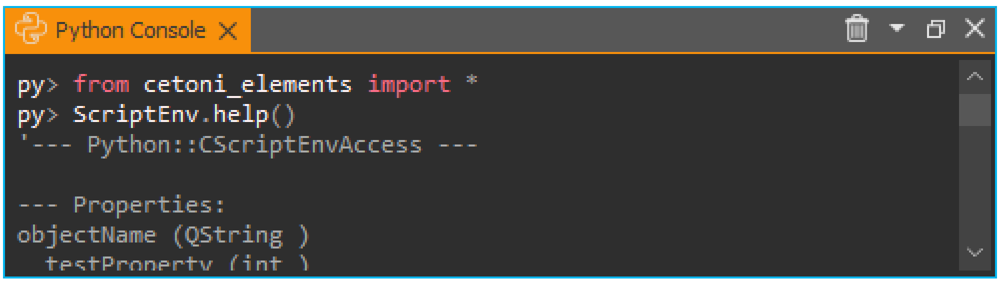
Sets multiple script variables using a Python dictionary. Example:

```
ScriptEnv.setVars({"$Value1" : 0.5, "$Value2" : 1.5})
```

help()

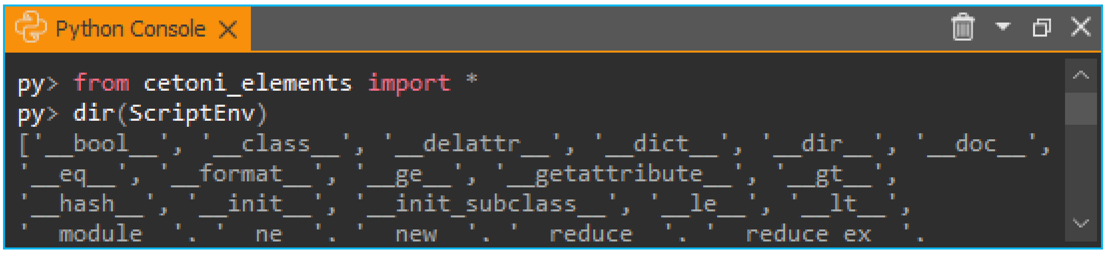
Displays an overview of all properties and functions of the object.

The following image shows how to import the `cetoni_elements` module and then call the `help()` function of the `ScriptEnv` object:

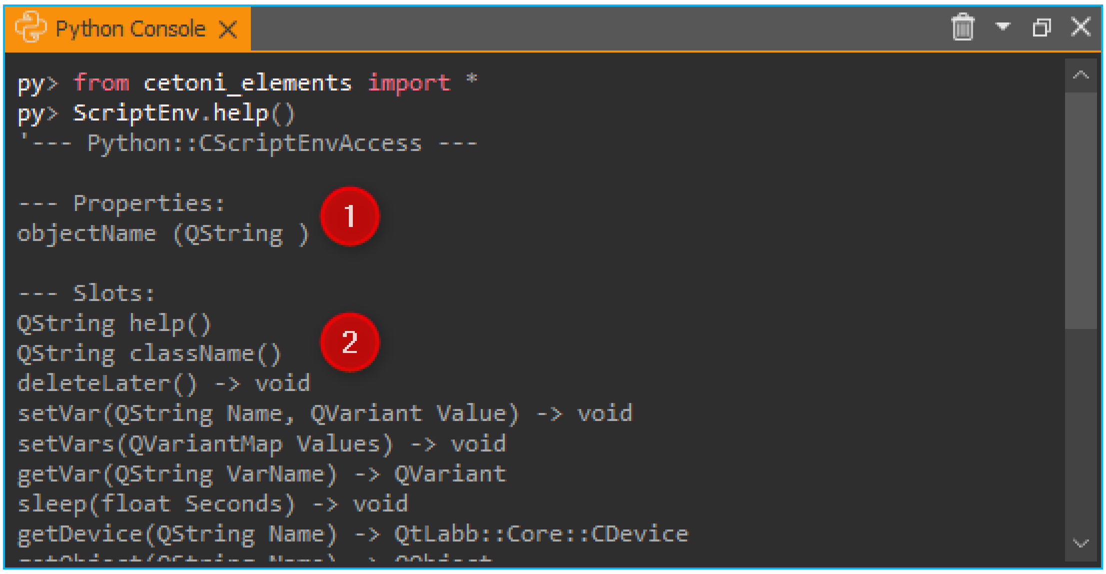


23.5.2 Display properties and methods of objects

Using the Python function `dir(object)` you can display all methods and properties of application objects. For example, you can use it to display all the functions and properties of the `ScriptEnv` object.



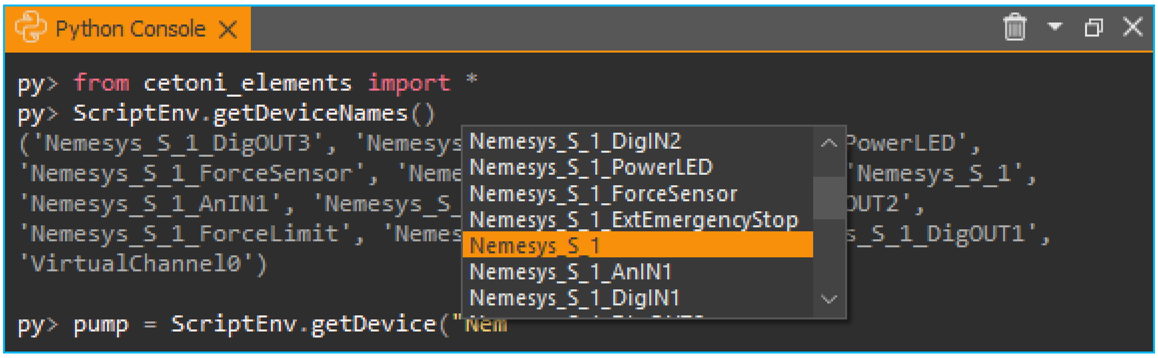
Alternatively, all application objects also have a `help()` function that provides a clearer display of an object's properties and methods. In the figure below you can see the call of `ScriptEnv.help()`. The console shows the *Properties* 1 and methods (*Slots* 2) of the *ScriptEnv* object:



HINT. Use the `dir(object)` and `Object.help()` functions to get an overview of the methods and properties of a given object.

23.5.3 Using device objects

You can access device objects using the `ScriptEnv.getDevice()` function. To get an overview of available device names you can call the `ScriptEnv.getDeviceNames()` function.



The code completion helps you to enter a device name by showing you a list of matching names (see figure above). If you call the `getDevice()` function without an assignment to a variable, then you can see in the console if the call was successful:

```
py> ScriptEnv.getDevice("Nemesys_S_1")
CNemesys4Pump (QtLabb::CNemesys4Pump at: 0x000002402DDCBF20)
```

To access a device, assign the result of the call to `getDevice()` to a variable. In the following example, we assign the device object for the first Nemesys S pump to the variable `pump`:

```
py> pump = ScriptEnv.getDevice("Nemesys_S_1")
```

Now you can use the `dir(pump)` and `pump.help()` functions to get an overview of the available methods and functions of the pump object.

HINT. Use the `dir(object)` and `Object.help()` functions to get an overview of the methods and properties of device objects.

IMPORTANT. For many devices, accessing device properties and methods or displaying help via `Object.help()` is only possible if the application is connected to the devices.

ATTENTION. Risk of malfunction / data loss !
Via device functions you may have access to functions that are not available in the graphical user interface. Always test functions outside running processes and not during

productive use. Incorrect input or access to resources, methods or properties can lead to malfunctions or a crash of the software.

When you are connected to the device, you can now access the device functionality via the device object functions. For example, you can trigger a refill process for the pump:

```
py> pump.refillSyringe()
```

or stop the pumping operation:

```
py> pump.stopPumping()
```

The following code shows how to import the `cetoni_elements` module, get the device object for the Nemesys S pump via the `ScriptEnv` object, and then start an emptying process of the syringe:

```
py> from cetoni_elements import *
py> pump = ScriptEnv.getDevice("Nemesys_S_1")
py> pump.emptySyringe()
```

23.5.4 Using application objects

Similar to device objects, you can also access application objects that are no devices. Use the two functions `ScriptEnv.getObject()` and `ScriptEnv.getObjectNames()` for this

The following code shows how to import the `cetoni_elements` module, get the application object of the graphical logger via the `ScriptEnv` object and then start logging:

```
py> from cetoni_elements import *
py> plot = ScriptEnv.getObject("ProcessDataGraph")
py> plot.startLogging()
```

HINT. Use the `dir(object)` and `Object.help()` functions to get an overview of the methods and properties of application objects.

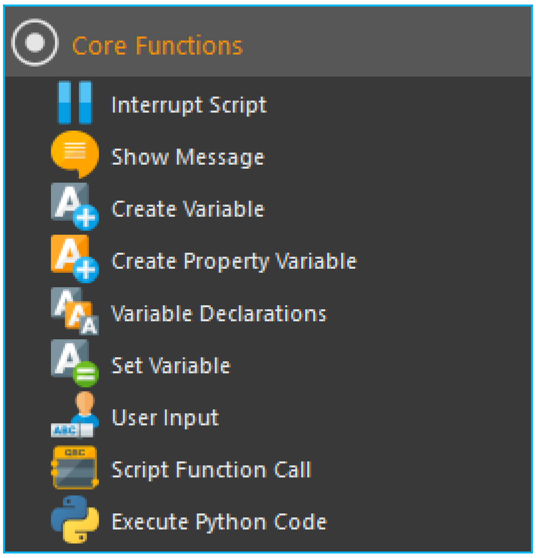
ATTENTION. Risk of malfunction / data loss !

The functions of the application objects may give you access to functions that are not available in the graphical user interface. Always test functions outside running processes and not during productive use. Incorrect input or access to resources, methods or properties can lead to malfunctions or a crash of the software.

23.6 Python Script Function - Execute Python Code

23.6.1 Overview

The Python plugin adds the *Execute Python Code* script function to the *Core Functions* category of the *Script Pool:*



This function allows you to execute Python code in the script system of the application. When you insert the function into your script, you will see the initial Python script in the configuration area. This script contains the two functions `script_exec()` and `script_abort()`:

```
# Implement your script code in this function
# Avoid blocking function calls
def script_exec():
    return


# Implement your clean up code here in case of script stop
# Keep execution time of this function short and do not use
# any blocking function calls
def script_abort():
    return
```

When the script is executed, it is loaded by the Python interpreter as a separate module and then the `script_exec()` function is called. I.e. this function is the main function of the script and the logic should be implemented there.

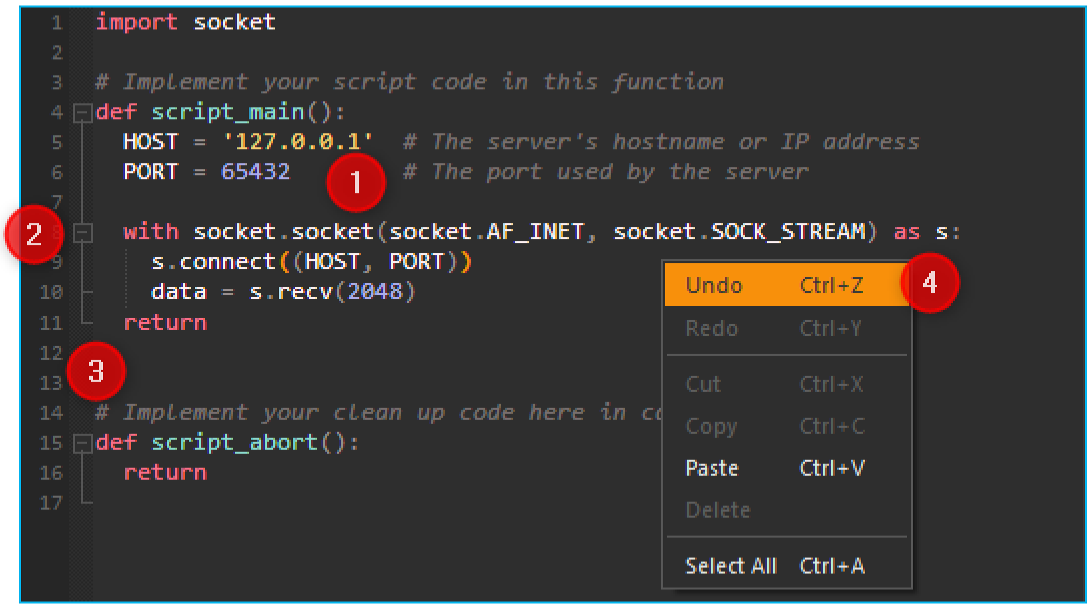
HINT. All options you have in the Python console to access device objects and application objects are also available in the script function.

The Python interpreter can only execute one Python script at a time. Parallel execution is not possible. If you use Python scripts in parallel sequences, then the scripts are executed one after the other, i.e. a parallel execution branch blocks until the execution of a script in another branch is completed. This is another reason why you should keep the execution time of scripts as short as possible. If you use non-blocking Python scripts with short execution times, "almost-parallel" execution in parallel sequences is possible.

IMPORTANT. Parallel execution of multiple Python functions is not possible. If Python scripts are used in parallel sequences, they are executed one after the other.

23.6.2 Python Script Editor

The Python Script function has a Python code editor to assist you in writing Python code.



The editor has the following features

- Syntax highlighting for Python code 1
- a simple code completion
- Code Folding 2
- Line numbers 3
- Undo / Redo functionality 4

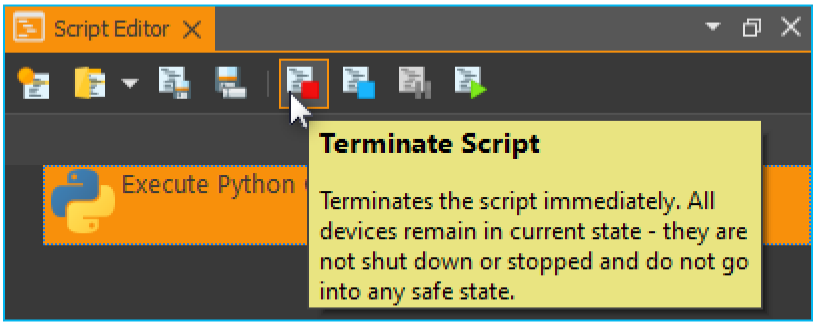
Some functions of the editor are available via the context menu, other functions are available via keyboard shortcuts. Here are some of the functions:

- Increase font size – **Ctrl +**
- Decrease font size – **Ctrl -**
- Reset font size to default – **Ctrl+0**
- Indent selected code block: **Tab**
- Unindent selected code block: **Shift+Tab**
- Undo: **Ctrl+Z** or context menu
- Redo: **Ctrl+Y** or contexte menu

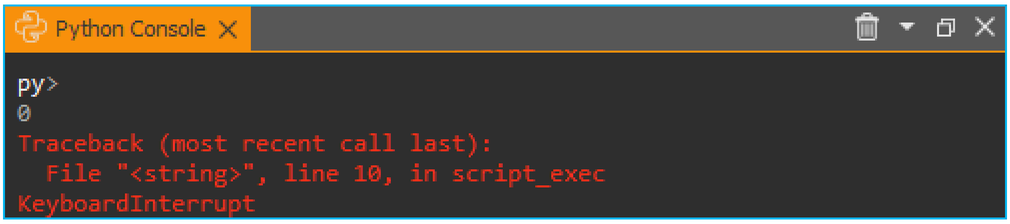
IMPORTANT. Editing the Python source code is only possible when the script is not running. Once the script has been started, editing of the source code is disabled. In case of an error you have to terminate the script via the *Terminate Script* button before you can edit the Python code.

23.6.3 Handle script termination - script_abort()

If the running script is terminated via the *Terminate Script* button of the Script Editor (see figure below), then the execution of `script_exec()` is interrupted and the `script_abort()` function is executed:



In case of such an abort, you will see a corresponding error message in the Python console:



If you want to act on the abort of the script, e.g. to release resources or to inform the user, you can do this in the `script_abort()` function. If you want to access data or objects (e.g. file handles, sockets or similar) in the `script_abort()` function which you have used before in the `script_exec()` function, you can do this via global variables. The following script shows a corresponding example. The `script_abort()` function outputs the number of loop cycles that were executed before the script was aborted. Both functions access the global variable `counter`:

```
import time

counter = None

def script_exec():
 global counter
 for i in range(1000):
  counter = i
  print(i)
  time.sleep(1)
 return

def script_abort():
 global counter
 print("script_abort() after ", counter, " loops")
 return
```

23.6.4 Implementation of the function logic in script_exec()

When implementing the script in `script_exec()` you should be careful not to use blocking functions or blocking waits. The Python interpreter can be interrupted via the *Terminate Script* button only after the execution of the current Python statement. If the current statement is a blocking function call, e.g. `time.wait(10)`, the interpreter can only be interrupted after 10 seconds when the wait call has finished. Therefore, always add a timeout to blocking function calls.

IMPORTANT. Do not use blocking function calls to avoid blocking termination of script execution. Always add a timeout to blocking function calls.

In the following example, the call to `socket.recv()` in line 8 is blocking. I.e. the call returns only when data has been received. As long as no data is received, the function blocks and the script cannot be aborted cleanly:

```
import socket

def script_exec():
 HOST = '127.0.0.1'  # The server's hostname or IP address
 PORT = 65432        # The port used by the server
 with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
  s.connect((HOST, PORT))
  data = s.recv(2048)
 return
```

To fix this problem the call to `socket.recv()` should have a timeout. This has been implemented in the following example using the `socket.settimeout()` function:

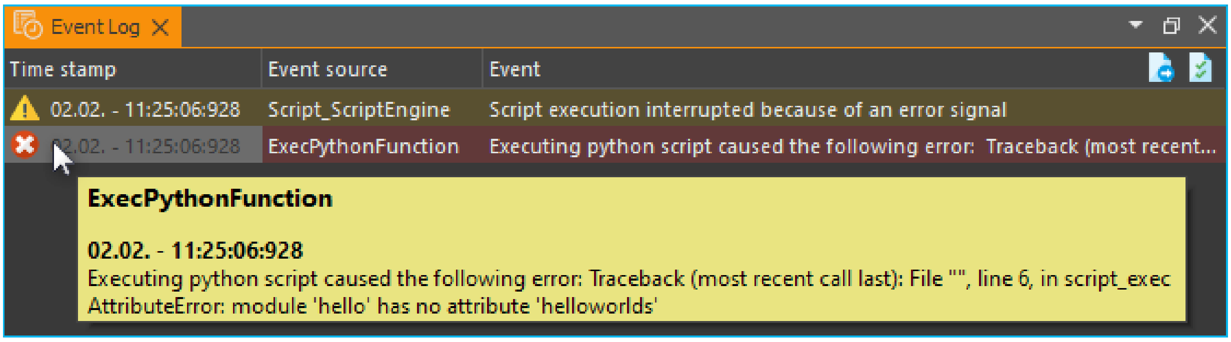
```
import socket

def script_exec():
 HOST = '127.0.0.1'  # The server's hostname or IP address
 PORT = 65432        # The port used by the server
 with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
  s.connect((HOST, PORT))
  s.settimeout(0.5)
  try:
```

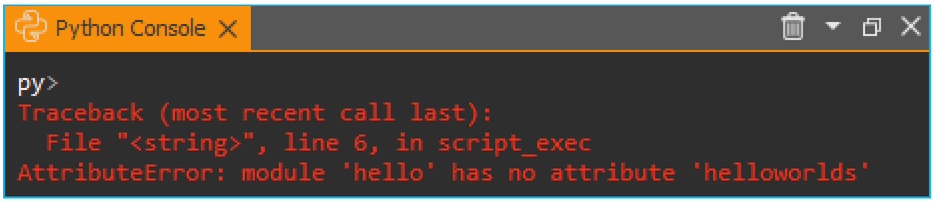
```
     data = s.recv(2048)
    except socket.timeout as err:
     print(err)
   return
```

23.6.5 Script execution errors

If errors occur during the execution of a script, you will see them in the *event log* and in the *Python console*. If you hover over the error message in the event log, you will see a hint window with details:



In the Python console, the error message is displayed to you as red error text::



In the error message you will also get the information in which line of the script an error occurred. This will help you to find and fix the error in the script editor.

IMORTANT. Editing the Python source code is only possible when the script is not running. Once the script has been started, editing of the source code is disabled. In case of an error you have to terminate the script via the *Terminate Script* button before you can edit the Python code.

23.6.6 Using custom modules

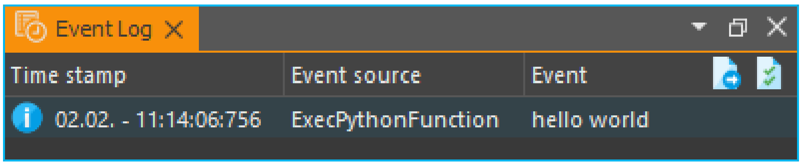
If your script contains very complex and extensive logic, you may want to offload the code to an external module and include it using the import functionality (see section Import custom modules). You can then call the functions of the imported module from within `script_exec()`.

In the following example we import our own module `hello` and call its function `helloworld()`.

```
import hello

def script_exec():
 hello.helloworld()
 return
```

The output of the script appears both in the event log:



and in the Python console:

If you make changes to the external module after the module has already been imported, then these changes will not be available in your Python script in the application. This is the normal way the Python interpreter works - once a module has been imported, it will not be imported again. As a test, add the `hellouniverse()` function to your own `hello` module:

```
def hellouniverse():
  print("hello universe")
```

Now call the new function from your application script. When running, you will receive the information that the `hellouniverse` function is not available.

To avoid restarting the application after changing the external module, you can explicitly request a reload in your script using the reload() function from the importlib module. To do this, modify your Python script in the application as follows:

```
import hello
from importlib import reload

def script_exec():
 reload(hello) # trigger explicit reload of hello module
 hello.helloworld()
 hello.hellouniverse()
 return
```

The script is now executed without errors.

HINT. Use the importlib.reloead() function when making changes to external modules after you have already imported them via import into your Python script in the application.

23.6.7 Accessing script variables

To be able to pass the results of calculations in Python code to the running script or to react to values from the script, it is necessary to access script variables. To do this, import the cetoni_elements module (see section The cetoni_elements module).

Once you have imported the module, you can access script variables using the ScriptEnv.getVar(), ScriptEnv.setVar() and ScriptEnv.setVars() functions. The following example shows how to read the script variable $Flow, increment it by 2 and then store the calculated value back into the script variable:

```
from cetoni_elements import *

def script_exec():
 flow = ScriptEnv.getVar("$Flow")
 print(flow)
 flow = flow + 2;
 ScriptEnv.setVar("$Flow", flow)
```

```
 print(flow)
 return
```

IMPORTANT. Remember that script variables in Python script always start with a dollar sign - $VarName.

Script variables can also be used to access devices. Script variables can store device references. By reading the device reference and assigning it to a Python variable, the device object can be accessed. In the following example, the `$Pump1` script variable contains a device reference of the *Nemesys_S_1* pump that was assigned to the variable in the *Create Variable* function. The device reference is read from the variable and assigned to the Python variable `pump`. Now device functions can be accessed via the `pump` variable. In this example, the syringe is emptied via `pump.emptySyringe()`:

```
from cetoni_elements import *

def script_exec():
 pump = ScriptEnv.getVar("$Pump1")
 print(pump)
 pump.emptySyringe()
 return
```

It is also possible to create lists in Python and to store them in a script variable. The following example creates a list of 4 values and stores them in the script variable `$Positions`:

```
from cetoni_elements import *

def script_exec():
 ScriptEnv.setVar("$Positions", [0, 3.5, 12, 7])
 return
```

Instead of simple values, it is possible to create lists of devices and store them in variables. In the following example a list is created which contains the two digital inputs of the first *Nemesys S* pump. This list is stored in the script variable `$DigitalInputs`.

```python
from cetoni_elements import *


def script_exec():
 di1 = ScriptEnv.getDevice("Nemesys_S_1_DigIN1")
 di2 = ScriptEnv.getDevice("Nemesys_S_1_DigIN2")
 ScriptEnv.setVar("$DigitalInputs", [di1, di2])
 return
```

23.7 Example scripts

23.7.1 Barcode Scanner

The following example script shows how the camera support of CETONI Elements can be used to implement a barcode scanner using the pyzbar module:

```
from pyzbar import pyzbar
import time
import qimage2ndarray
from cetoni_elements import *

# Main script function
def script_exec():
 barcode = None
 camera = ScriptEnv.getObject('Qmix_CAM_1')
 qimage = camera.capturedPreviewImage()
 frame = qimage2ndarray.rgb_view(qimage)
 barcodes = pyzbar.decode(frame)
 for b in barcodes:
  barcode = b.data.decode('utf-8')
  break

 # output
 if barcode is not None:
  ScriptEnv.setVar('$Barcode', barcode)
 else:
  ScriptEnv.setVar('$Barcode', 0)
 return
```

By importing the `cetoni_elements` module, the CETONI Elements camera can be accessed and an image can be captured:

```
camera = ScriptEnv.getObject('Qmix_CAM_1')
qimage = camera.capturedPreviewImage()
```

The qimage2ndarrays module helps us to convert the captured image in *QImage* format into a *numpy.ndarray*, which is required by the pyzbar module.

```
frame = qimage2ndarray.rgb_view(qimage)
```

Now we can use the pyzbar module to decode the barcode:

```
 barcodes = pyzbar.decode(frame)
 for b in barcodes:
  barcode = b.data.decode('utf-8')
  break
```

In the last step the barcode is stored in the script variable `$Barcode` with the function `ScriptEnv.setVar()` so that it is available in the script and can be evaluated further.

```
if barcode is not None:
  ScriptEnv.setVar('$Barcode', barcode)
 else:
  ScriptEnv.setVar('$Barcode', 0)
 return
```

24 SiLA2 Add-on

24.1 Installation

SiLA 2 add-on is not included in the standard CETONI Elements installation package, but must be installed additionally as an add-on. The versions of SiLA 2 Plugin and installed CETONI Elements software should match. For example, if you want to install SiLA 2 Plugin with version number 20210707, CETONI Elements software should be installed with version number 20210707.

IMPORTANT. The version number of the CETONI Elements software and the SiLA 2 add-on should match. The version of the SiLA 2 add-on must never be greater than the version of the CETONI Elements software. The greater the difference between the version numbers, the greater the risk of problems or errors occurring.

Please close all other programs before installation.

Start the file CETONI_Elements_SiLA_64bit_Setup.exe for installation. The installation wizard will then guide you through the installation of the software and hardware drivers.

IMPORTANT. On Windows, you must be logged in with administrator privileges to install the hardware drivers.

24.2 SiLA 2 Basics

SiLA 2 (Standardization in Lab Automation) is a communication standard for laboratory devices. The primary goal of SiLA is to create an international, open connectivity standard in laboratory automation that allows devices from different manufacturers to communicate with each other in one application.

24.2.1 Client-Server concep

Communication is client-server based and uses established network standards. A SiLA server is usually provided by a device while a SiLA client is usually a control software (e.g. a LIMS or CETONI Elements). Since a SiLA server represents a device in most cases, the terms SiLA server (or just server) and device are used synonymously in this document.

The client always acts as the master by initiating the connection to a server. The connection is established via a TCP/IP network. A central component of SiLA 2 is the discovery of SiLA servers by SiLA clients in a local network. A client can use the SiLA Server Discovery mechanism to determine all available servers and their functionalities.

In SiLA 2, these functionalities of a device are called features. Features are another central component of SiLA 2. They describe which data a server provides and which actions the device can perform.

24.2.2 Commands and Properties

In SiLA 2, the data are called *properties* and the actions are called *commands*. Properties and commands can be both *unobservable* and *observable*.

Unobservable properties are static data of a server that do not change during the runtime of a server (e.g. the number of possible switching positions of a valve).

Observable properties are dynamic data of a server that can change during the runtime of a server and about which clients should be informed (e.g. the current flow rate of a syringe pump). Clients can subscribe to observable properties so that they are notified of any changes to the property.

Unobservable commands are actions that can be performed on a device and that take very little time (e.g., to change settings such as the syringe parameters of a pump).

Observable commands are actions that can potentially take a long time to execute on the device (e.g., fluid dispensing). Here, a client can subscribe to the so-called *Execution Info* to be notified about the progress of the command execution.

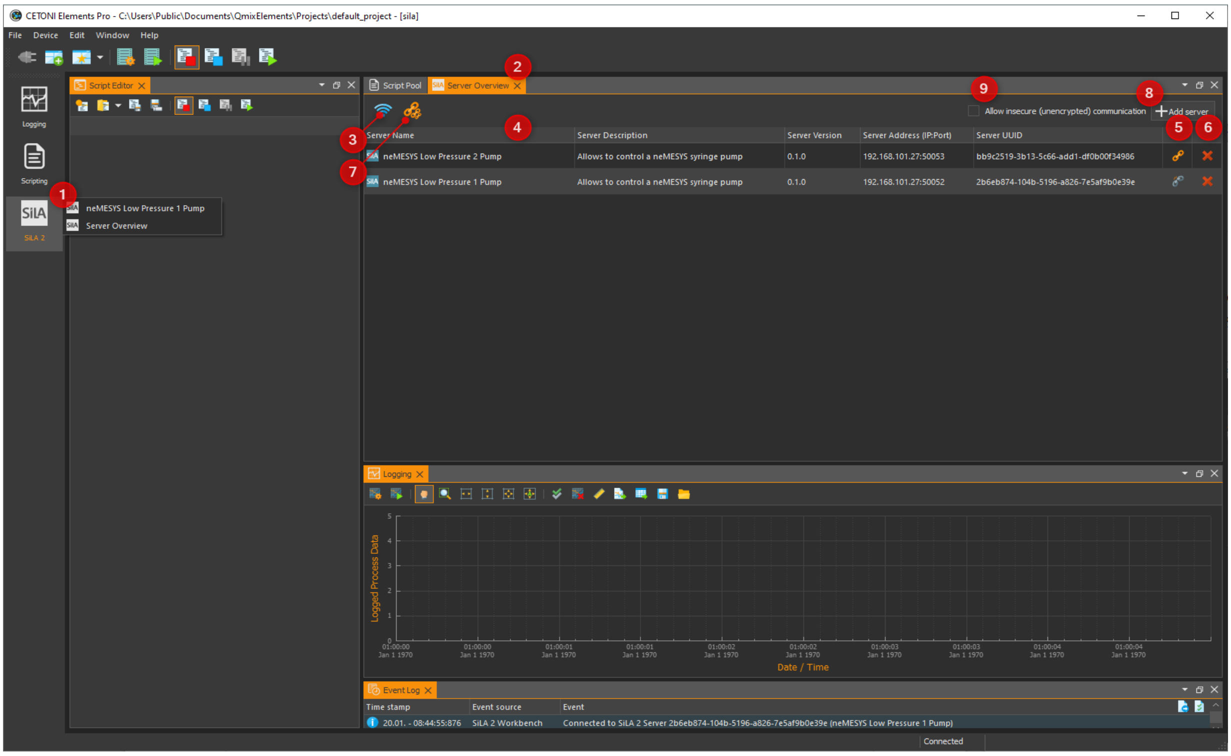
Commands (unlike properties) can have parameters and return values. Parameters can be used to influence the execution of a command (e.g. to specify the desired flow rate and the volume to be dosed for a dose). Return values can be used to transmit the result of a command execution to the client (e.g. the acquisition data of a spectrometry).

Additional metadata may be required to read a property or execute a command. These behave similarly to command parameters in the sense that depending on the metadata the value of a property or the effect of a command execution can vary.

TIP. A comprehensive introduction to the SiLA standard is available on the SiLA homepage: https://sila-standard.com/standards/

24.3 Introduction to the SiLA Add-on

By clicking the SiLA 2 button and selecting *Server Overview* 1 in the sidebar, you switch to the SiLA 2 add-on (see figure below).



1 SiLA 2 button for display of *Server Overview* and connected SiLA 2 servers.

2 Tabs for selecting the *Server Overview* or a SiLA 2 server

3 Button to perform a network scan to find available SiLA 2 servers

4 Display of all SiLA 2 servers found in the network or added manually

5 Button to connect or disconnect from a SiLA 2 server

6 Button to delete the server from the list

7 Button to connect to all servers in the list

8 Button for adding a SiLA 2 server manually

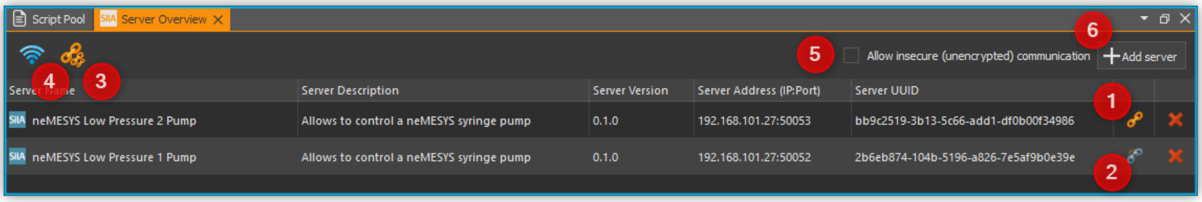
9 Selection box for allowing unencrypted connections.

You can use the sidebar button 1 or the tabs 2 to switch between the currently connected SiLA 2 servers and the *Server Overview*. A tab is created for each server.

24.4 Connecting to a SiLA 2 server

24.4.1 Connecting to automatically detected servers

To connect to a SiLA 2 server that was found in the local network via SiLA Server Discovery, simply click on the *Connect* icon 1 in the corresponding row in the *Server Overview*. After the connection has been successfully established, the *Disconnect* icon 2 appears instead of the *Connect* icon. By clicking on this icon you can disconnect from this device again.



TIP. The *Connect-To-All* 3 button allows you to connect to all devices at once, rather than having to connect to each device individually.

TIP. If the software did not find one or more of your SiLA devices, it may help to restart the servers and run a new network scan 4.

TIP. If you close the software while still connected to one or more SiLA servers, it will attempt to restore those connections the next time you start the software.

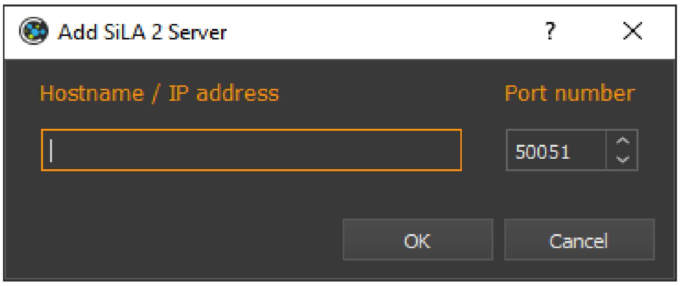
IMPORTANT. If you delete a server with an active connection to the software from the *Server Overview* list, then the connection is automatically disconnected.

Normally, communication between SiLA server and client is encrypted. However, if one of your devices does not provide encryption, the connection attempt will fail with an error message in the event log. For local testing, you can still connect to this device if you check *Allow insecure (unencrypted) connection* 5.

ATTENTION. You should never establish an unencrypted connection to a device that is outside your local network, otherwise all communication between the device and the software can be read by third parties.

24.4.2 Manually add and connect to a server

If a server in your local network is not found automatically, or if you want to connect to a server outside your local network, you must add this server manually. To do this, click the *Add server* 6 button (see figure above). The following dialog box appears.

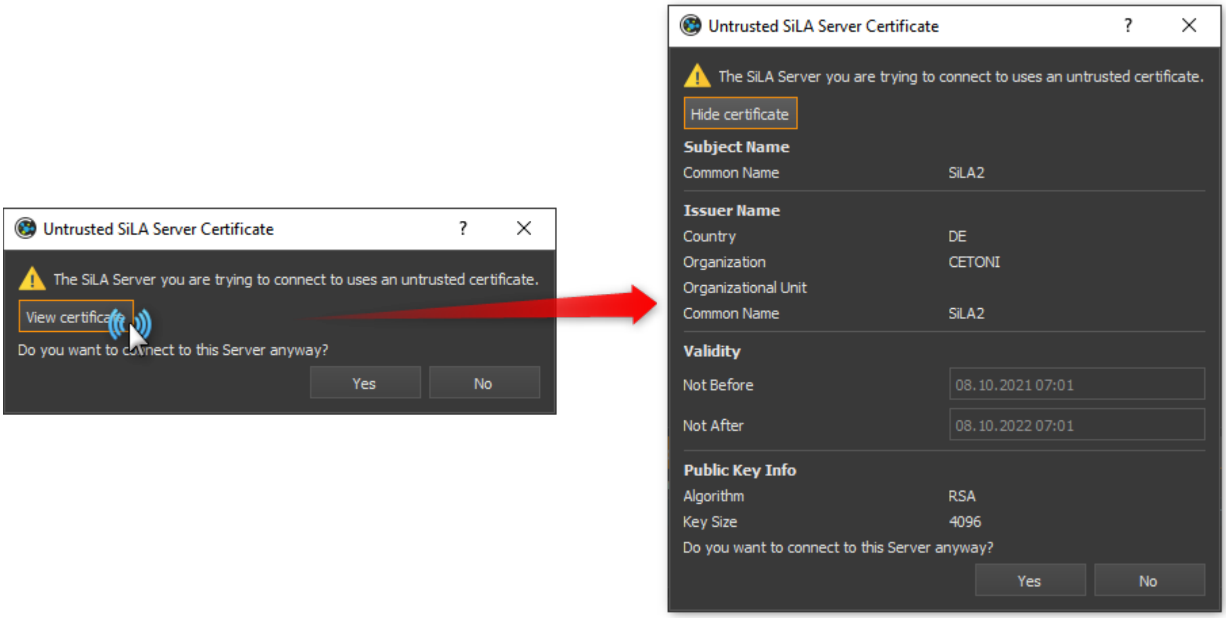


Enter either the host name or the IP address of your device and the port on which the SiLA server is running. Then click *OK*.

The software will now try to connect to this server. If this is successful, the server will appear in the *Server Overview* list. If the connection cannot be established, you will find an error message in the *Event Log* with information about why the connection failed.

24.4.3 Accept untrusted certificates

SSL certificates are used to encrypt the communication between server and client, which are also used to encrypt the communication on the Internet. Usually, these certificates are signed by a trusted certificate authority. If you try to connect to a server that uses a self-signed certificate, the following dialog box appears:



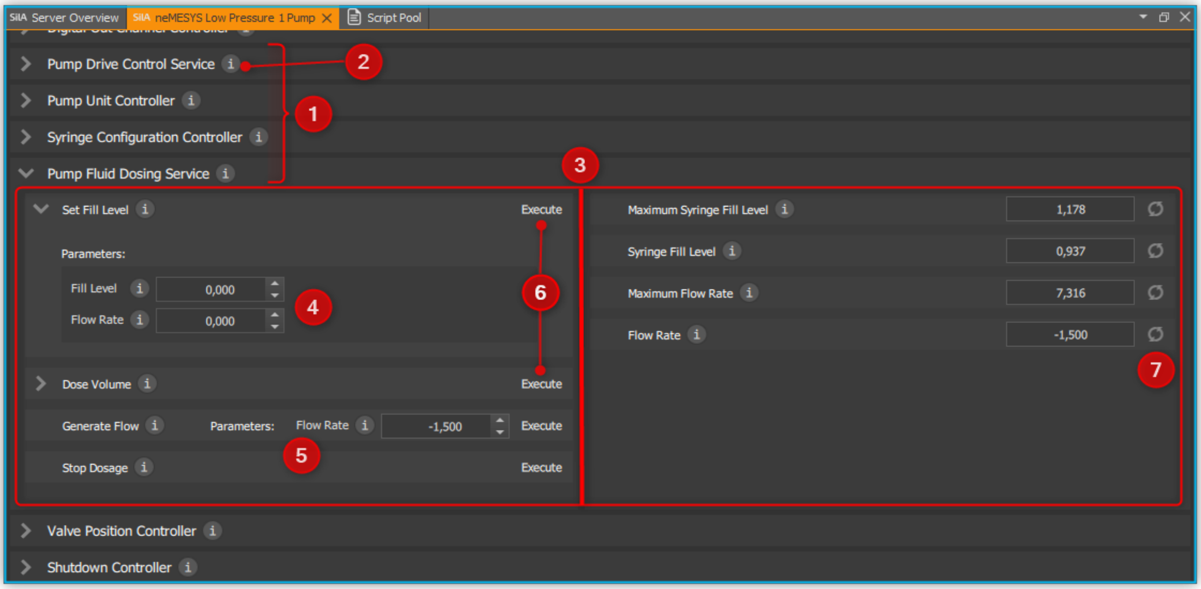
Here you can decide whether or not to connect despite the untrusted certificate. By clicking *View Certificate* you get more information about the certificate. There you can see, most importantly, which institution created and signed the certificate.

If you trust the certificate, click *Yes* in the dialog box to continue the connection, otherwise click *No*. In this case, the connection will be aborted.

24.5 Control of a SiLA 2 server

After you have successfully connected to a SiLA server, you can control it via a generic interface. This interface looks the same for any type of SiLA server, that is, for any device regardless of the type or manufacturer of the device. While this interface allows you to control every aspect of a SiLA server, it cannot be used to create complex workflows. For this purpose, the script functions provided by the plugin, which are presented in Section 24.6, are more suitable.

Open the generic interface by selecting the desired server with the SiLA 2 button in the sidebar (see section Introduction to the SiLA Add-on). You will first see all SiLA 2 features provided by the server 1. If you move the mouse pointer over one of the **i** icons, a tooltip with the description of the feature 2 will appear. By clicking on the feature name or the arrow to the left of the name, you can expand the feature. Now you can see all available commands and properties of this feature 3. Commands can be found on the left side, properties on the right side.

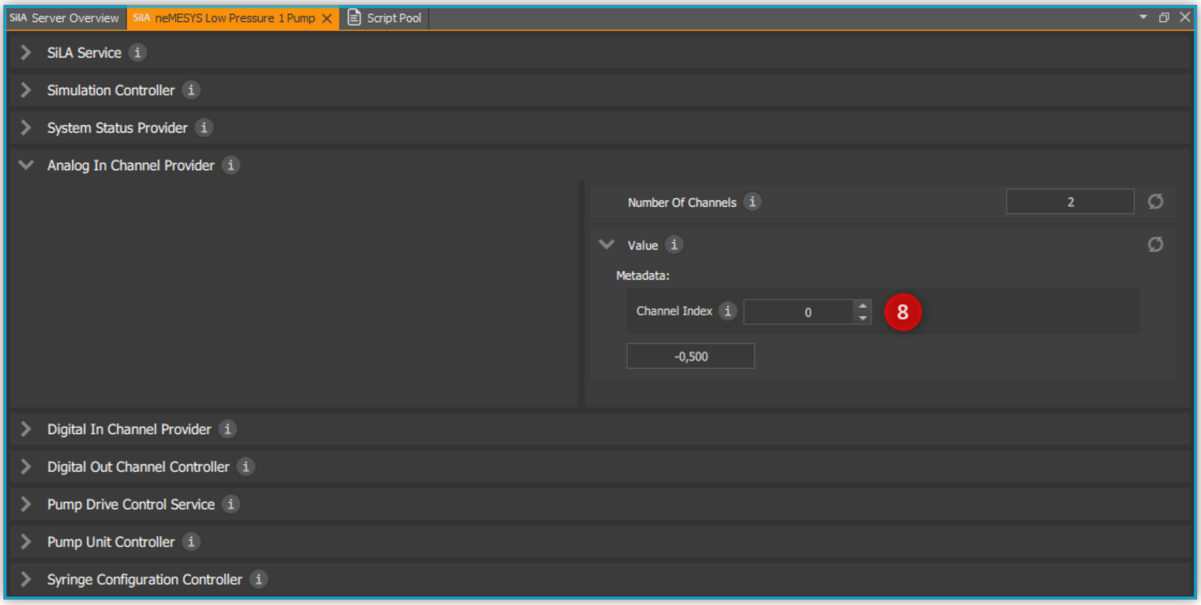


If a command has parameters, metadata or return values, it can also be expanded to enter the parameters or metadata and display the return values 4. If a command has no parameters or only one, then the command cannot be expanded and at most one input field is displayed directly next to the command name 5. To execute the command, click the *Execute* button 6.

Normally, properties cannot be expanded. Here, only a field with the current value of the property is displayed. To the right is a button for re-polling the property 7. Clicking the button triggers one of the following two actions:

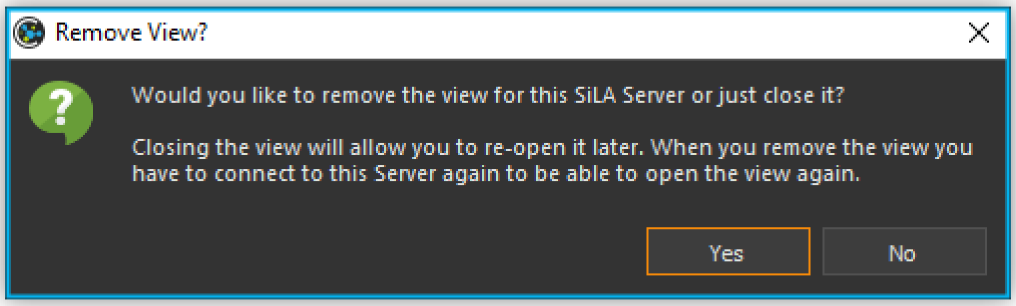
- If the property is *Unobservable* (see Section 24.2.2), then it is simply queried again by the server.
- If the property is *Observable*, the subscription running in the background is canceled and a new subscription is started. This may be necessary, for example, if the subscription was automatically canceled due to an error.

It is possible that a property requires metadata. In this case, the property can be expanded like a command and the input fields for the metadata 8 appear. After you have entered the metadata, you must query the property again or start a new subscription so that this data is sent to the server.



IMPORTANT. When the generic interface is opened for the first time, all unobservable properties are queried once and subscriptions are automatically started in the background for observable properties. However, this only works if the property does not require metadata. In this case you have to enter the metadata first and then query the value yourself or start a subscription yourself.

If you terminate the connection to a server or the connection is automatically terminated, e.g. due to a network error, the interface remains open, but all input fields and buttons are locked. If you then want to close the tab for the affected server, the following dialog box appears:

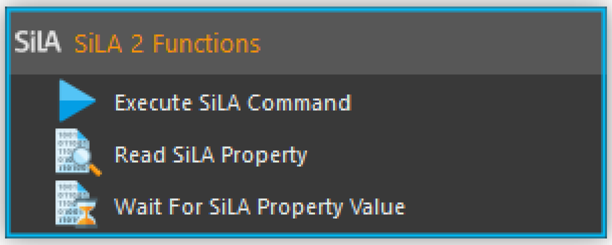


Clicking on *"Yes"* will close the interface (*View*) for this server and recreate it when the same server is reconnected later. By clicking on *"No"*, the view is retained and is only hidden. This allows you to reopen the view later, for example to see certain parameters or command responses. Furthermore, this view is automatically reused when the connection to the server is re-established.

24.6 SiLA 2 Script Functions

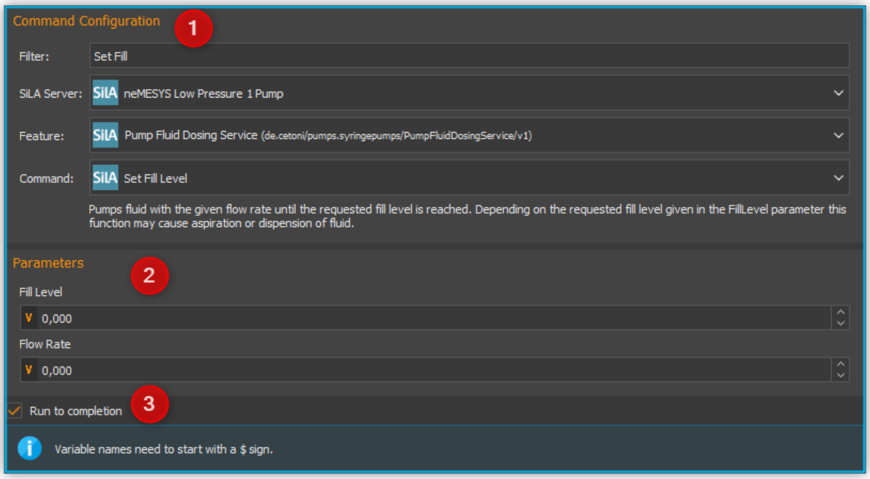
24.6.1 Introduction

The SiLA 2 plugin contains various script functions for script-driven execution of commands and querying of properties.



24.6.2 Execute SiLA Command

With this function you can execute a SiLA command and store the result of the execution in a variable.



You can set the command to be executed 1, as well as the required parameters and metadata 2 in the configuration area.

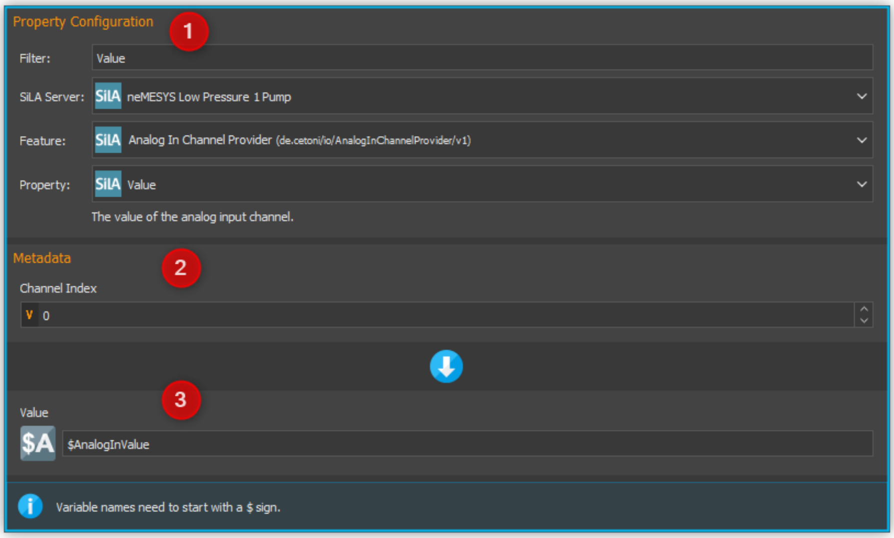
In addition, you can enable or disable the *Run to completion* 3 option for *Observable Commands*. If *Run to completion* is enabled, script execution will not continue until the command has been fully executed. If this option is not active, the command is only started and then the next script function is processed immediately.

If the command has return values, you can also specify variables 4 in which these values are to be stored.

TIP. All SiLA script functions support the use of variables. This means that you can enter variables in all input fields marked with a colored V in the configuration area.

24.6.3 Read SiLA Property

With this function you can query a property and store the value into a variable.

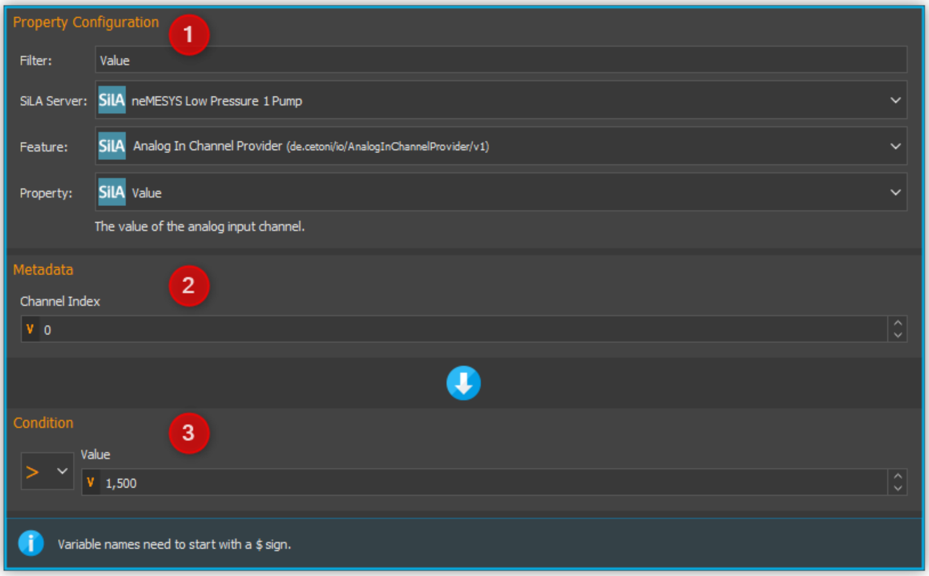


You can set the property 1 to be queried and the possible required metadata 2 in the configuration area.

Enter the name of the variable that is supposed to store the value of the read property in the field below 3. If the property returns a structure of several values, corresponding fields will appear for each element of the structure, in which you can enter the individual script variables.

24.6.4 Wait For SiLA Property Value

This function allows you to make the script execution wait for a specific property to meet a defined condition. The function will not continue until the condition is met.



In the configuration area, first select the property 1 you want to check and the optionally required metadata, as described for the Read SiLA Property function.

IMPORTANT. The SiLA property must be Observable to be used in this script function.

In the *Condition* area, you configure the check condition. To do this, first select a comparison operator 2 and then enter the value 3 to be compared with. In the value field 3 you can also use variables to set the check condition.

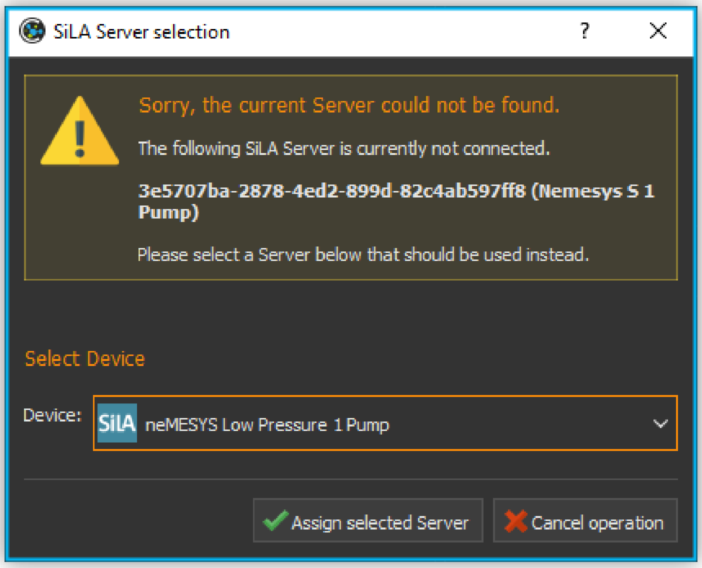
TIP. To quickly find a specific feature, command or property, you can enter the name or part of the name in the filter input field. The selection boxes for the feature, command or property will then only contain entries that match the filter entered.

24.6.5 Replace SiLA server

A SiLA server is always uniquely identified by its UUID (Universally Unique Identifier). When you create a script function for a specific server, this UUID is stored in the background. As a result, scripts are always bound to the SiLA server for which they were written. Now it can be possible that you want to use e.g. a script that you have written for a Nemesys low pressure pump with another low pressure pump or that a certain SiLA server had to be exchanged and therefore its UUID has changed. In these cases, you can replace the configured SiLA servers in your scripts with the new servers.

To do this, load the script in where you want to replace the servers into the script editor. Then select a SiLA 2 function where you want to change the server. The configuration interface of the function should be grayed out because the corresponding SiLA server is not connected. In addition, you will also see a warning message 1.

Now click on the icon for changing the SiLA server 2. The following dialog box opens:



Here you can now select one of the connected SiLA servers to be used instead of the configured server.

IMPORTANT. The new server must provide exactly the same features as the previous server. The selection dialog will inform you if this is not the case.

If you have selected a compatible server, click on "*Assign selected Server*". Now the previous server will be replaced by the selected server in **all** script functions. Afterwards, the configuration area is no longer grayed out and you can edit the function again. When you save the script, the server you just selected will be saved and used the next time you load the script.

24.7 Read SiLA properties via process data identifiers

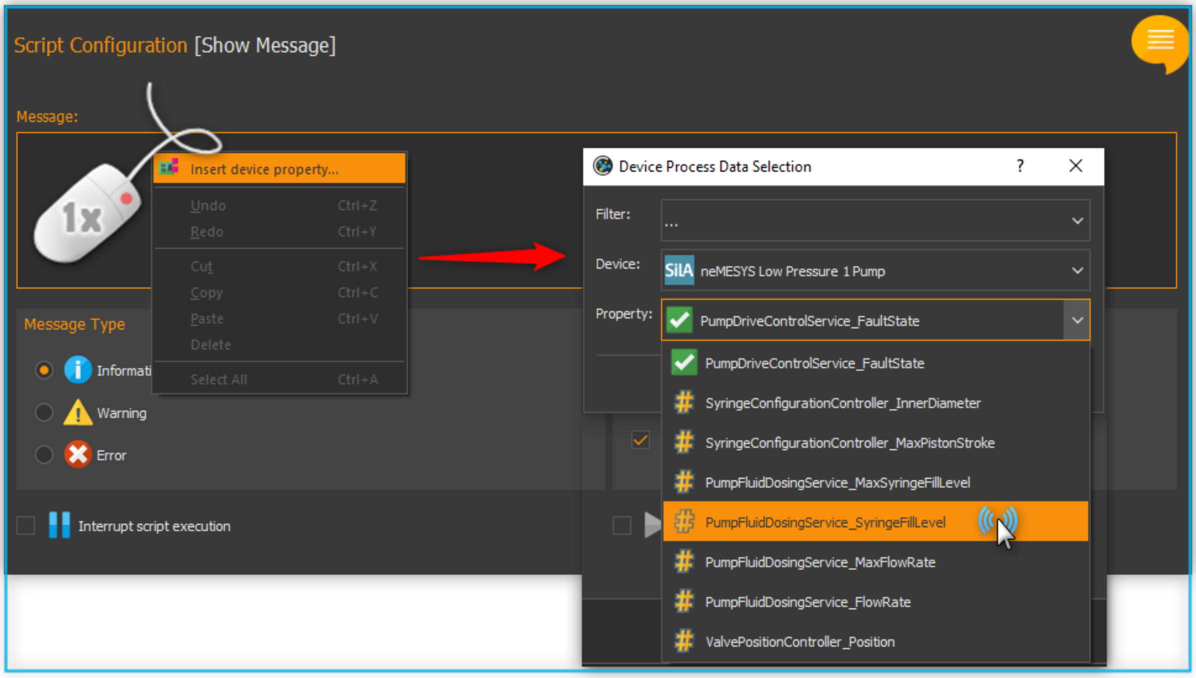
Certain SiLA properties are automatically provided as process data. This makes it possible, for example, to initialize script variables directly via the respective process date or to use this data in the graphical process data logger.

For this purpose, the SiLA Properties must fulfill the following requirements:

- the property must be observable.
- the property must not require metadata.
- the data type of the property must be a numeric (integer or real) or Boolean data type.

24.7.1 Use in other script functions

SiLA process data can be accessed in exactly the same way as other process data from other devices:



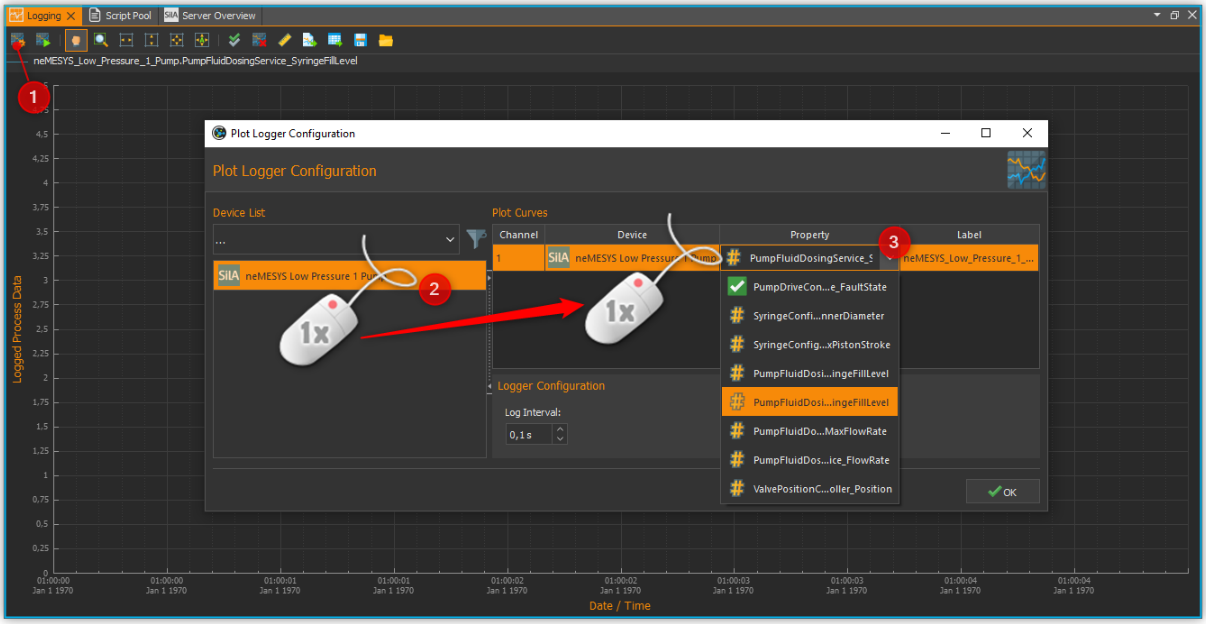
The selected process data identifier is now entered in the input field. These also have a specific form based on the usual process data identifier:

$$ServerName_ServerUUID.FeatureIdentifier_PropertyIdentifier

Each identifier starts with two dollar signs and a dot separates the unique device name from the unique process data name. The device name is derived from the name of the SiLA server as well as its unique ID (UUID, Universally Unique Identifier) and the name for the process data is derived from the name of the feature and the respective property. The entire process data identifier must not contain any spaces or other special characters.

24.7.2 Use in graphical process data logger

SiLA process data can be used in the graphical process data logger as known from other devices:



First, open the Plotlogger configuration ❶. There, drag and drop the desired SiLA device into the list of plot curves ❷. Then double-click on the device property cell to select the SiLA property to be recorded ❸.