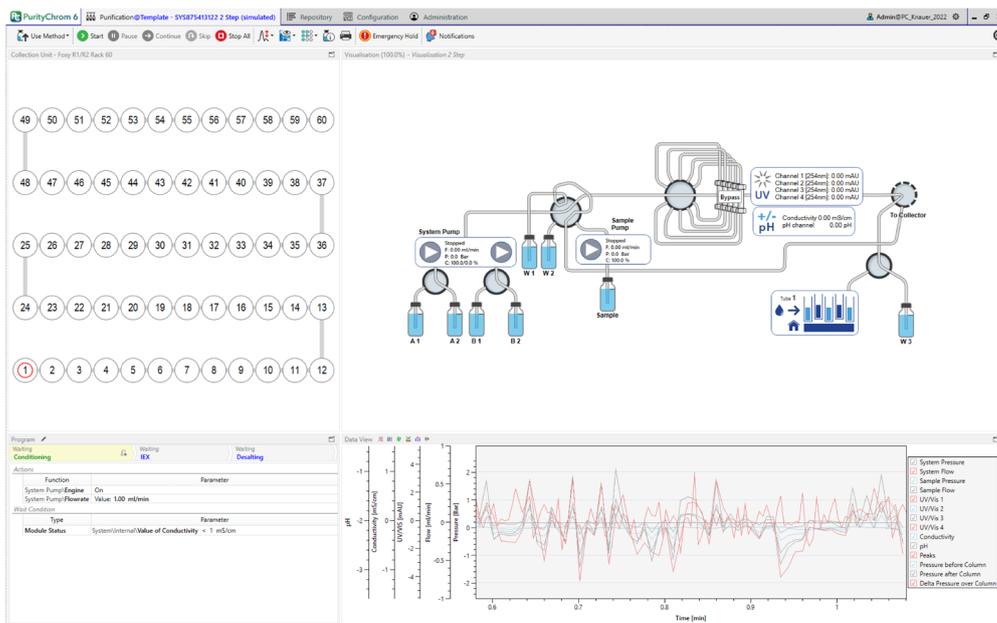




# PurityChrom 6

## Chromatography Software Instructions



# SOFTWARE



**Note:** For your own safety, read the instructions and follow the warnings and safety information on the device and in the instructions. Keep the instructions for future reference.



**Note:** In case you require this instruction in another language, please submit your request including the corresponding document number via e-mail or fax to KNAUER.

**Support:** Do you have questions about the installation or the operation of your instrument or software?

**International Support:**

Contact your local KNAUER partner for support:

[www.knauer.net/en/Support/Distributors-worldwide](http://www.knauer.net/en/Support/Distributors-worldwide)

**Support in Germany**

(Austria & Switzerland on case-to-case basis):

Phone: +49 30 809727-111 (workdays 9-17h CET)

Fax: +49 30 8015010

E-mail: [support@knauer.net](mailto:support@knauer.net)

**Publisher:** KNAUER Wissenschaftliche Geräte GmbH

Hegauer Weg 38

14163 Berlin

Germany

Phone: +49 30 809727-0

Fax: +49 30 8015010

Internet: [www.knauer.net](http://www.knauer.net)

E-mail: [info@knauer.net](mailto:info@knauer.net)

**Version information:** Document number: V2670

Version number: 1.1

Release date: September 1, 2023

Original edition

The information in this document is subject to change without prior notice. For the latest version of the instructions, visit our website: [www.knauer.net/library](http://www.knauer.net/library).



**Copyright:** This document contains confidential information and may not be reproduced without written consent of KNAUER Wissenschaftliche Geräte GmbH.

© KNAUER Wissenschaftliche Geräte GmbH 2023

All rights reserved.

# Table of contents

<b>1. Introduction</b>	<b>1</b>
<b>2. Software installation</b>	<b>2</b>
2.1 Recommended minimum system requirements	2
2.2 Configuration of the installation.	2
<b>3. Main Window</b>	<b>10</b>
<b>4. Application part: Purification.</b>	<b>11</b>
4.1 Introduction to the user interface	11
4.2 Control-bar	12
4.3 Classic menu for module control	13
4.4 Collection unit	14
4.5 Program display and online-editing	15
4.6 Visualisation	16
4.7 Data view.	16
4.8 Scheduler.	18
4.9 Communication	19
<b>5. Application Part: Repository</b>	<b>20</b>
5.1 Introduction to the user interface	20
5.2 Editor: Method	21
5.2.1 Method part: General	22
5.2.2 Method part: Procedure-editor.	23
5.3 Editor: Sequence	31
5.4 Editor: Reconstruction.	33
5.4.1 General overview.	35
5.4.2 Integration	37
5.4.3 DAD spectrum	40
5.4.4 Documentation	41
5.4.5 Documentation snapshots	42
5.4.6 Status recording.	42
5.4.7 Signature history	43
5.5 Editor: Macro	44
5.6 Editor: Value fraction	45
5.7 Editor: Compare chromatograms	45
5.8 Data exchange: Mapping & archives.	47
5.9 Editor: Print template.	48
5.9.1 Basics of the editor	48
5.9.2 Design elements	49
<b>6. Application part: Configuration</b>	<b>52</b>
6.1 Introduction to the editor	52
6.2 Configuration part: General	53

6.3	Configuration part: Modules .....	53
6.4	Configuration part: Channels .....	54
6.5	Configuration part: Volume programming .....	55
6.6	Configuration part: Emergency pause .....	56
6.7	Configuration part: Extended solvent setup .....	56
6.8	Configuration part: Valve locking setup .....	57
6.9	Configuration part: System documentation .....	58
6.10	Configuration part: Collection unit .....	58
6.11	Configuration part: Visualisation .....	60
6.11.1	Concept .....	60
6.11.2	Structure of the visualisation editor .....	61
6.12	Configuration part: Collector racks .....	61
6.12.1	Teaching of fraction racks .....	65
<b>7.</b>	<b>Application part: Administration .....</b>	<b>66</b>
7.1	Introduction to the editor .....	66
7.2	User properties .....	66
7.3	Role-properties .....	68
7.4	Security settings .....	69
7.5	Audit trail .....	69
7.6	Installation qualification .....	70
<b>8.</b>	<b>Program settings .....</b>	<b>71</b>
<b>9.</b>	<b>Special features .....</b>	<b>73</b>
9.1	Pump Union for high flow applications .....	73
9.2	Stacked injections .....	74
9.2.1	Stacked injections: General .....	74
9.2.2	Stacked injections via autosampler .....	75
9.2.3	Stacked injections using a sample pump .....	75
<b>10.</b>	<b>How to .....</b>	<b>76</b>
10.1	Recommendations for backups .....	76
<b>11.</b>	<b>Appendix: Concepts, structure &amp; ways of working .....</b>	<b>77</b>
11.1	Installation concept and user administration .....	77
11.2	System-related data: Configuration .....	78
11.3	Working with sequences, methods and result files .....	79
11.4	The runtime environment .....	80
11.5	Driver concept .....	80
11.6	Print Templates .....	81
11.7	Fractionation concept .....	81
11.8	Acquisition of measurement data .....	82
11.9	Evaluating measurement data / peak detection / integration .....	85
11.10	Documentation / Audit-Trail .....	86

---

# 1. Introduction

As a preparative data and control system, PurityChrom® 6 meets all the requirements expected of a modern preparative chromatography software. Thanks to its great flexibility and modular structure, the software can be adapted to the user's system. As a further development of previous software versions, PurityChrom 6 focuses on even simpler user guidance and improved method creation, as well as greater flexibility in terms of instrument configuration and combination options. National and international regulations such as 21 CFR Part 11 are fully complied with, and the software is developed according to GAMP5 guidelines.

The software allows isocratic elution, gradient programming and conveying. Switching valves and fraction collection can be controlled depending on time, volume, detector thresholds or automatic peak detection. Sequence programming can be done in units of time, delivered solvent volume, or column volume.

Pump status, flow rate, pressure, solvent supply, detector signals and valve positions can be displayed on the screen. The entire sequence is shown in one or more system visualisations and in a timetable with the current step highlighted.

During a preparative process, the functions of a method can be adjusted by using a graphical editor. In this way, the user can react on spontaneously occurring events in the process or a retention time shift can be compensated.

The result file of a run contains all recorded chromatograms with integration results, the timetable, a log of the entire run and a fraction table with assignment to fraction collector racks and the associated chromatograms. Furthermore, electronic signatures can be given to the result files. PurityChrom 6 offers the possibility to recover all relevant information from the result file alone to reproduce the procedure.

The structure of this manual has been designed in such a way that a description of each user interface is explained in the first chapters while the reader, who is interested in more details about PurityChrom 6 concepts, is referred to the appendix section.

## 2. Software installation

### 2.1 Recommended minimum system requirements

The system should at least fulfill the following requirements:

Operating System	Windows 10 64 Bit
RAM	8 GB
CPU	Intel Core i5 6th Gen. or comparable
Input Devices	Mouse and Keyboard
Disc space	10 GB
Screen resolution	1920 x 1280 @ 96 DPI
Interfaces	Depending on the devices used (USB/LAN/RS232). For LAN-controlled devices, a second network card and an independent LAN is recommended.  An additional free USB port is required for the license dongle to enable system control in an online system. For offline versions, this is not needed.

All Windows users who are to use PurityChrom 6 need read and write permissions in the working directory of PurityChrom 6 (default C:\ProgramData\PurityChrom6; see also next chapter). In addition, read and write permissions are required in the directories of the respective configuration.

**PrepCon.exe** and **PrepConAnalysis.exe** need permissions in the local firewall if devices are to be controlled via a network interface. Normally, the permissions are automatically created by the installer during the installation in the Windows firewall.

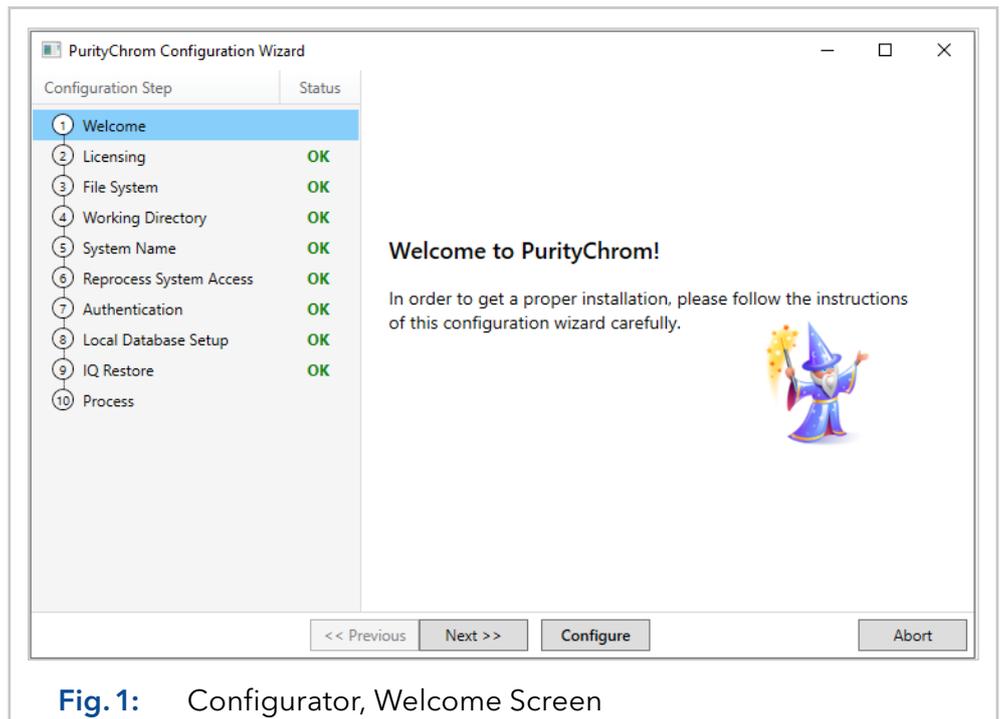
### 2.2 Configuration of the installation

To install PurityChrom 6, the setup has to be run with administrator privileges first. The setup initially only copies the software files into the installation directory and creates shortcuts. After the copying process is completed, the installer starts the PurityChrom configuration tool from the application directory. You can use the configurator to make the settings of the PurityChrom 6 software. You can change these software settings later by manually running the configurator **PrepConConfiguration.exe** (stored in C:\ProgramFiles\PurityChrom6) with administrator privileges.

The PurityChrom 6 configurator is a step-by-step wizard that navigates the user comfortably through the settings. The individual pages of the wizard are explained below:

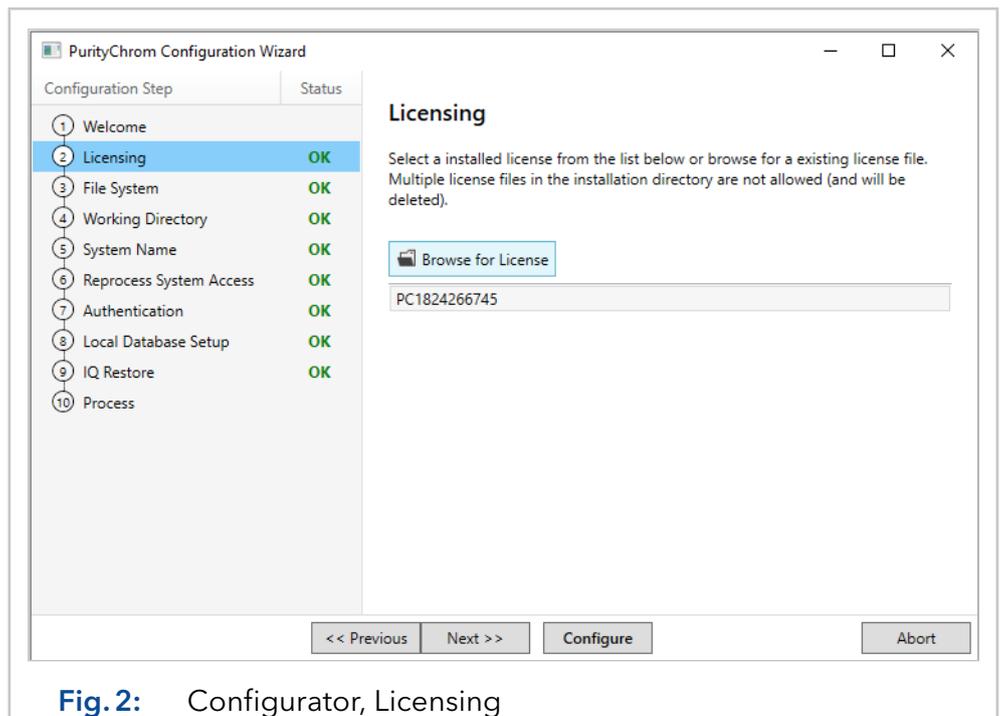
## Step 1: Welcome-Screen

The configurator starts with a welcome screen.



## Step 2: License selection

The licenses which have already been installed in PurityChrom 6 are displayed first. With the "Browse" button, a license from another directory can be selected. The selected license is copied into the application directory at the end.



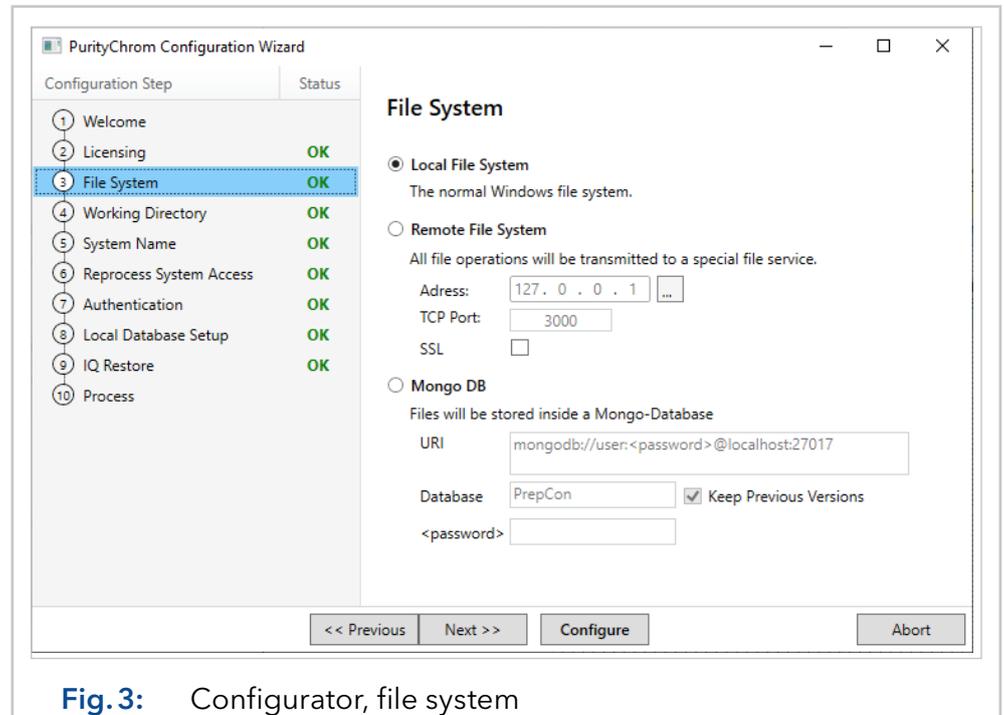
### Step 3: File system selection

Here you can select whether PurityChrom 6 uses the local Windows file system or whether file access is established via a PurityChrom 6 file server or a Mongo database.

Choose the option Local File System to save all data files (e.g. methods, result files, sequences, configurations etc.) locally on the system computer under the working directory chosen in the next step of the installation.

Choose the option Remote File System to save all data files (e.g. methods, result files, sequences, configurations etc.) under the defined external file system. For using this option, a special software package is required to configure the file system. This option helps to avoid unwanted deletion of files via the Windows user account.

Choose the option Mongo DB, to save all data files (e.g. methods, result files, sequences, configurations etc.) on an external server via a Mongo database. For using this option, an external server is required. This option enables secure storage, exchange, and synchronization of data for multiple PurityChrom 6 systems. Furthermore, a data backup can be configured via this external server and deleted files can be recovered.



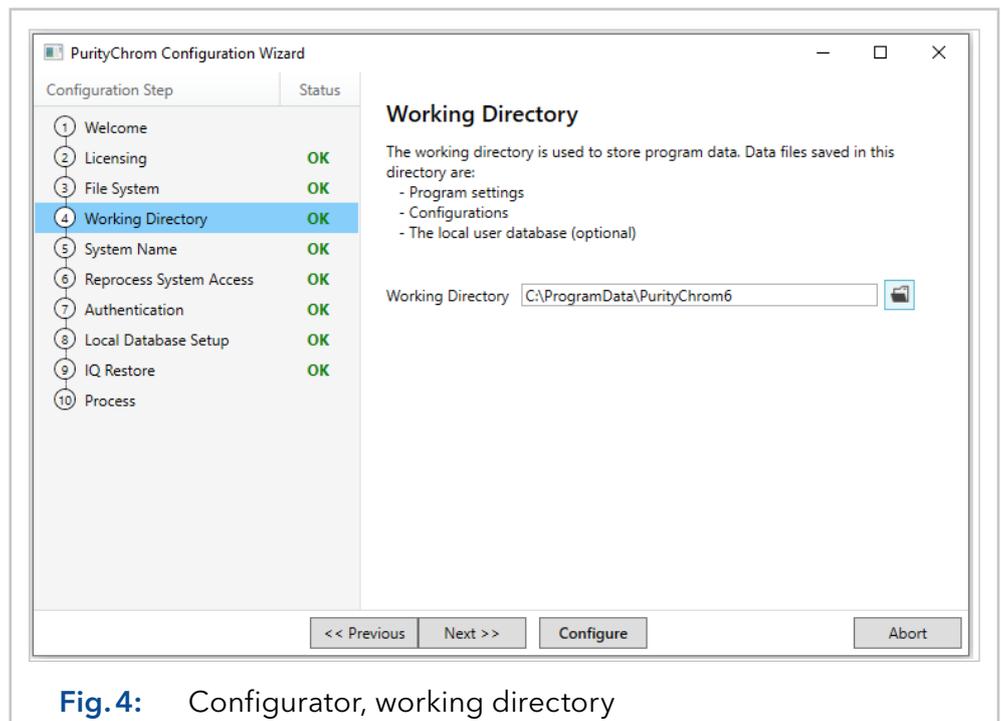
**Fig. 3:** Configurator, file system

### Step 4: Working Directory Setup

For a local file system, the default working directory of PurityChrom 6 is **C:\ProgramData\PurityChrom6**.

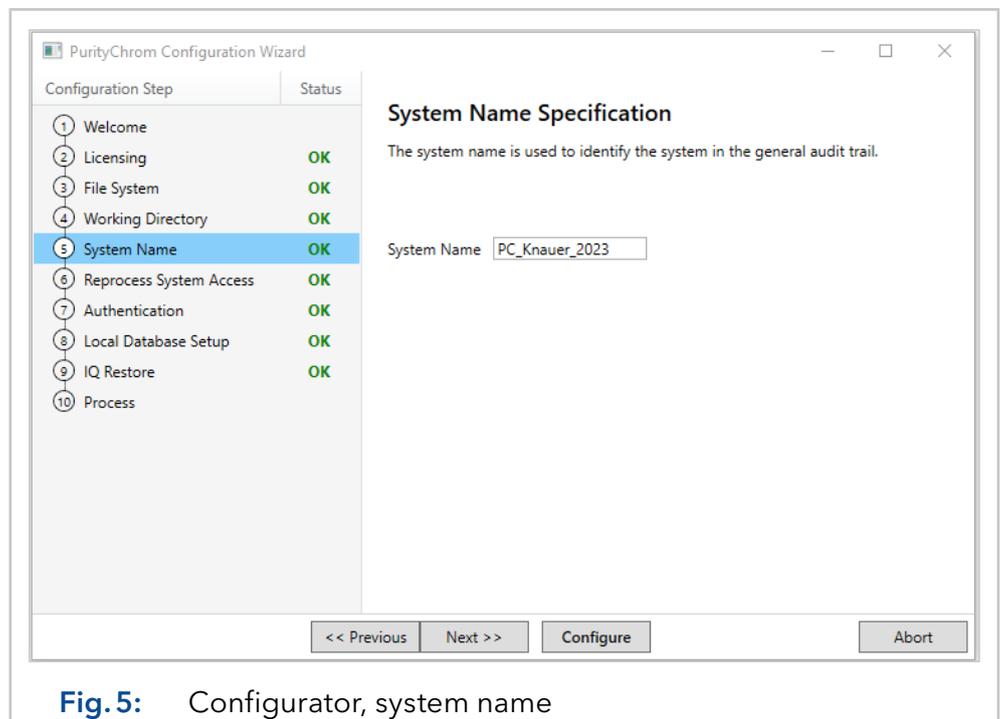


**Note:** All Windows users who are to use PurityChrom 6 need read and write permissions in this directory. The installer sets these permissions automatically for the group "Everyone" during the installation.



### Step 5: System Name

The system name chosen in step 5 of the configurator is logged in the audit trail in case of active user administration and displayed in the header of the software window.

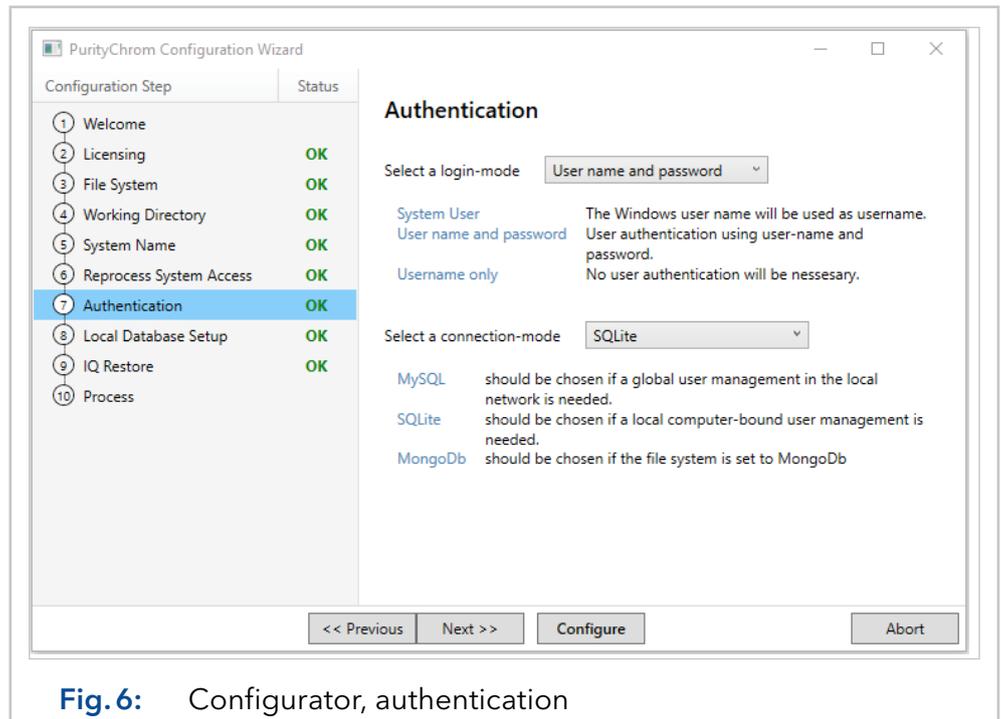


### Step 6: Reprocess System Access

This step is only needed if a remote file system or a MongoDB was chosen in step 3. Add the working directory and the system name of other systems in the same network to enable access to the data of these systems.

### Step 7: Authentication

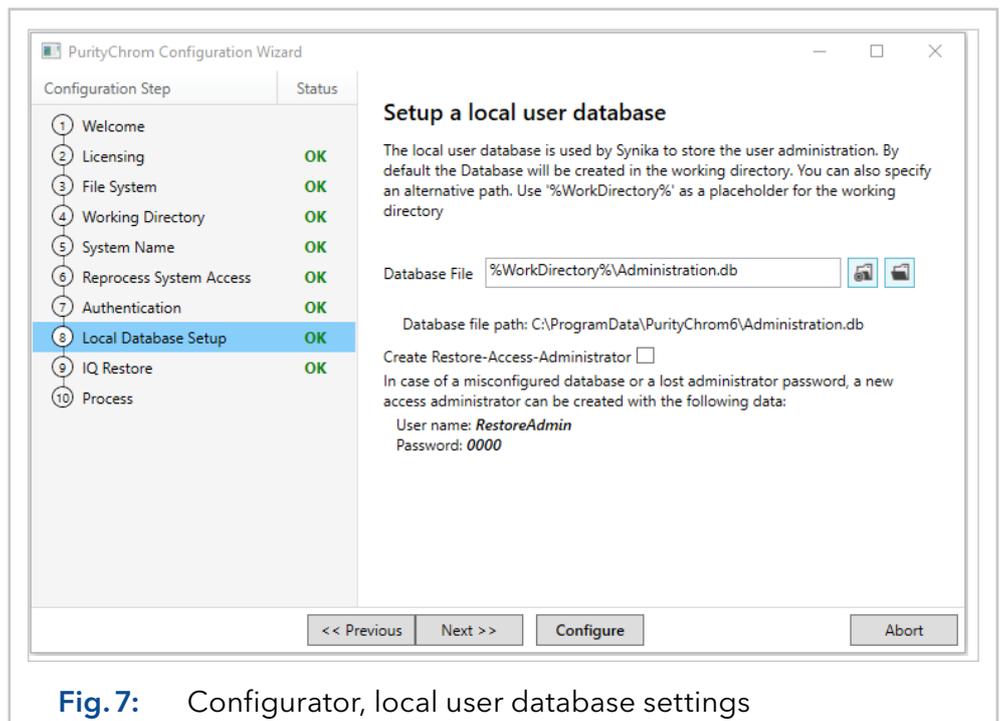
In step 7, the login mode will be chosen. If a database is used, you can further specify whether a MySQL server, a local SQLite database or a Mongo Database is to be used.



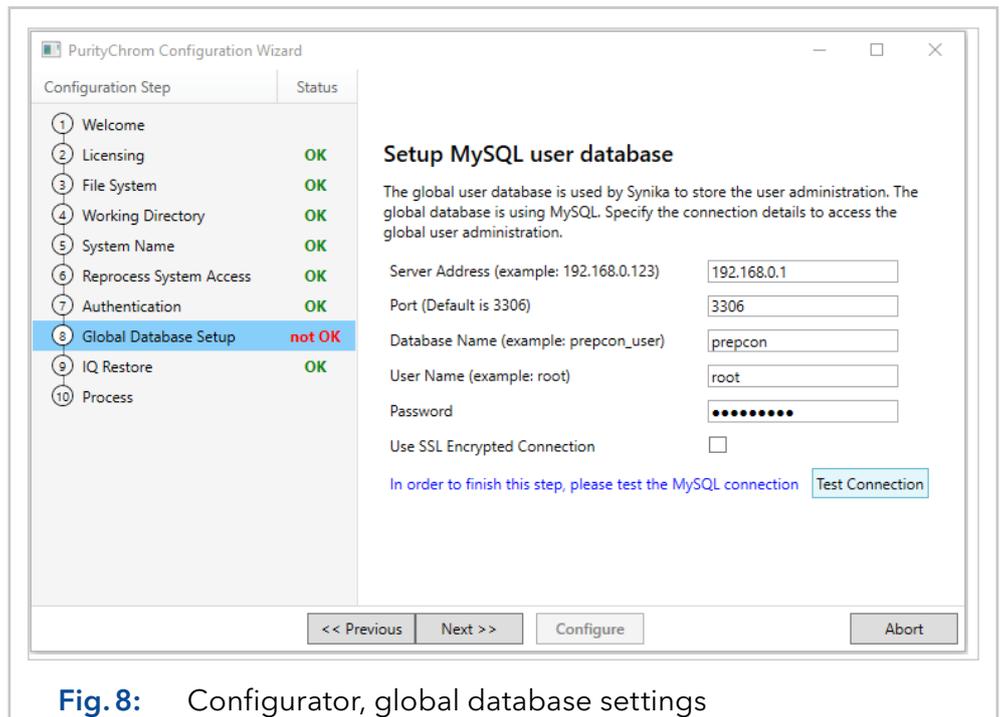
**Fig. 6:** Configurator, authentication

### Step 8: Database setup

If a local SQLite database is used, the database path has to be specified in step 8. The placeholder "WorkDirectory" is replaced by the real working directory path from step 4 during program use. The option "Create Restore-Access-Administrator" should be not activated during an installation, because it is intended to be a safety feature. If the administrator account of PurityChrom is locked because of too many wrong login attempts, the system administrator can run the PrepConConfiguration.exe (saved under C:\ProgramFiles\PurityChrom6) again and activate this option to create a new administrator account with the user name "RestoreAdmin" and the password "0000".



If a MySQL server is used, the server connection information has to be entered. The given data can be checked with the "Test Connection" button.



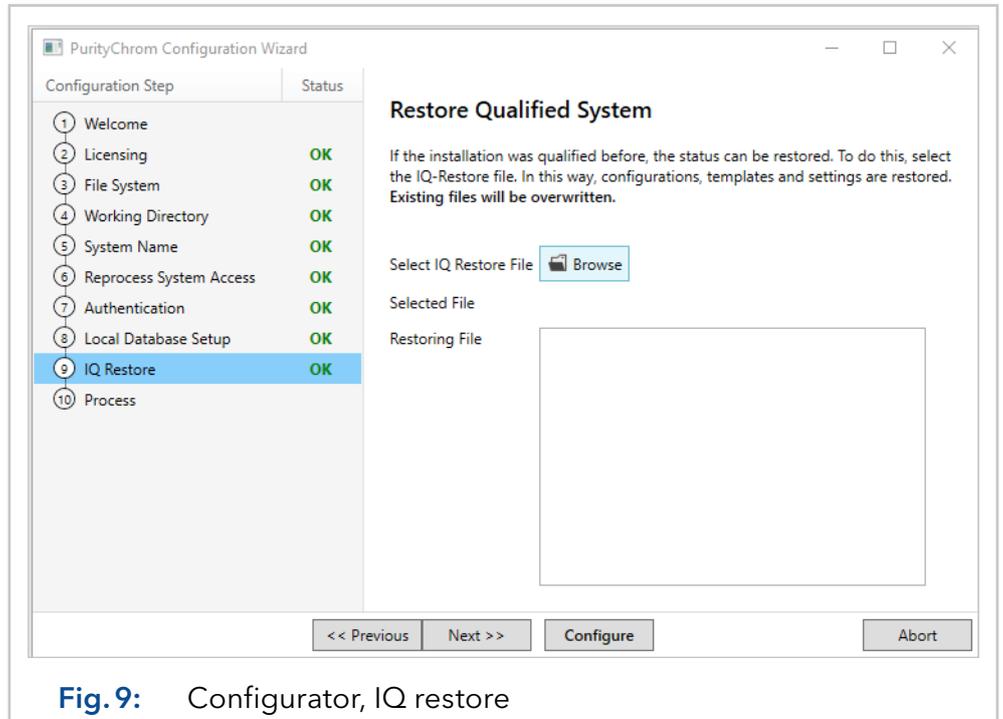
Database setup is not displayed in the PurityChrom Configuration Wizard if Mongo DB was chosen as connection-mode in step 7.

### Step 9: IQ Restore

This step can be used to restore configuration, templates and software settings from a previous installation. For that, press the **Browse** button and select the IQ file of the previous installation which you want to restore.



**Note:** Acknowledge, that already existing files will be overwritten by this step. For an initial installation, this step can be skipped.



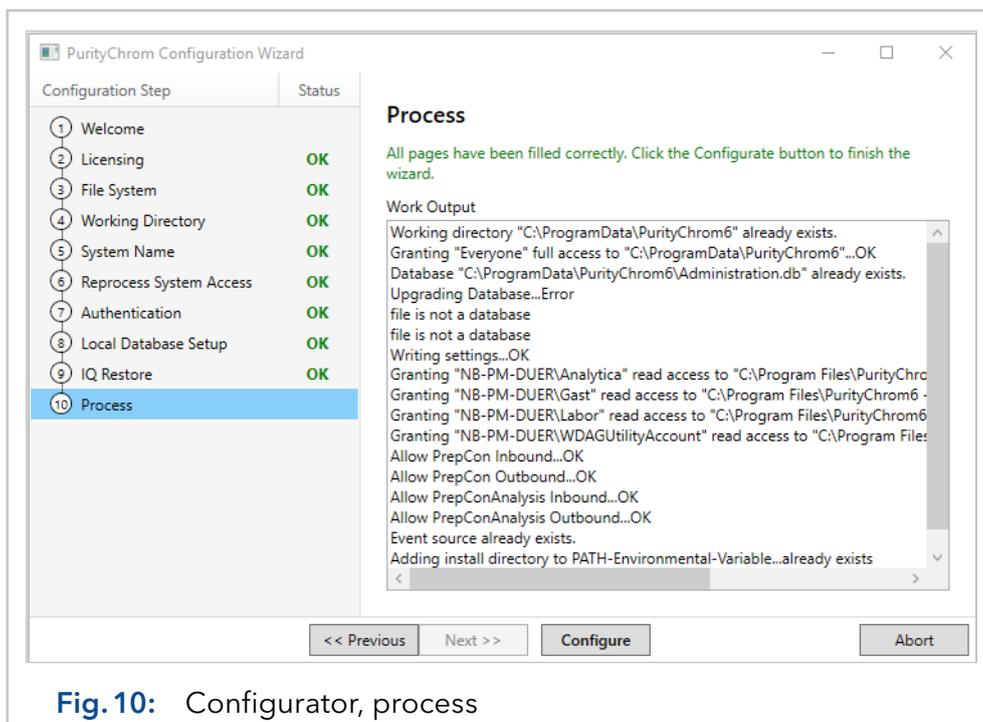
**Fig. 9:** Configurator, IQ restore

**Step 10:** Process

With "Configure" the settings are executed and the configuration is created. In the "Work-Output" text field, the executed actions can be traced. If a local database is created, the initial user and the respective initial password are also displayed in the text field.

The default login is the following:

- Name: Admin
- Password: 0000

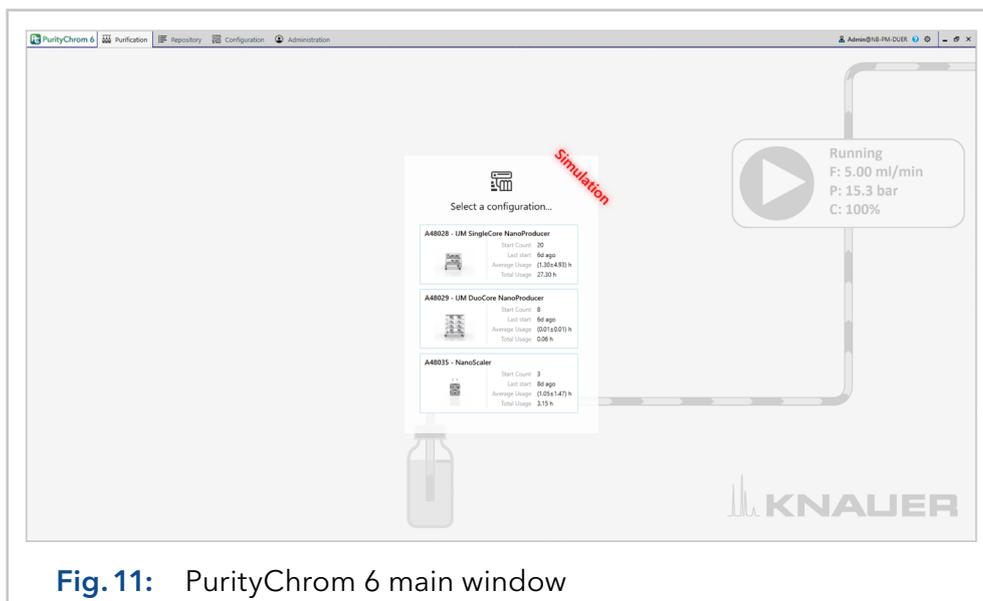


**Fig. 10:** Configurator, process

After installation, first start of the software and entering the login credentials the user is requested to chose a new password. The new password is requested to be eight characters long. To adjust the requirements for the length of the password, see chapter 7 on page 66.

### 3. Main Window

The PurityChrom 6 main window can be divided into two areas: The title bar and the content area. Through the title bar a navigation between the application settings Purification, Repository, Configuration and Administration is possible. The sections content is then displayed in the content area below the title bar. Access to the different parts of the application can be restricted by permissions. If an application section is not accessible to a user due to chosen restrictions in the user administration, it is not visible in the title bar.

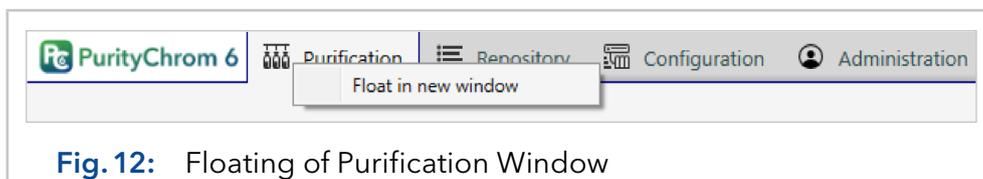


**Fig. 11:** PurityChrom 6 main window

Under  the current user you can log out, change the password, and the current Microsoft Windows user can be linked to the PurityChrom 6 user. Furthermore, the time until the password expires and until the user account expires can be viewed here.

The button  opens the "Info and Settings" window, in which general information about the software as well as general settings can be found. Here the user will find the version ("Build") of PurityChrom 6 as well as the option to switch the system to simulation mode (via the tab "General" and the option "Device Simulation") or to change the color settings of the software see chapter 8 on page 71. The button  opens the PurityChrom 6 Manual.

The Purification application part can be undocked by right-clicking and choosing "float in new window", as shown in the "Fig. 12: Floating of Purification Window" on page 10. The two divided windows can be distributed over two monitors, for example. By closing the undocked window, it is embedded in the main window again.



**Fig. 12:** Floating of Purification Window

## 4. Application part: Purification

The Purification application part is the active runtime part. After a system configuration has been selected, a connection to the devices is established in order to control the devices of the system. Using various control elements, the user can execute device functions directly or load and execute previously created methods and sequences.

### 4.1 Introduction to the user interface

After selecting the configured system to which PurityChrom 6 should connect, the following interface is displayed.

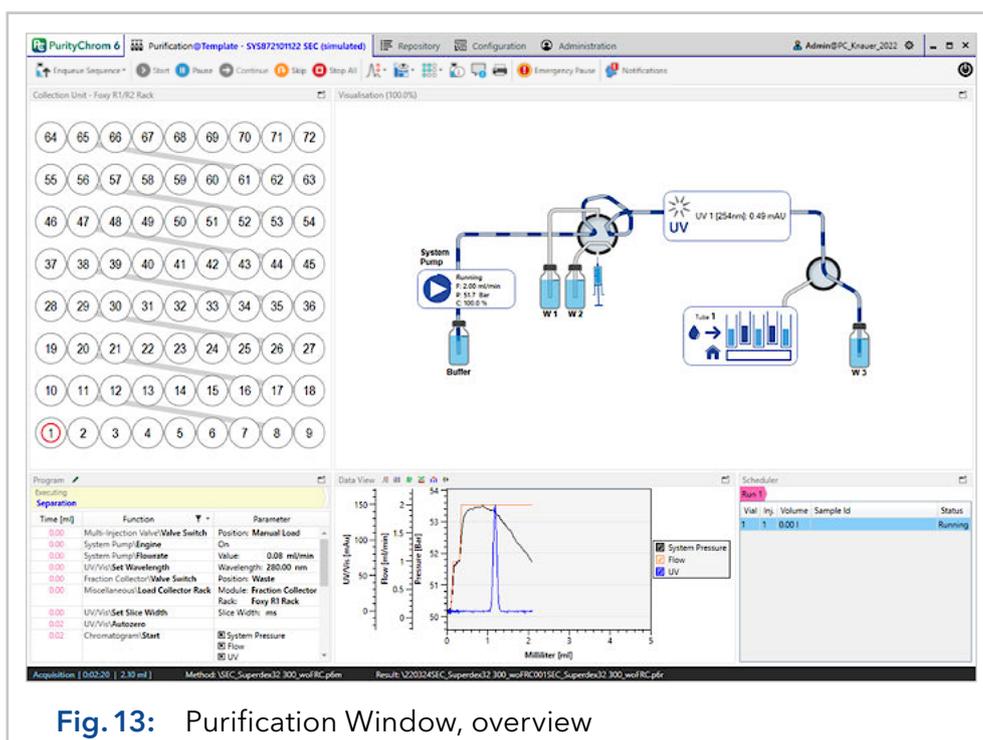
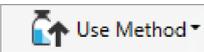
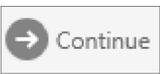
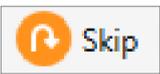
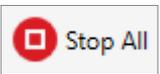
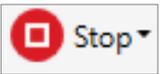


Fig. 13: Purification Window, overview

The user interface can be divided into four areas: The control menu, the classic menu, an area for displaying additional content and the status bar. In the following, the areas of the user interface are explained in detail.

## 4.2 Control-bar

The following functions are found behind the menu icons of the control bar.

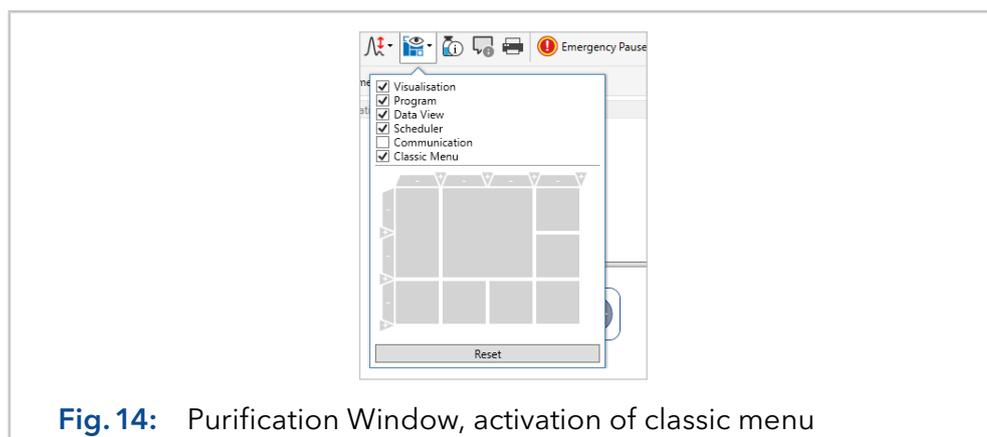
Icon	Description
	Opens a file selection via which methods and sequences associated with the system can be loaded.
	During the execution of a method or sequence additional methods or sequences can be enqueued to be executed after completion of the current run.
	Starts the currently loaded process.
	Pauses the current process.
	Continues a paused process.
	Skips the currently executed function. The exact behavior is depended on the executed method step type: <ul style="list-style-type: none"> <li>■ Command step: Cancel waiting condition and continue with next step.</li> <li>■ Time table step: Jump to the next line of the time table.</li> </ul>
	During process execution, this button cancels the current process after a confirmation prompt. Outside of a process, it stops all devices and data recording.
	During a running sequence the stop button proposes two options. Either the complete purification can be stopped or the current purification step is completed and the process is stopped afterwards.
	The Scaling Menu opens a submenu for adjusting the display of the data channels.
	The View Menu opens a submenu for customizing the visible content. Each content is a separate window that can be docked in the content part below the control bar.
	The Collector Rack Menu selects the rack to be displayed per fraction collectors.
	The Solvent State Menu shows an overview of the fill levels of all solvents and waste containers, a forecast of the expected consumption during the next run and an option to mark all containers as filled or emptied.
	An Online Custom Annotation is only available during a running method and offers the user the option to add a note to the current chromatogram at the specific time point.

Icon	Description
	The Automatic Print displays the status of the automatic print function and enables access to the printer settings.
	The Emergency Pause executes all functions set in the configuration for an emergency pause and thus, gives an option to quickly stop the process in case of an emergency.
	Via Notifications, information, warnings and device errors are displayed.

### 4.3 Classic menu for module control

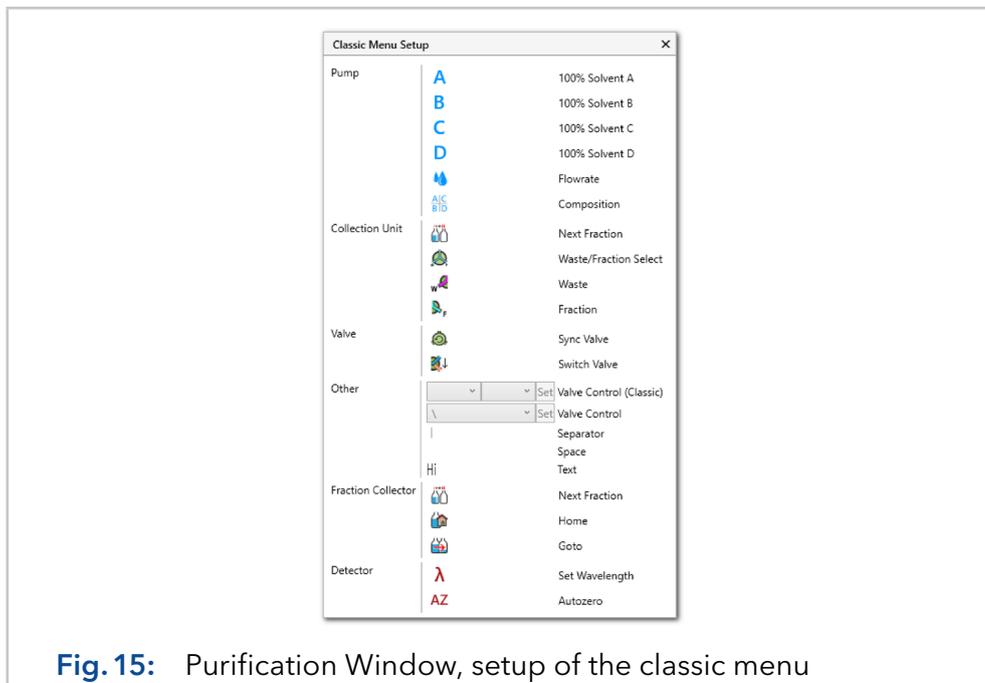
With a well-designed visualisation, it should be possible to operate all system-specific device functions. However, users of PurityChrom 5 may miss the previous menu, in which there was a button for each specific function (e.g. 100% solvent A conveying). To provide buttons for specific functions, the PurityChrom 6 Classic Menu has been introduced, which is displayed under the regular control bar.

The Classic menu can be shown or hidden via the "View Menu".



**Fig. 14:** Purification Window, activation of classic menu

The Classic menu is initially empty and has to be configured. By right-clicking on any point in the menu, the setup window can be opened via the context menu that appears. The setup window shows available buttons and elements that can be dragged and dropped into the menu.

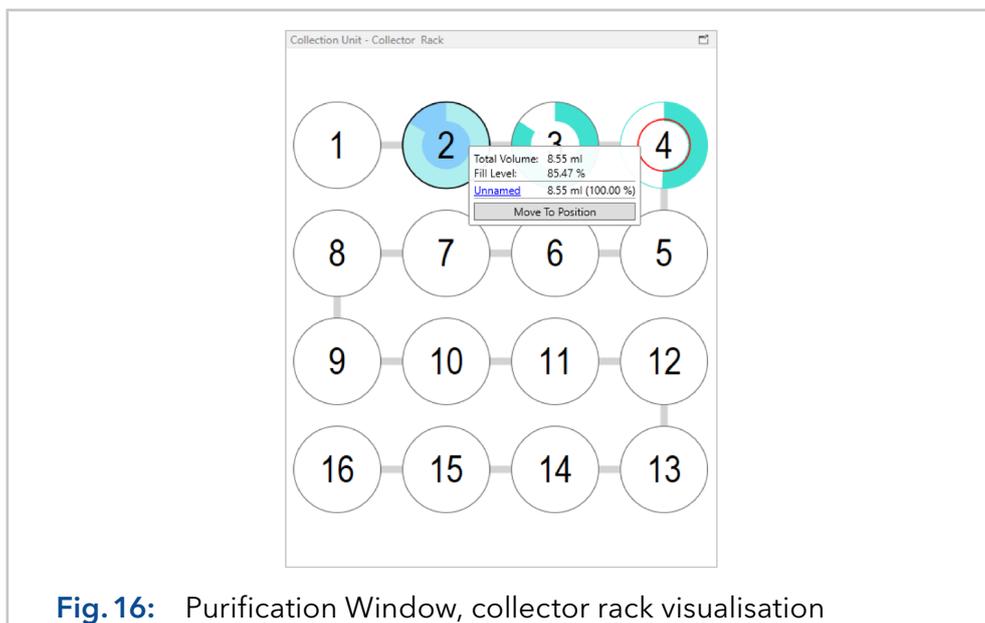


**Fig. 15:** Purification Window, setup of the classic menu

While the setup window is displayed, the buttons cannot be clicked. Clicking on a button while the setup is active, opens a configuration pop-up instead, which can be used to configure the button, such as which module the button refers to. Furthermore, the arrangement of the elements in the menu itself can be changed while the setup is opened by drag & drop.

## 4.4 Collection unit

The fraction collector rack is a content element of the purification application part and can be shown and hidden through the “View Menu”. Per collection unit, one rack display can be shown at a time. The rack display itself shows the geometric arrangement of the positions of the fraction collector. The current position is highlighted by a red circle. If the fractionation is active, the position is outlined by turquoise. The filling level of each tube is visualised by a proportional drawing of turquoise borders.

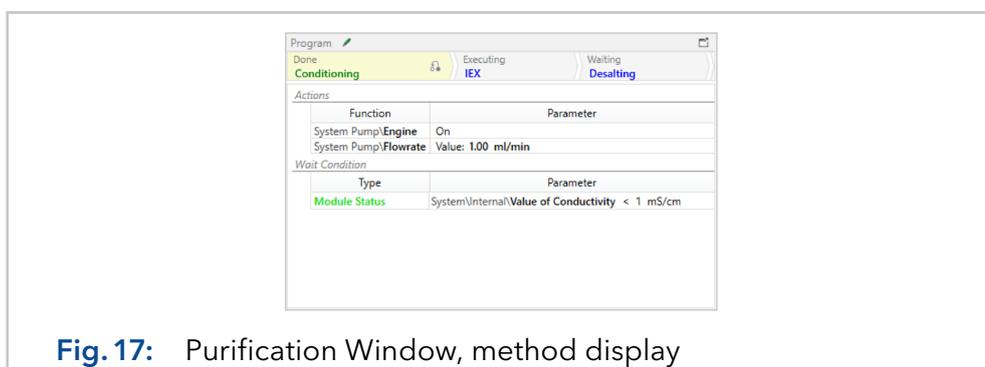


**Fig. 16:** Purification Window, collector rack visualisation

A mouse click on a position displays a pop-up with details of the marked position. The selected fraction is highlighted in light blue color. These details show the total volume delivered into the position, as well as the resulting percentage fill level. The volume filled into the position is further broken down by substance, whereby the substance name is taken from an active threshold or a peak window. For each substance, the origin of the fractionation is further broken down per method execution. The "Move To Position" button can be used to manually move the collection unit (a fraction collector or a fraction valve) to the selected position.

## 4.5 Program display and online-editing

The program display shows the currently loaded program and is a content element of the purification application part, which is shown and hidden by the View Menu. The program is displayed analogously to the method editor display, with some process displays added.

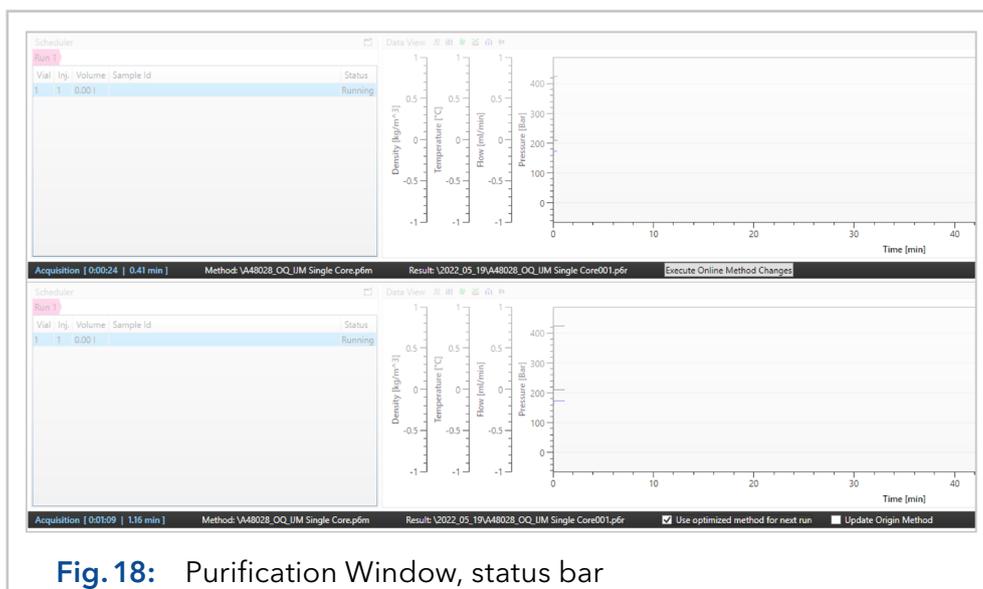


**Fig. 17:** Purification Window, method display

For each step, the status of the execution is displayed. Fig. 17 shows that the "Conditioning" step is currently being processed and the Ion Exchange "IEX" step will follow. For command steps, the status of the waiting condition is also highlighted in color (red = condition not fulfilled; green = fulfilled).

The editing mode can be activated via the pencil icon  in the title bar if the method has already been started. In this mode, the individual steps and containing functions can be edited at runtime. Of course, only steps or time table rows that have not yet been executed should be edited. After the desired changes have been made, the editing mode has to be exited again with the pencil symbol. In the black bar at the bottom of the window, the status bar, there is now a button "Execute Online Method Changes". Only when this button is pressed, the changes will be accepted and executed accordingly.

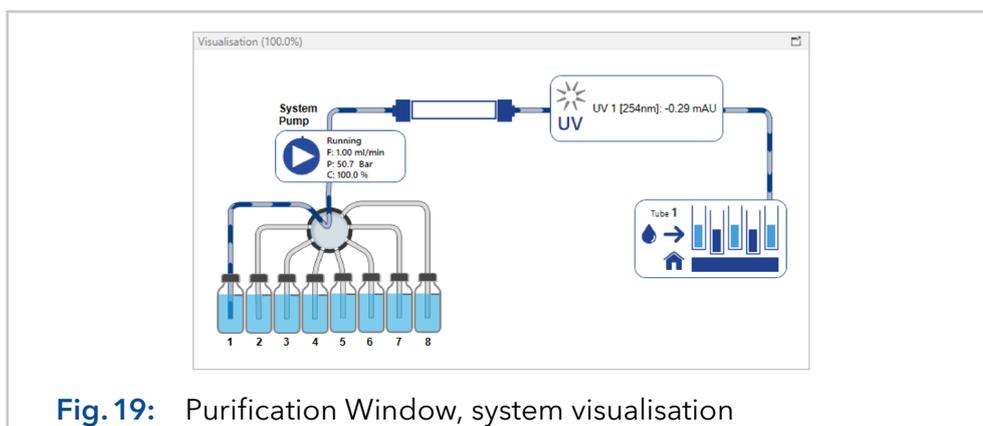
If the changes are also to be used in the next run, a checkmark has to be placed in the "Use optimized method for next run" checkbox in the status bar.



The changes can then also be permanently applied to the original method by also selecting “Update Origin Method” in the status bar additionally. However, the method will only be updated after the end of the current run.

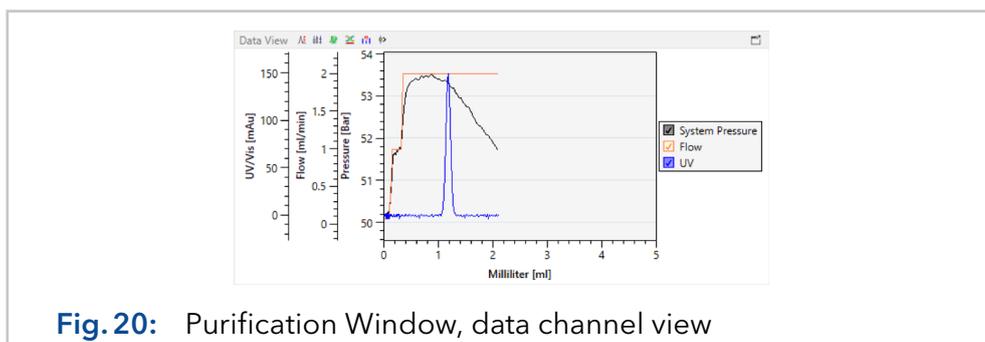
## 4.6 Visualisation

The Visualisation shows the configured visualisation and is a content element of the purification application part, which can be shown and hidden using the “View Menu”. If there are several different visualisation in the configuration, it is possible to switch between them via the title bar showing the name of the currently displayed visualisation.



## 4.7 Data view

The channels created in the configuration are displayed in the content element data view. The defined channel display is also a content element of the purification application part, which can be shown and hidden through the “View Menu”.



**Fig. 20:** Purification Window, data channel view

The axis title and unit can be modified in the configuration. Individual data channels are shown and hidden via the associated legend check box. By clicking on the data channel name, the respective channel is shown with a thicker line than the other ones and the corresponding y-axis will be shown singularly. Clicking again shows the axes of all data channels and all lines in a similar way.

In baseline mode, when no method is executed, a data channel is displayed in the baseline colors. If a data channel is recorded by a method, the recorded data channel is overdrawn in the "recording" chromatogram color chosen in the scaling menu .

Comfortable zooming can be accessed by right clicking and activating "Surpress Auto-Scroll". In the following the user can interact with the graph as follows:

Action	Result
Drag & Drop on the axis	Moving the scale
Mouse wheel + cursor on the axis	Zooming the scale
Ctrl + Drag & Drop on the chart area	Shifting all axis
Shift + Drag & Drop on the chart area	Zoom all axis
Mouse wheel + cursor in the chart area	Shift of all X-axis
Shift + mouse wheel + cursor in the chart area	Zoom all axes in cursor direction
Right mouse button	Context menu
Space bar	Show point values

The automatically scaled graphic display can be regained by pressing the keys Alt+S simultaneously or by deactivating "Surpress Auto-Scroll".

The data channels are displayed in the Data View. The title bar contains the following buttons:

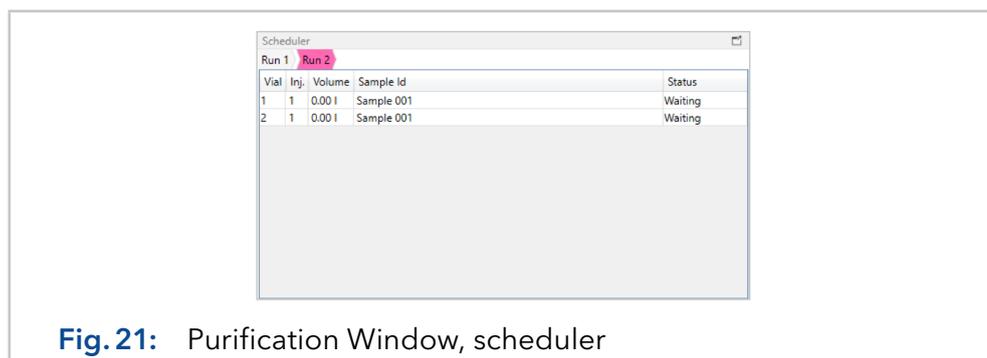
	As in the main menu, the <b>Scaling</b> is used to configure the color and arrangement of the data channels.
	<b>View/Hide functions:</b> Shows and hides the programmed method functions.

	<b>View/Hide sampling Window:</b> Shows and hides the peak windows.
	<b>View/Hide gradient:</b> Shows and hides gradient progression.
	<b>View/Hide annotations:</b> Shows and hides annotations.
	<b>Enable function editing:</b> In online editing, functions can be moved during a method run. To do this, enable online editing via the button, move the mouse pointer over the desired function and move it by drag and drop.
	<p><b>Do multi-function drag:</b> If online editing is enabled, the multi-drag can also be switched on. In this mode, an area in the data channel window can be selected with the left mouse button pressed and then all functions in this area can be moved together with the mouse.</p> <p>Changes made by online editing via the channel view are applied immediately. Only functions that have not yet been executed should be edited. Changes to functions that have already been executed can lead to undesired results.</p>

To use the changes in the next run, check the “Use optimized method per next run” box in the black bar at the bottom of the window, the status bar. To apply the changes permanently to the original method, a checkmark can be placed next to “Update Origin Method” in the status bar.

## 4.8 Scheduler

The scheduler provides an overview of already executed and scheduled methods. When loading a method or a sequence, a new “Run” is created in the scheduler, under which the planned executions are listed.



**Fig. 21:** Purification Window, scheduler

If a method is loaded, the number of executions is not fixed, as it can be started by the user as often as desired. The scheduler will always display the next execution as well as the previous ones. When loading a sequence, the number of executions has already been defined in it, so the scheduler lists all planned executions.

The injection volume and the sample ID can be entered for scheduled executions at runtime. By clicking on a line, details of the execution can

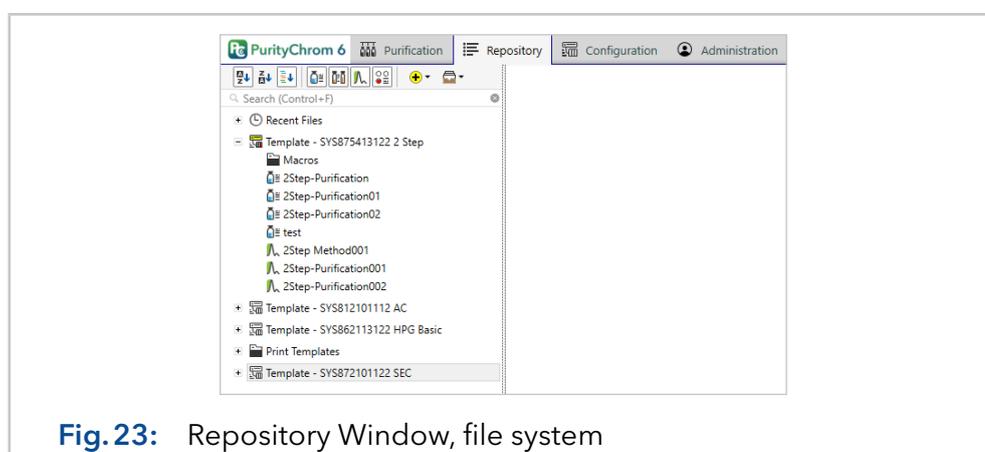


## 5. Application Part: Repository

The repository part of the application provides unified access to system-related files. Files are edited using the file type-specific editor. The basic idea of the repository is to provide an uniform and simple overview of the files, regardless of where the files are stored.

### 5.1 Introduction to the user interface

The repository, which is shown as an example in Fig. 23, contains a tree structure with files on the left-hand side. The files (methods, sequences and result files) are initially grouped by configuration. The top group "Recent Files" shows a list of the most recently opened files. The group "Print Template" shows the existing templates for printing data (see chapter 5.9 on page 48). Double-clicking on a file displays it in the associated editor on the right-hand side of the window.



**Fig. 23:** Repository Window, file system

The search field filters the tree structure according to the entered substring. To facilitate file selection, the command bar provides the following display filters.

	<b>Sort by name ascending:</b> sorts the files by name in ascending order
	<b>Sort by name descending:</b> sorts the files by name in descending order
	<b>Sort by type</b> (methods, sequences, result files etc.): organizes the data files in the tree structure according to their type
	<b>Show methods:</b> shows or hides the method files in the tree structure
	<b>Show sequences:</b> shows or hides the sequence files in the tree structure
	<b>Show reconstructions:</b> shows or hides the result files in the tree structure
	<b>Show macros:</b> shows or hides the macro files in the tree structure

The icon  can be used to add further folders, methods, sequences, compare reconstructions and value-fraction tables, or load archives. Files can be combined into an archive for file exchange or external storage.

Files can be added to archives via drag and drop onto the symbol  or right click on the desired files or folders while choosing the function add to archive. If the archive menu is opened with a mouse click, the elements are listed. With another click on "Save", the chosen data files can be stored at a chosen location as an archive. This archive can be later imported on another computer or be sent to the [KNAUER customer support](#) for advice.

For the import of an archive, choose the option "Open Archive" via the button . The included chromatograms, methods and sequences are displayed in the tree structure. To also import the system configuration, use a right-mouse-click on the opened archive and choose the option "import archive data" and define the path for data storage (C:\ProgramData\PurityChrom6 is recommended). The system configuration is displayed in the list of configured systems in the Purification Window and the Configuration Window.

## 5.2 Editor: Method

The method editor is opened in the repository by double-clicking on a method from the file tree. A new method can be added via the button . Essentially, the method editor consists of two "pages": The General page and the Procedure page.

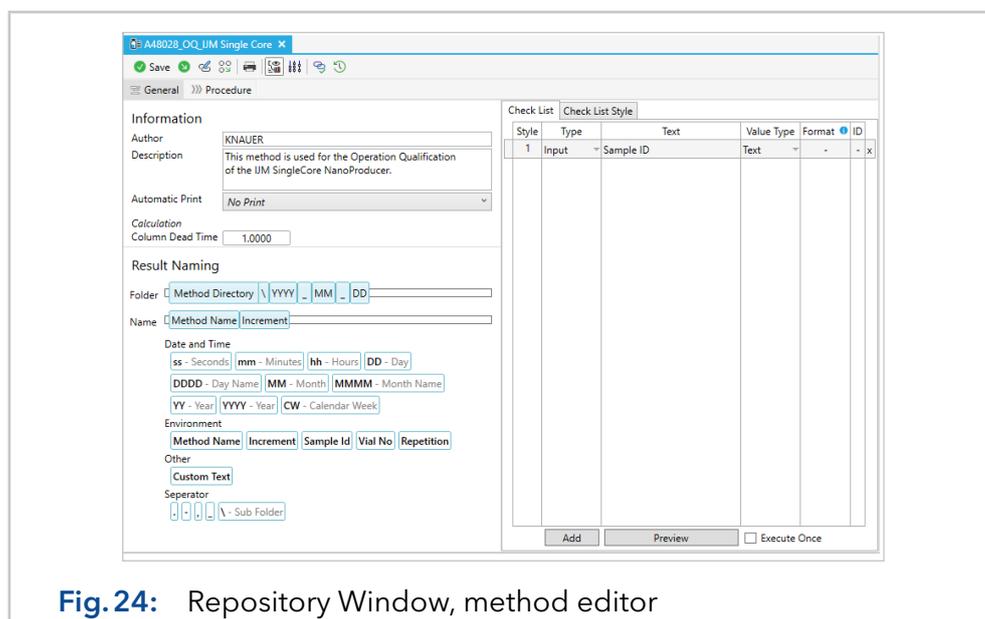
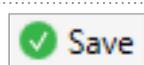


Fig. 24: Repository Window, method editor

The following buttons are located in the toolbar of the method editor:



**Save:** Saves the method



**Saves as:** Save the method under a different name

	<b>Signature:</b> add an electronic signature to the method. If the option "validated" was activated, no changes will be possible afterwards. Electronic signatures are only available in case of an activated user administration.
	<b>Save as macro:</b> Save the method as a macro.
	<b>Print:</b> Prints the method
	<b>Show visualisation:</b> Shows or hides the system visualisation in the method editor
	<b>Show graphical function editor:</b> Shows or hides the graphical function editor
	<b>Show revisions:</b> Open a selection window to view or save older versions of the method
	<b>Signature History:</b> shows the given electronic signatures for the method

## 5.2.1 Method part: General

The **General** page is divided into three sections: **Information**, **Result Naming**, and **Check list**. Under **Information**, the author, a general description of the method, and the naming of the solvents at the individual pumps can be documented. In addition, a template for an automatic printout after the execution of the method can also be selected here. This template is loaded automatically after loading the method in the purification for the automatic printout function. Further, a column dead time can be entered, which is used for the calculation of the column indices in the result files.

Under **Result Naming**, the storage path and the structure of the file name of the result files can be configured. To do this, simply drag the desired components into the **Folder** or **Name** field. Deleting a selected element is done with the delete button of your keyboard. The selection is made by left-clicking on an element with the mouse.

The definition of the checklist, which has to be filled by the operating user before the method is executed, is done under **Check List** on the right side of the window. Lines can be added to the checklist via the **Add** button. Each line has the following columns:

Column	Values and meaning	
<b>Style</b>	The assignment to a style defined on the Check List Style tab	
<b>Type</b>	Input	The line expects an user input.
	Label	The line is displayed as text in the checklist.
	Separator	The line contains a horizontal line as a separator to the following lines.
	Space	The line will be empty.

Column	Values and meaning	
<b>Text</b>	This field should contain the displayed text. It is only available for the types Label and Input.	
<b>Value Type</b>	Text	Input of a text
	Check	Input of a checkbox
	Value	Input of a value with a defined format
<b>Format</b>	For the value type Value, an input format needs to be defined here. For more explanations, see the table below or the tooltip in the column header.	
<b>ID</b>	Specification of an ID, which can then be used as an element of the naming convention for file path or filename.	

Possible placeholders (Format-Column)

<b>0</b>	Any digit from 0 to 9. Input required
<b>9</b>	Any digit from 0 to 9. Input optional
<b>#</b>	Any digit from 0 to 9, a space, + or -. If not entered a space is inserted automatically
<b>L</b>	Letter (a-z or A-Z). Input required
<b>?</b>	Letter (a-z or A-Z). Input optional
<b>A</b>	Alphanumeric character (a-z, A-Z, 0-9). Input required
<b>a</b>	Alphanumeric character (a-z, A-Z, 0-9). Input optional
<b>.</b>	Placeholder for decimal separator.
<b>,</b>	Placeholder for thousands separator
<b>:</b>	Hours: minutes separator for times
<b>/</b>	Separator for dates
<b>\$</b>	Currency symbol
<b>&lt;</b>	Converts all the following input characters to lowercase.
<b>&gt;</b>	Converts all following input characters to lowercase.

The configured checklist can be displayed via **Preview**. The checkbox **Execute Once** defines that the checklist is only executed once after loading the method in the purification, but not in the following runs. Please mention, that if a method containing a checklist is used in a sequence, the checklist is executed only once at the beginning of the sequence.

## 5.2.2 Method part: Procedure-editor

By selecting the "Procedure" tab, the procedure page appears, as shown in the following screenshot. On the left side of the screen is the **step editor**, on the right, the **system visualisation** and a **graphical function editor** can be enabled .

The step editor consists of a **step list** in the area of the title bar and an editor field below. New steps can be added via the plus button, additionally the step type can be chosen. The sequence order can be changed through drag and drop of the single steps and naming of the steps can be changed through left-click on the text.

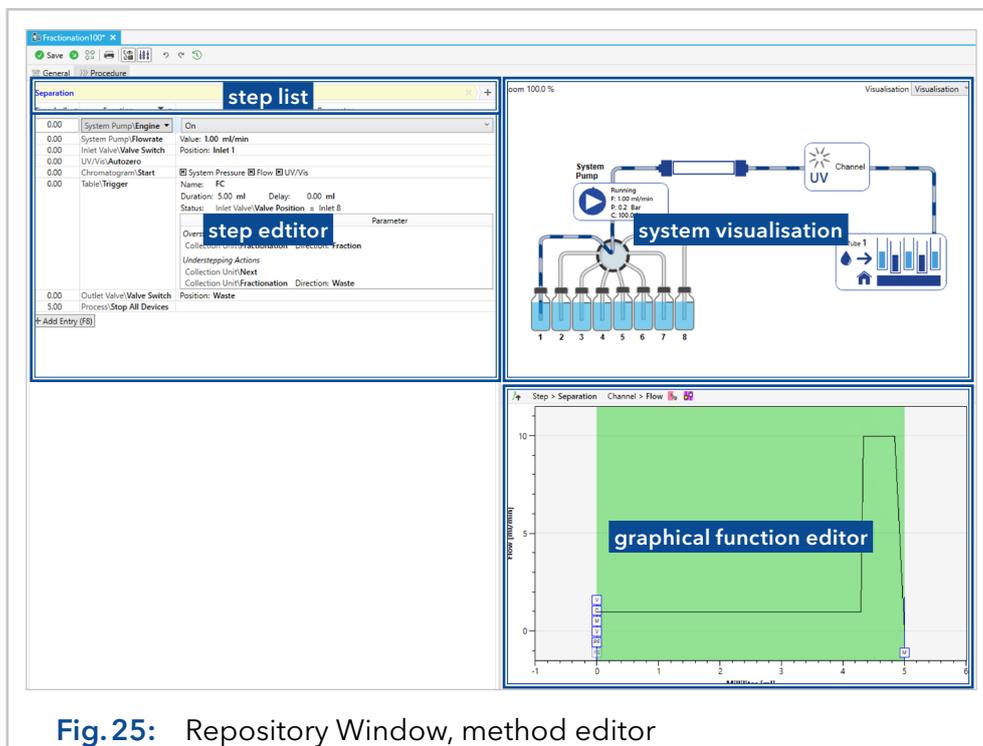
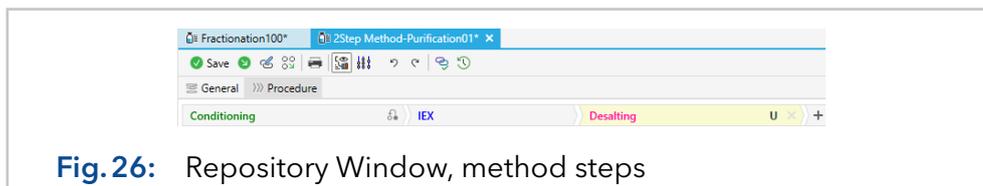


Fig. 25: Repository Window, method editor

The visibility of the system visualisation and the graphical function editor can be controlled via the buttons **Show visualisation**  and **Show graphical function editor** . The visualisation is used to preview the system status up to the selected point in time (e.g. the selected time table line). If a device function with parameters is to be entered in the step editor, this entry can optionally also be made via the visualisation. Just insert a new line into the step editor via F8 of your keyboard or the **Add Entry** button and then, click into system visualisation to adjust the method line.

The graphic function editor shows the programmed functions of a step on a time axis. The functions displayed in this way can be shifted in time with the mouse. Furthermore, threshold values and time window functions can be created and adjusted here. To see what the programmed functions look like in an already recorded run, a previous result file can be displayed as a reference via the button **Load Chromatogram**  located in the title bar of the graphical function window.

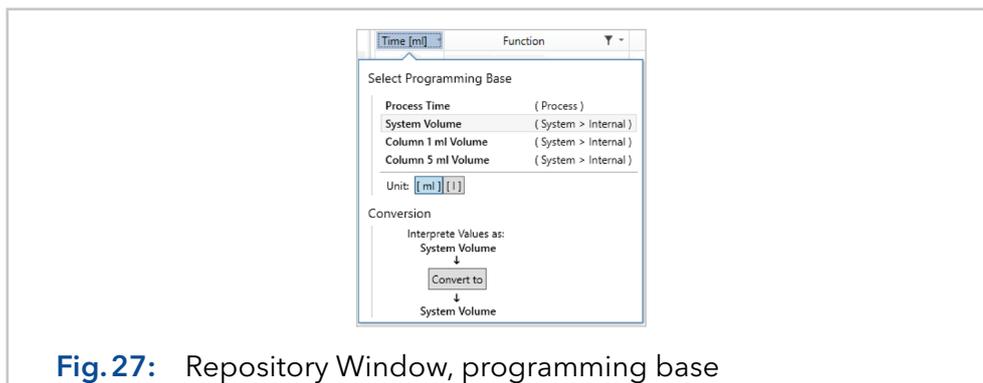
When creating a method, the steps **Conditioning** and **Separation** are inserted by default. The idea behind these standard steps is that in a common preparative application, the system has to be prepared first. A good place for the functions and conditions programmed for this is the conditioning step. Once the system is appropriately conditioned, the time-dependent system control takes place in the purification step.



**Fig. 26:** Repository Window, method steps

PurityChrom 6 defines three different step types:

- Command step:** A command step contains a list of functions that are all executed at once. It is possible to define one or more waiting conditions. The fulfillment of those leads to the completion of the step. Furthermore, a timeout can be defined, which, if exceeded, causes an action to be executed. The flag "Execute on Initialize"  can be used to define whether the step should be executed immediately after loading the method or just after pressing the start button.
- Time-table step:** A time table contains a sequence of programmed functions at specific times. The programming base of the time table can be selected by clicking on the time column header. In addition to the normal process time, the system volume or the column volume can also be used as programming base.



**Fig. 27:** Repository Window, programming base

The selection of the programming base only results in the interpretation of the entered times under a different unit. For the conversion of the values into another programming base, the "Convert to" button can be used.

Macro-Step:	<p>Frequently recurring procedure parts can be outsourced as a macro step (e.g. valve positions of several valves, rinsing steps etc.). The macro is integrated via a macro step into the method. During normal use of the macro step, it is also displayed as a macro step in the procedure during method execution. With the "U" flag ("U" for "Unwrap") in the step title ( <code>Desalting U</code> ), it can be specified that the macro step is replaced by the macro procedure at runtime. In the case of an inactive flag, the macro step is displayed only with its name instead of its time table during an active method.</p> <p><b>Note:</b> If a macro is used in its wrapped format, no data is recorded during this step for later revision.</p>
-------------	---

An overview of the programmable functions is given below. It should be noted that some device functions are only available if the corresponding module driver has been integrated via the configuration part of the software. Also, not all functions are supported by all devices/drivers and may therefore not be available. Parameters in "{}" are considered optional.

Function	Short description / parameter
AD Converter\ <b>Slice Width</b>	Sets the data point spacing per channel.
AS61L Sampling Module\ <b>HeadSpacePressure</b>	De-/activates the head space pressure option of the autosampler during injection. This option will ensure an exact injection volume and should be only used with closed vials.
AS61L Sampling Module\ <b>Inject</b>	Sends the start signal for an injection to the autosampler.
AS61L Sampling Module\ <b>InjectionMode</b>	Defines the type of injection for the autosampler. This command will overwrite the setting used in the system configuration.
AS61L Sampling Module\ <b>InjectionValveSwitch</b>	Switches the injection valve of the autosampler in inject or load position. This function is part of the normal injection routine and thus, should be only programmed when needed for a special injection routine.
AS61L Sampling Module\ <b>Prepare Injection</b>	Initializes the preparation for the next injection of the autosampler. This command should be used in combination with a waiting condition in a command step. When using stacked injections, this command is automatically added to the method by the software.
AS61L Sampling Module\ <b>TrayTemp</b>	Defines the temperature used for cooling the autosampler tray. This command will overwrite the setting used in the system configuration.

Function	Short description / parameter
AS61L Sampling Module\ <b>Wash</b>	Starts an initial wash procedure using wash or transport solution to flush all lines thoroughly. This command is not needed to start a wash procedure after injection because the wash program that was defined in the system configuration will always be automatically performed after injection.
Balance\ <b>Calibrate</b>	Starts the calibration routine of the scale. Ensure that the scale was emptied before.
Balance\ <b>Custom Density</b>	Sets a density of the weighted liquid.
Balance\ <b>Select Density</b>	Selects one of the pre-defined densities for the weighted liquid.
Balance\ <b>SetTare</b>	Defines the weight that should be used as zero point.
Balance\ <b>Tare</b>	Sets the weight to zero while the weighing pan is loaded i. e. with an empty container.
Balance\ <b>Text</b>	Displays the defined text in the display of the scale.
Balance\ <b>Zeroing</b>	Sets the weight to zero. Only possible while the weighing pan is empty.
Chromatogram\ <b>Start</b>	Starts data recording for one or more channels.  Parameter: Channels to start; {length of data track}.  Note: If the length of the data track, the "fixed duration", is specified, the data track does not have to be stopped manually. If the chromatogram end is longer than the time table, the acquisition will continue even after the procedure is finished.
Chromatogram\ <b>Stop</b>	Stops data recording for selected channels.  Parameter: Channels to stop.
Collection Unit\ <b>Fractionation</b>	Positional choice: Waste or Fraction.
Collection Unit\ <b>Next</b>	Moving to the next fraction.
Collection Unit\ <b>Set Fraction Limit</b>	Uses the tube size from the rack as the limit after which the next fraction is switched to.
Collection Unit\ <b>Set Time Limit</b>	Sets the time span after which the next fraction is switched to.

<b>Function</b>	<b>Short description / parameter</b>
Collection Unit\ <b>Set Volume Limit</b>	Sets the fraction volume after which the next fraction is switched to.
Collector\ <b>Home</b>	Moves the collector arm to home position.
Collector\ <b>Next</b>	Moves the collector arm into the next position.
Collector\ <b>Position</b>	Moves the collector arm to the specified position.
Collector\ <b>Valve Switch</b>	Switches the collector valve.
Collector Valve Proxy\ <b>Fraction</b>	Starts fractionation via fraction valve.
Collector Valve Proxy\ <b>Position</b>	Switches the fractionation valve into the position
Collector Valve Proxy\ <b>Next</b>	Switches the fractionation valve into the next position.
Detector\ <b>Autozero</b>	Zero point adjustment
Detector\ <b>Set Slice Width</b>	Sets the data point spacing of a channel of a detector.
Detector\ <b>Set Wavelength</b>	Sets the wavelength.
Detector\ <b>Switch Lamp</b>	Switches the lamp on or off.
Documentation\ <b>Page Snapshot</b>	Creates an overview of device stati as defined in the system configuration in the tab „System Documentation“. Here also the page is defined which can be set as parameter. All documentation snapshots will be shown in a separate chapter in the result file.
Documentation\ <b>Start Status Recording</b>	Starts the continuous documentation of all device stati. A separate chapter will be added to the result file including the status recording.
Documentation\ <b>Stop Status Recording</b>	Stops the documentation of all device stati.
KNAUERDAD\ <b>Documentate Spectrum</b>	Saves the Spectrum under a chosen name at this time point during the purification.
KNAUERDAD\ <b>Reference Spectrum</b>	Takes a reference Spectrum and zeros the signal.

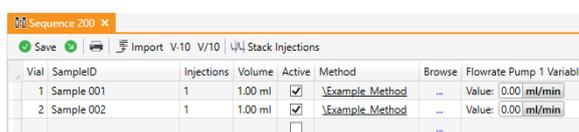
Function	Short description / parameter
KNAUERDAD\ <b>Set Channel Wavelength</b>	Sets the Wavelength of the chosen Channel.
KNAUERDAD\ <b>Switch lamp</b>	Switches the detector lamp on or off.
Miscellaneous\ <b>Clear Collector Rack</b>	Resets fill levels for a collection unit.
Miscellaneous\ <b>Load Collector Rack</b>	Loads a rack definition for a collection unit.
PID Controller\ <b>Control</b>	Sets the parameter which should be controlled.
PID Controller\ <b>Enable</b>	Activates or deactivates the Controller.
Process\ <b>Annotation</b>	Sets annotations in the chromatogram.
Process\ <b>Break Step</b>	This function is only available inside a threshold or trigger. It can be used to skip the next lines of a method and continue with the next method step depending on the trigger or threshold signal.
Process\ <b>Custom Injection Guide</b>	Defines injection parameters when method is used outside of a sequence.
Process\ <b>Instant Method Execution</b>	Immediately executes the specified method.
Process\ <b>Notification</b>	Creates a customizable message that will be shown in the notification window. This notification will be also documented in the result file.
Process\ <b>Script Execution</b>	Enables the execution of a script file. For this function, the user needs special knowledge about usage of automatic scripts.
Process\ <b>Stop All Devices</b>	Stop all devices.
Process\ <b>Wait</b>	Holds the process until a condition is fulfilled. Parameter: Condition; Pause or Hold. Pause: Pauses the current method. Hold: Pauses the method and continuously shifts the steps still to be processed backward until the method is continued while the pumps keep running.
Pump\ <b>Engine</b>	Switches the pump on or off.
Pump\ <b>Flowrate</b>	Sets flow rate.
Pump\ <b>Max. Pressure</b>	Sets maximal pressure limit.

<b>Function</b>	<b>Short description / parameter</b>
Pump\ <b>Min. Pressure</b>	Sets minimal pressure limit.
Pump\ <b>SSV A</b>	Defines the position of the solvent selection valve A.
Pump\ <b>SSV B</b>	Defines the position of the solvent selection valve B.
Pump\ <b>Stop Pump on Pause</b>	Defines whether the pump is stopped when the time control file is paused or not.
Pump\ <b>Solvents</b>	Sets solvent composition.
Pump Union\ <b>Sub-Gradient</b>	Sets the solvent composition of a pump union consisting of low pressure gradient pumps.
RI Detector\ <b>Open Flush Valve</b>	Opens the flush valve for a to be defined time span in order to flush the reference cell of the detector.
RI Detector\ <b>Set Ext. Range</b>	Enables the extension of the dynamic measuring range to +100 % or -100 %.
RI Detector\ <b>Set Temperature</b>	Setting of the device temperature.
RI Detector\ <b>Switch Flush Valve</b>	Switches flush valve in the position of choice. Make sure to switch the valve back to close before starting a run.
RI Detector\ <b>Zero Glass Adjustment</b>	Recalculation of the zero glass position.
Standard Valve\ <b>Next</b>	Switches the valve to the next position.
Standard Valve\ <b>Switch</b>	Switches the valve to a certain position.
Table\ <b>Acoustic Signal</b>	Playing an audio file as signal.
Table\ <b>Conditional Restart</b>	Restarts the method in the chosen step when the programmed conditions are met.
Table\ <b>Continue Marker</b>	If the condition of the conditional marker is fulfilled (which means that the value exceeded and again fell below the threshold value), the process, which might be a sequence or a looped method using the conditional restart function, will be continued. If the condition is not fulfilled, the process will be stopped.

Function	Short description / parameter
Table\ <b>Threshold</b>	Actions can be executed when the threshold value is over- or understepped. The execution can take place immediately or with a delay. A threshold value always monitors only one data channel, but multiple thresholds can be configured in a method.
Table\ <b>Trigger</b>	The trigger monitors the status of a module and can perform actions when the status is changed, e.g. when an air sensor detects air.
Table\ <b>Sampling Window</b>	Definition of a time window during which functions for peak events can be programmed.
Text Provider\ <b>Text</b>	Displays the defined text in the system visualisation. To enable this function, a text field has to be defined in the system configuration in the tab system visualisation, bound to the module "text provider" and "[Text]" has to be given as entry for the text.

### 5.3 Editor: Sequence

Sequences are used to process one or more methods one after the other. A new sequence is created via the button  in the title bar. By double-clicking on a sequence in the file tree, the sequence editor is opened on the right side of the window.



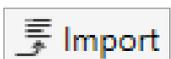
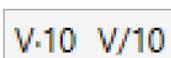
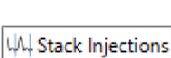
Vial	SampleID	Injections	Volume	Active	Method	Browse	Flowrate Pump 1 Variable
1	Sample 001	1	1.00 ml	<input checked="" type="checkbox"/>	\Example.Method	...	Value: 0.00 ml/min
2	Sample 002	1	1.00 ml	<input checked="" type="checkbox"/>	\Example.Method	...	Value: 0.00 ml/min

**Fig. 28:** Repository Window, sequence table

A sequence is a list of runs containing the following entries:

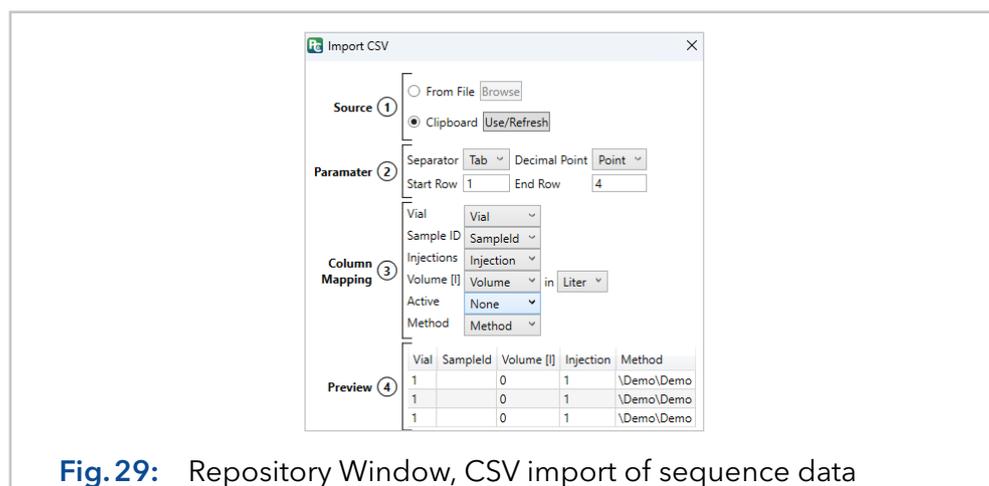
Vial:	Sets the vial number which should be used for injection (only necessary for autosampler use).
Sample ID:	Documents the name of the sample to be recognized later in your chromatogram data.
Injections:	Sets the number of repetitions for the respective row.
Volume:	Sets the injected volume (without autosampler usage, only for documentation).
Active:	If the checkmark is set, the respective row will be performed in the sequence.
Method:	Displays the name of the used method file.
Browse:	Enables searching for methods in the file tree.

Additional columns may appear if the method used includes variables or placeholders. A variable can be defined in the Procedure Editor through a right click on a compatible function (i.e. flow rate) and activation of the checkbox is variable. Here also a name can be defined. Afterwards, the variable appears in the sequence editor to be adjusted for individual runs. Furthermore, the following functions can be found in the title bar of the sequence editor:

	<b>Save:</b> Saves the sequence
	<b>Save as:</b> Saves the sequence under a new name
	<b>Print:</b> Prints the sequence table
	<b>Import CSV Data:</b> Imports CSV data into the sequence table
	<b>Multiply/Divide:</b> Multiplies or divides the injection volume of a selected row by 10
	<b>Stack injections:</b> opens/closes the window for setting up stacked injections.

New rows are automatically created as soon as the last row was filled with information. To add a row above a selected one, use a right-mouse click and choose insert row. To delete a row of a sequence table, simply select the respective row and use the delete key of your keyboard.

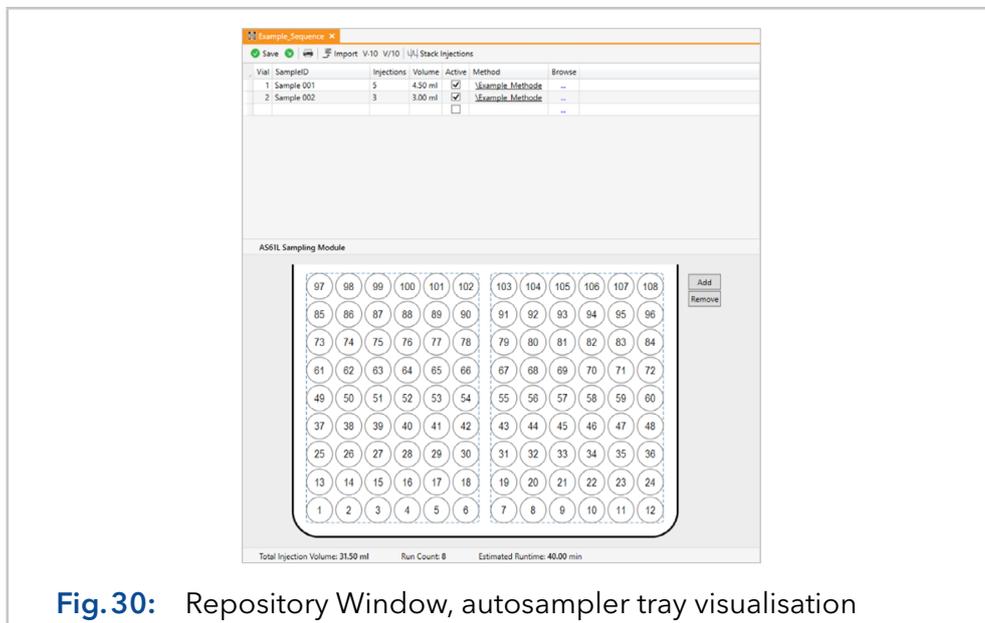
Sequence tables can be exported by copy and paste. In the same way, tables of the same format can be copied from excel and pasted to the sequence editor of PurityChrom 6. For importing a table as CSV file, whose columns are in a different order, the **Import** button opens a dialog window where the user can make the appropriate settings.



**Fig. 29:** Repository Window, CSV import of sequence data

Users who are familiar with PurityChrom 5, might miss the option to write sequences via an autosampler control file. To enable a similar interface appearance in PurityChrom 6, disable the option "Use Sequence Table Editor" in the tab "User Interface" of the software settings under the  button (see chapter 8 on page 71).

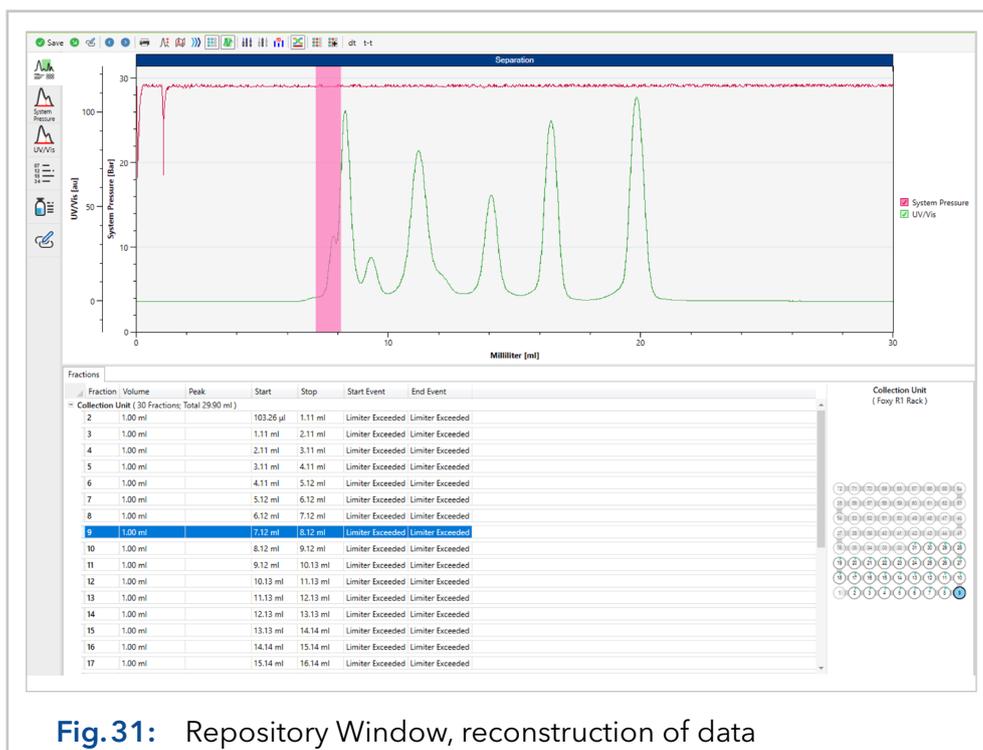
If the autosampler AS 6.1L is used in a system, the sequence editor will also show the configured autosampler tray with the numeration of the individual vials. By selecting one of the vials and pressing the **Add** button the vial number will be inserted in the sequence table. The **Remove** button can be used to delete all lines of the sequence table that include the selected vial number. The status bar under the tray visualisation summarizes the total injection volume, the number of runs (run count), and the estimated runtime of the written sequence.



**Fig. 30:** Repository Window, autosampler tray visualisation

## 5.4 Editor: Reconstruction

The reconstruction window shows a result file that documents a method execution. To open it, use a double-click on a result file in the file tree. On the left side, there are different sections (overview, integration, documentation, documentation snapshots, status recording, method, signature) listed with the content of each displayed on the right side. The control bar is at the top of the window, with each section providing its own control set.



**Fig.31:** Repository Window, reconstruction of data

The following menu items are available in each section.



**Save:** Saves the current reconstruction view.



**Save as:** Saves the reconstruction view under a new name.



**Sign:** After inserting the user's password and confirming the result file as validated, a new section Signature History is added to the result file summarizing the checks performed on the result file.



**Open previous reconstruction:** The previous run saved in the same directory will be opened in the same tab.



**Open next reconstruction:** The next run saved in the same directory will be opened in the same tab.

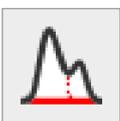


**Print:** Prints a report about the reconstruction.

On the left side of the editor, it is possible to switch between the sections. The symbols are described below.



**General overview:** Shows an overview of the method execution, which includes all data tracks, the executed functions, the peak windows and the collected fractions.



**Integration:** Shows a single data track and the associated results of the integration. The integration parameters can be graphically adjusted here.

	<p><b>Documentation:</b> Shows the filled checklist, the general documentation and the audit trail of the result file.</p>
	<p><b>Method:</b> Shows the documented method in the state that produced the generation of this result file.</p>
	<p><b>Signature History:</b> If the result file has been signed, the signing history with validity information is displayed here.</p>
	<p><b>Documentation Snapshot:</b> This section is only displayed if the command „Page Snapshot“ was used in the method. The section will list all device stati that were chosen in the configuration at the programmed time point in the method.</p>
	<p><b>Status recording:</b> This section is only displayed if the command „Start/Stop Status Recording“ was used in the method. The section will show the system visualisation and its changes during the chosen recording period. Additionally device stati or data traces at specific time points can be displayed here.</p>
	<p><b>DAD Spectrum:</b> This section is only displayed when a DAD is used in the system and displays the DAD data and spectrum.</p>

### 5.4.1 General overview

The **general overview** gives an overview of the recorded data, the executed functions and the collected fractions.



Fig. 32: Repository Window, general overview in the reconstruction

At runtime, the steps defined in the method are processed sequentially, whereby a command step always runs in the process time and a timetable step in the timetable time. This can be the process time or volume. To display the data, which potentially refer to different time bases, they are grouped. The grouping combines all sequential command steps into one graph. Timetable steps are displayed separately as a single graph.

Fig. 32 shows a method consisting of one time table step. The functions are indicated by vertical lines with a rectangle. Each rectangle contains a code that stands for the executed function. When the mouse is moved over the function rectangle, a tooltip displays more detailed information about the executed function. Rectangles of method functions are differently colored than rectangles of event functions, i.e. a function that was triggered by an exceeded threshold value. You can customize these colors in the program settings (see chapter 8 on page 71).

The vertical text above the functions is called annotation and describes the connection to the collected fractions. If a function leads to the collection of a fraction or the end of the collection process, this is indicated by an annotation text. Furthermore, the user can program annotations in the method that will be displayed later in the chromatogram.

The graphs can be interacted with as follows:

Action	Result
Drag & Drop on the axis	Moving the scale
Mouse wheel + cursor on the axis	Zooming the scale
Ctrl + Drag & Drop on the chart area	Shifting all axis
Shift + Drag & Drop on the chart area	Zoom all axis
Mouse wheel + cursor in the chart area	Shift of all X-axis
Shift + mouse wheel + cursor in the chart area	Zoom all axes in cursor direction
Right mouse button	Context menu
Space bar	Show point values

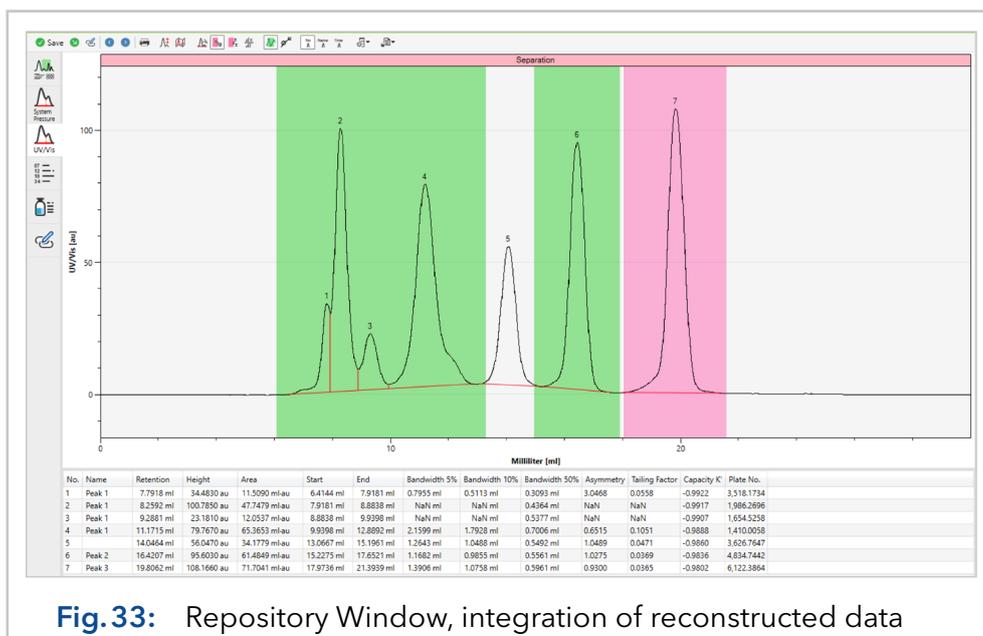
Additionally, the command bar offers the following actions:

	<b>Scaling:</b> Opens a menu for scaling the axes and setting the color of each channel.
	<b>Unfold:</b> Shows all data related to process time. This results in a single graph summarizing all steps.
	<b>Isolated Step View:</b> Shows and hides steps in the graph.
	<b>Toggle rack Visibility:</b> Shows or hides fraction rack.

	<b>Toggle Sample Window Visibility:</b> Shows or hides peak windows.
	<b>Function Visibility:</b> Shows or hides functions and events.
	<b>Show only event functions:</b> Hides in combination with an activated Function Visibility button the method functions so that only the events of the run are shown.
	<b>Annotation Visibility:</b> Shows or hides annotations.
	<b>Toggle Gradient Visibility:</b> Shows and hides the solvent gradient.
	<b>Show Value Fraction Table:</b> Opens the value fraction table.
	<b>Add all non-empty fractions to the value fraction table:</b> Adds fractions of the run to the value fraction table that could summarize fractions of several runs.
	<b>Show processing time differences:</b> Displays the communication delay between the actual measured data and their appearance in the graphical user interface (GUI).
	<b>Show slice starvation:</b> Displays the relation of process time to slice time and shows therefore how many datapoints are recognized.

## 5.4.2 Integration

For each data channel that is marked as integrable in the Configuration application part (see chapter 6.4 on page 54), the section **Integration** is available in the reconstruction view. The content of this section consists of a graph analogous to the one in the overview, but with only one data channel. This section is used for the integration of peaks and the integration results are displayed below the graph in a table.



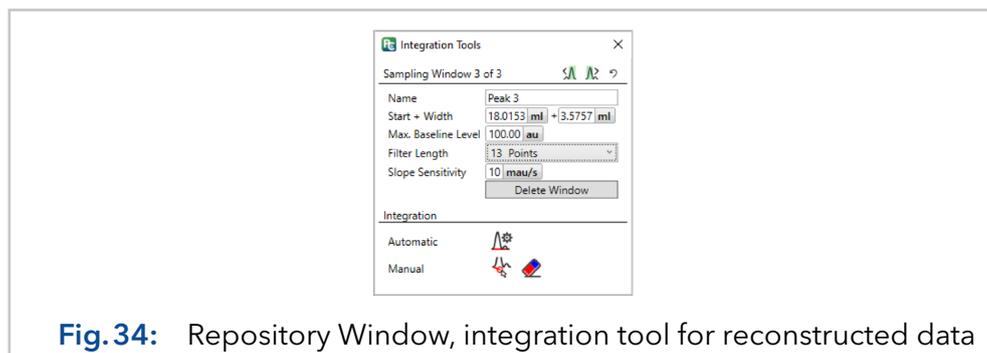
**Fig. 33:** Repository Window, integration of reconstructed data

The following options are available in the command bar:

	<b>Scaling:</b> Opens a menu for scaling the axes and setting the color of each channel.
	<b>Unfold:</b> Shows all data related to process time. This results in a single graph with all steps.
	<b>Show Integration Tools:</b> Opens the integration window.
	<b>Add Window:</b> Enables the drawing of a new peak window via mouse cursor.
	<b>Add Peak windows in Lazy Mode:</b> Searches for and adds peak windows automatically.
	<b>Show First Derivative:</b> Shows or hides the first derivative of the graph for easy recognition of peaks.
	<b>Toggle Window Visibility:</b> Shows or hides peak windows.
	<b>Toggle Data Point Visibility:</b> Shows or hides the data points of the graph.
	<b>Toggle Number Visibility:</b> Shows or hides peak numbers as peak annotations.
	<b>Toggle Name Visibility:</b> Shows or hides peak names as peak annotations.
	<b>Toggle Time Visibility:</b> Shows or hides retention times as peak annotations.
	<b>Columns Selection:</b> Enables the configuration of the results table.
	<b>Integration Default:</b> Enables configuration of integration parameters for the channel.

The first integration takes place for all integrable channels after the method process has been completed using the standard integration parameters. The temporal adjustment of the integration parameters is done via the peak sampling time table function of the procedure. Editing the peak windows and inserting new peak windows is done in this integration section.

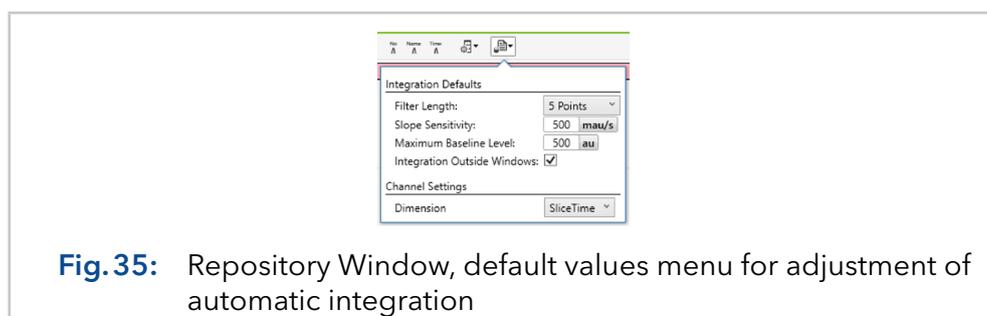
Inserting a new peak window is done with the **Add Peak Window** button in the control bar. Alternatively, the peak windows can also be inserted automatically via the function **Add Peak Windows** in Lazy Mode. The peak window parameters are set via the integration tool, which can be opened via the control bar.



**Fig. 34:** Repository Window, integration tool for reconstructed data

The name of the peak window is applied to the peaks found in the window. The boundaries of the window can be adjusted both graphically and via the input **Start + Width**. The parameter **Max. Baseline Level** determines the signal value above which no peak-ends are detected. The **Filter Length** defines the number of data points from which the first derivative is calculated. The **Slope Sensitivity** is a threshold value in the first derivative. If a certain number of points are above or below this threshold, a peak-start or a peak-end is detected. The unit of slope sensitivity is  $10^{-3}$ ·Points/Second.

Before PurityChrom 6 can calculate the peak-specific values in the results table, the peak boundaries have to be detected (integration). Peak detection is done in automatic mode via the “slope sensitivity” threshold, which in turn refers to a time dimension in the data. The process time, the slice time or the chromatogram time can be used as time dimension (see “Concepts” in the appendix for more information). The default setting for the integration dimension is made in the channel configuration and can be adjusted here via the **Default values menu** .



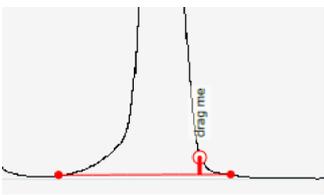
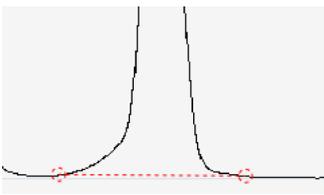
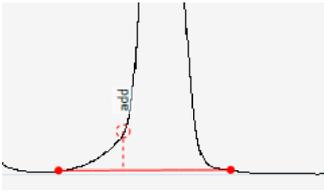
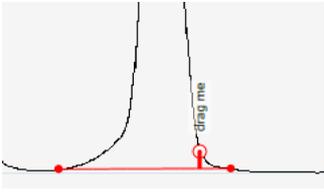
**Fig. 35:** Repository Window, default values menu for adjustment of automatic integration

In the case that the chromatogram time is used as the integration time basis and the chromatogram is further processed in volume, the displayed unit of the slope sensitivity in the user interface is not correct because it is now no longer “per second” but per volume. However, this is a special case and should not be used in practice.

It is strongly recommended to use only the slice time for the integration dimension. If a slice time is not available, the analysis time can be used as an integration dimension.

It is important to understand at this point that the integration dimension is only used for peak detection/baseline detection. The calculation of the result table is based on the previously detected baseline with the slice time and the chromatogram time. You can switch between the display of both results with the **Unfold** function (unfold active = slice time; otherwise chromatogram time). This type of calculation is possible because a slice contains several time dimensions.

The baseline can also be edited or created manually. To do this, **Enable manual baseline editing**  has to be activated in the integration tools window. If the mouse is then moved over the chromatogram, the following actions are possible:

Action	Figure	Description
Left-click on a free area		Create Peak Start
Left-click on a free area after a peak start		Create Peak End
Escape		disable manual baseline
Right mouse button on baseline or valley		Open context menu; Delete
Click on baseline area		Create Valley
Drag & Drop on Start, End or Valley		Move point or valley

### 5.4.3 DAD spectrum

The **DAD Spectrum** section will be available if a DAD Detector is operated, and a spectrum channel is configured. In this section the spectrum can be reviewed. In the upper window the spectrum is depicted over the whole runtime as 2D plot.

By left clicking the cross hair appears and shows the wavelength and intensity at the marked position.

In the lower left window, the whole spectrum at the marked time is shown. The lower right window shows the chromatogram for the marked wavelength over the whole run.

By right clicking the current view can be saved as PDF or printed via the Option "Print plot".



**Fig. 36:** Repository window, DAD spectra overview

#### 5.4.4 Documentation

The section **Documentation** shows the completed checklist, general information about the process and the audit trail of the run. The audit trail is further divided into the documentation of the analysis and the post-processing. In the analysis part, manual interventions and executed functions are documented. In the post-processing section, reconstruction activities like saving manual integration results and given signatures are documented.

The audit trail is structured in such a way that the entries are grouped by steps. For each step/group, the start process time is given. The entries within a step start again from zero. Furthermore, the documented system status can be viewed per step via **View System Status**.

During method execution, the following events are listed in the documentation section:

- Function execution
- User interventions
- System interventions (e.g. pause due to a unit error)
- Results of the method command **Continue-Marker**
- Online moving of time table entries

Executed functions are further classified into event and background functions. Event functions are triggered by an event, such as exceeding a threshold or manual device control. Background functions are triggered by a proxy. A proxy is, for example, the collection unit, which in turn forwards functions to the linked module.

The following options are available in the command bar of the documentation section:



**Show background functions:** Enables/disables the display of background function such as a command that was forwarded by the collection unit.



**Show only event functions:** Filters the run documentation by events such as an exceeded threshold or manual device control.

### 5.4.5 Documentation snapshots

This section is only visible in a result file if the command **Documentation\Page Snapshot** was used in the method.

The Documentation snapshot summarizes the device stati at the time point in the method, when the command **Documentation\Page Snapshot** was programmed. The content of the documentation snapshot is defined in the system configuration in the section System Documentation (see chapter 6.9 on page 58).

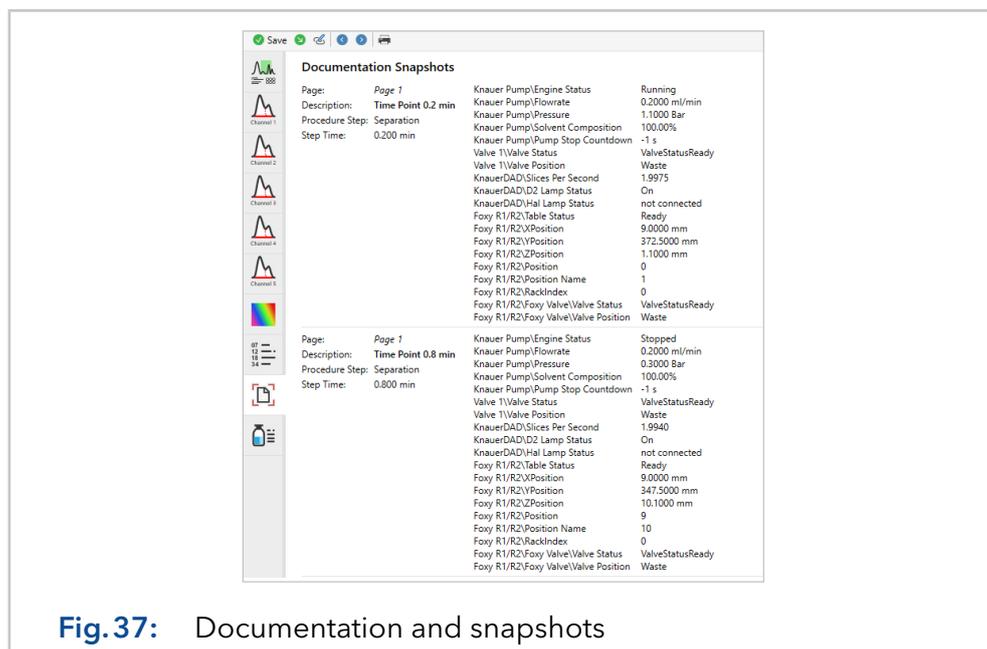


Fig. 37: Documentation and snapshots

### 5.4.6 Status recording

This section is only visible in a result file if the command **Documentation\Status Recording** was used in the method.

In the upper part of the status recording section, the system visualisation is shown with a grey bar representing the timeline of the method. The blue lines within the timeline represent the moments in which the stati of all devices were recorded. With a left mouse click on one of the blue lines, the recorded stati of all devices will be displayed in a pop-up window.

The parameter set in the Status Recording function specifies how many individual status changes (e.g. pump pressure) are recorded in the set intervall. Lower numbers equal more data points while the value 0 means that all data are to be recorded.

Furthermore, individual graphs can be displayed in the lower part of the status recording section. Select one of the available data traces or device stati and use the **+Add Data** button to visualise the respective graph.

The distance between each data point in the graph or each blue line in the timeline depends on the chosen resolution for the command **Documentation/Start Status Recording** in the method. Via the **Reset Graph** button, the graph can be removed.

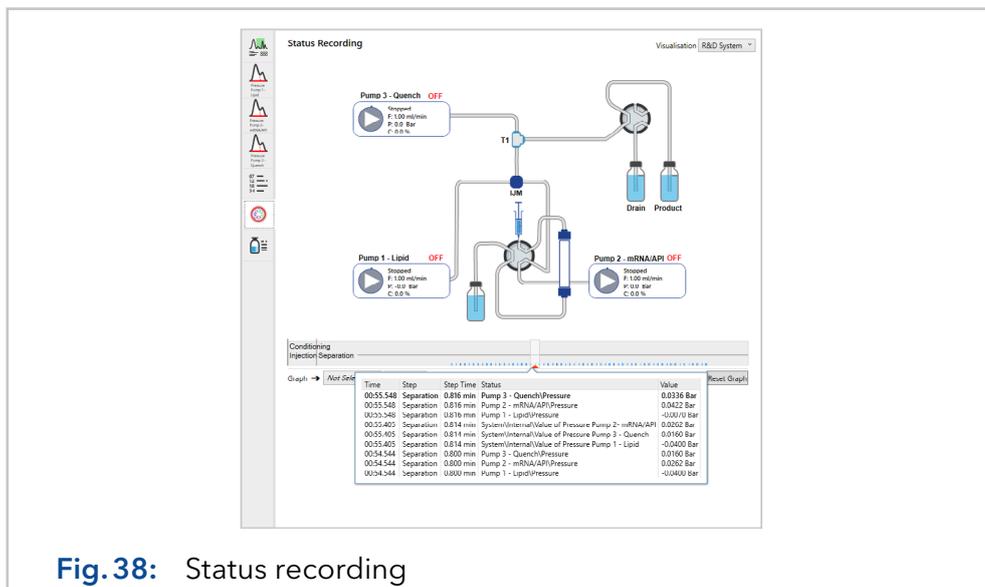


Fig.38: Status recording

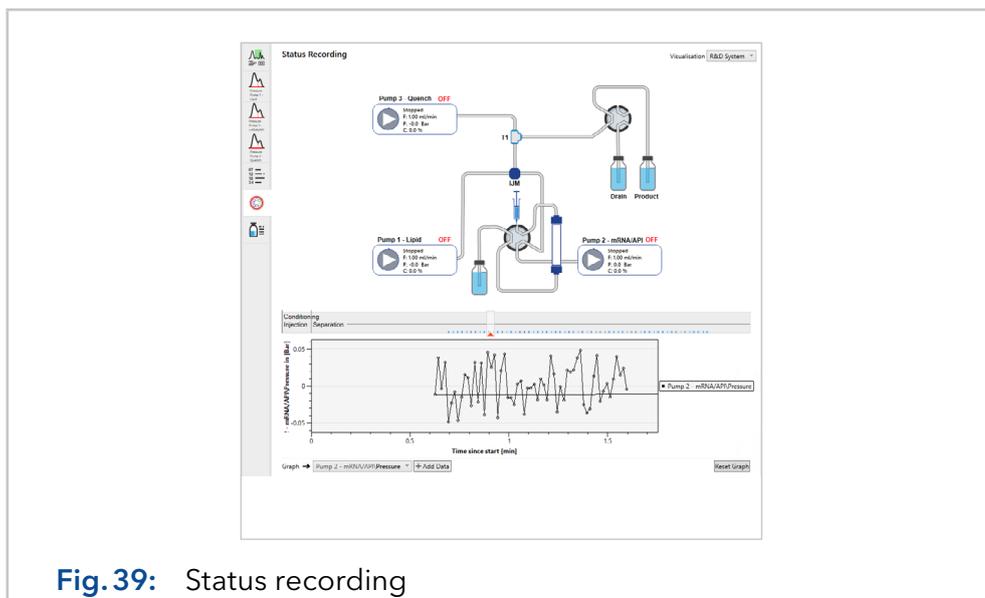


Fig.39: Status recording

### 5.4.7 Signature history

The result file can be signed by a user in the reconstruction menu via the button  (Sign). All signers are listed in the signature history section.

During a signature a comment can be added. Furthermore, the user login password restricts adding further signatures to authorized users. This section will be only visible after the first given signature.



**Fig.40:** Repository Window, verified signatures in the signature history

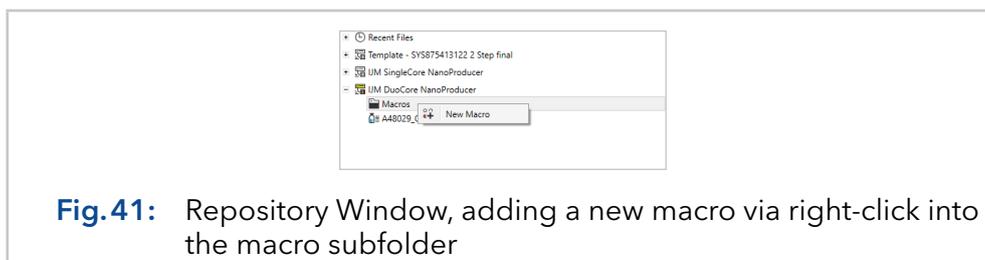
A signature entry contains the signature time, the user Id, a description that was given during signing, a Hash-value (SHA256) for selected parts of the result file, and a redocumentation of the result parts that were used for calculating the hash values.

To check the integrity of the data, the hash value is again calculated over the relevant data and compared with the stored data. If the calculated and stored hash values match, the result file is considered as verified in the **Integrity** column.

A result file can be signed as validated, in which case it cannot be saved again. Non-validated result files can be re-saved by users with the permission **save signed reconstruction documents**. If a result file is re-saved after an executed signature, the result file changes and the executed signatures are considered invalid (Integrity: Failed). Since resigning saves the result file, previous signatures are naturally considered invalid.

## 5.5 Editor: Macro

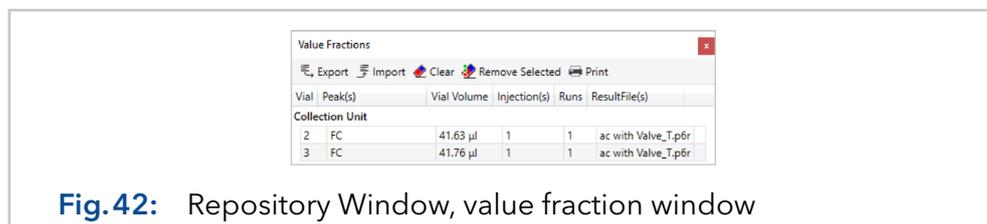
If a time table is used repetitively as a part of a method (e.g. pre-flushing of the system or a washing step for the column), this time table can be saved as a macro via the button  (Save the method as a macro) in the title bar of the method editor. Note that all time table steps included in a method are saved as macro. All macros are displayed in the **Macro** subfolder of the respective system configuration in the file tree. The macro editor is structured in the same way as the procedure editor of the method. A new macro can be created in the context menu of the macro folder available via right-click on it.



**Fig.41:** Repository Window, adding a new macro via right-click into the macro subfolder

## 5.6 Editor: Value fraction

There is the possibility that a collection unit is filled with fractions via several method executions. After a sequence was processed, an overview of selected fractions can be created with the value fraction window, which is shown below.



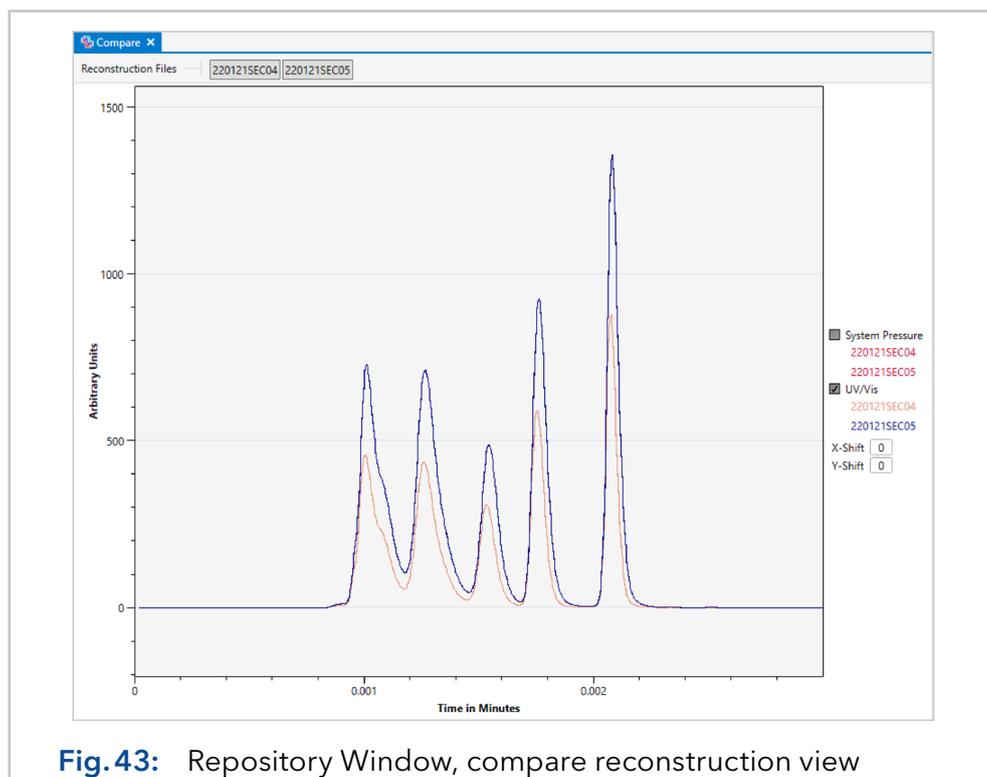
**Fig.42:** Repository Window, value fraction window

The value fraction window can be opened via the repository or the reconstruction editor with the button  (Show Value Fraction Table). Fractions can be added from the overview of the result file (Menu, Collector Rack or Fraction Table) via the button  (Add all non-empty fractions to the value fraction table).

## 5.7 Editor: Compare chromatograms

With the **Compare Reconstructions** window, the data channels of different result files can be compared with each other. This window can be opened via the button  in the **Repository** menu.

The user can select several result files from the file tree, use a right mouse click and choose the option **Add to compare** to generate the overlay view of these files. For each result file, a button is displayed above the chromatogram view enabling the selection of individual data channels.



**Fig.43:** Repository Window, compare reconstruction view

Via the legend on the right side, entire channel comparisons can be shown or hidden. With a click on the name of the result file, the colors of the graphs can be adjusted. With **X-** or **Y-shift** the data channels can be displayed shifted accordingly.

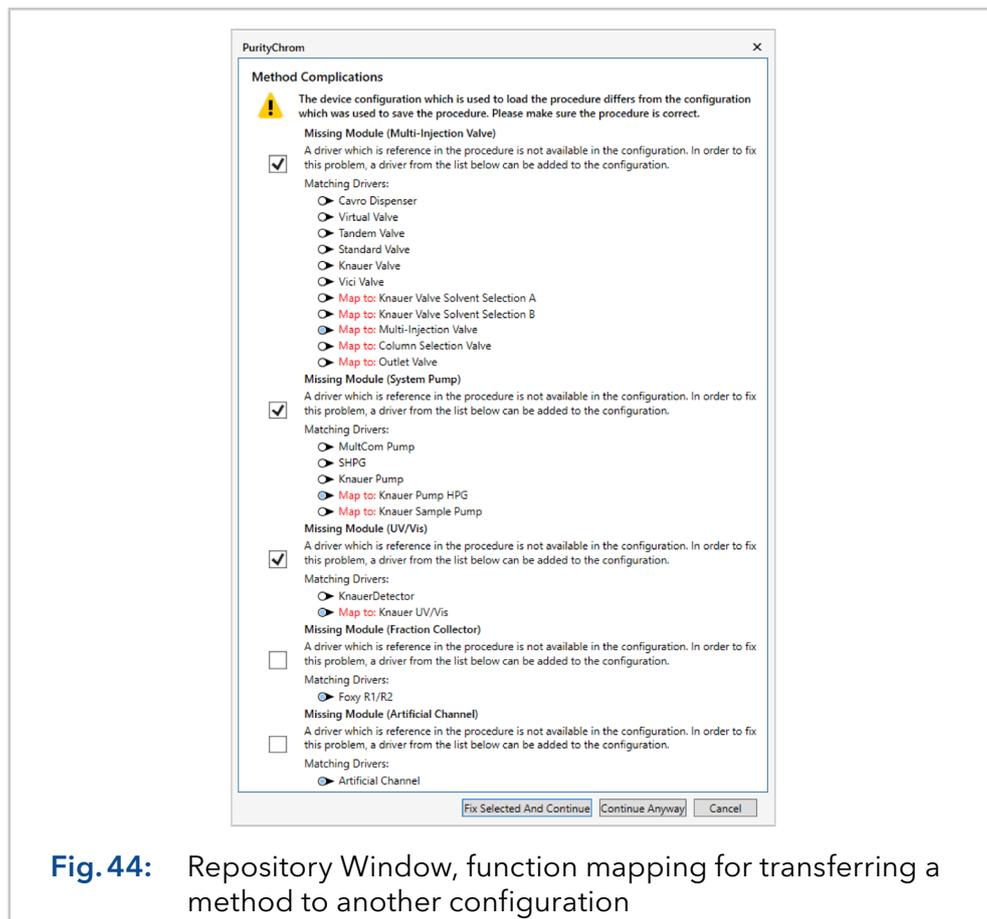
The graphs can be interacted with as follows:

Action	Result
Drag & Drop on Axis	Panning the scale of the respective axis
Mouse Wheel + Cursor on Axis	Zooming the scale of the respective axis
Ctrl + Drag & Drop on the Chart Area	Panning all axis
Shift + Drag & Drop on the Chart Area	Zooming all axis
Mouse Wheel + Cursor inside the Chart Area	Panning all x-axes
Shift + Mouse Wheel + Cursor inside the Chart Area	Zoom of all axes in cursor direction (only available in unfold modus)
Space-Key	Show point values

Via a right-mouse click into the compare window, the context menu will open. Unzooming and printing the **Compare Reconstruction** are available via this context menu.

## 5.8 Data exchange: Mapping & archives

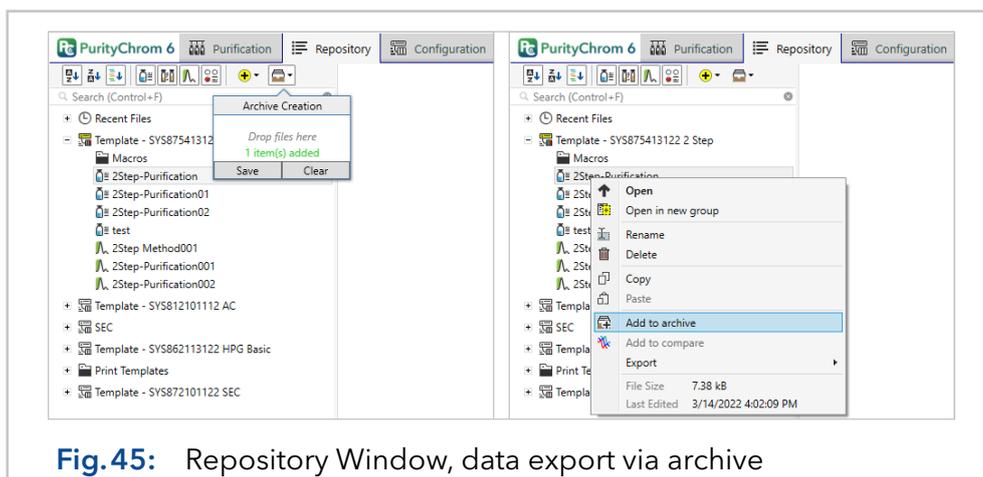
Methods, sequences and macros in PurityChrom 6 are dependent on the configuration. If one were to simply copy a method file from the file storage and copy it into the data directory of a foreign configuration, the method would be opened accordingly with this foreign configuration. This would have the consequence that the device function contained in the method would not be found at first and a function **Mapping** has to be carried out.



**Fig. 44:** Repository Window, function mapping for transferring a method to another configuration

Result files are an exception, as all relevant information are documented in them. Result files can be easily opened via the repository menu with the option **Open extern reconstruction** via the button .

If data is to be exchanged, it is recommended to use the archive function of the repository. Methods, sequences and result files can be simply dragged & dropped onto the archive, which is located in the title bar behind the icon . An alternative to adding files to an archive is via a right mouse click onto the respective data in the file tree and the option **Add to Archive**.



**Fig.45:** Repository Window, data export via archive

Click on the archive symbol to open the list of files, then choose **Save** to generate the archive. Archives that have already been created can be opened again by clicking on  and choosing the option **Open Archive**.

To transfer also the related configuration into the **Configuration Window**, use a right-mouse click on the opened archive file and select the option **import archive data**. Afterwards, the user has to define a data path directory for the archive data and after a restart of the software, the configuration of the imported archives can be seen and modified in the **Configuration Window**.

## 5.9 Editor: Print template

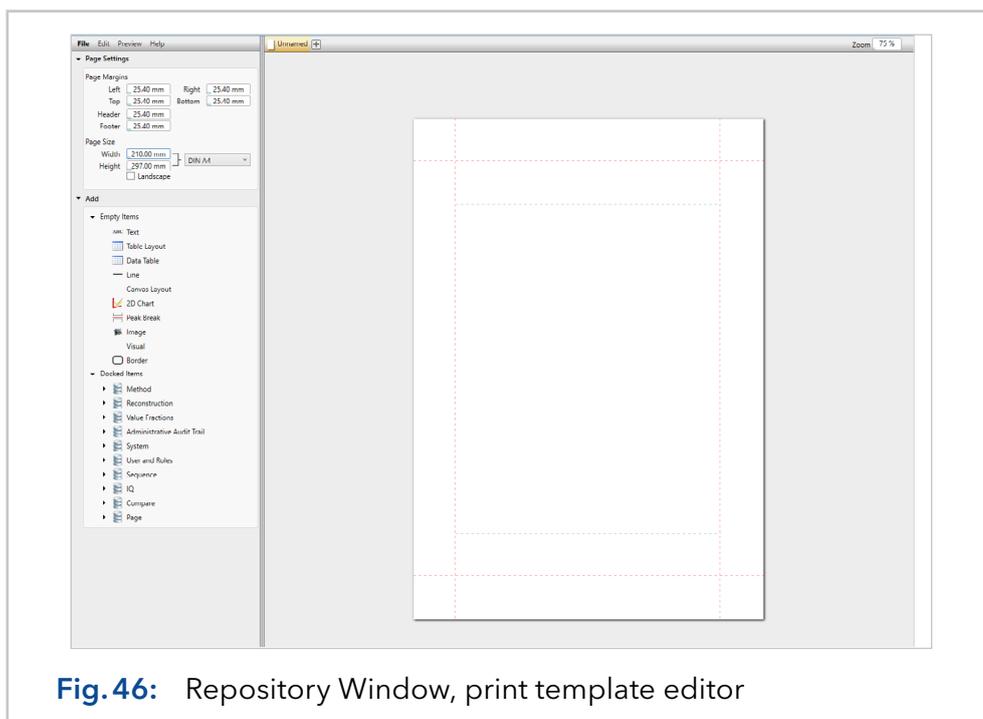
In general, printouts in PurityChrom 6 are made using print templates. The print template describes the visual presentation for printing selected data. Which data is provided in a template is determined by the file type to be printed.

Thus, a method printout of the template provides the corresponding method data, the reconstruction printout correspondingly provides the reconstruction data, etc. With the installation, some print templates are delivered as default templates that can be customized using a template editor.

Print templates are stored in the template directory chosen in the **Software Settings**  (see chapter 8 on page 71). The content of the set template directory is displayed in the **Repository** in the subfolder **Print Templates** of the file tree. New templates can be created, like other files, simply by using the context menu of the **Print Templates** folder available via a right-mouse click.

### 5.9.1 Basics of the editor

Double-click on one of the templates opens the template editor. On the left side of the editor, there are the page settings, as well as the insertable elements. On the right side, there is the design area with drawn spacing and subdivisions. A print template can describe several prints (sub-reports). Each sub-report has its own page numbering.



**Fig. 46:** Repository Window, print template editor

To insert elements, they are dragged and dropped onto the design area. New elements can be placed in the different page parts (page header, page body and page footer) or within other elements (e.g. table layout). Within the page parts, the elements are arranged sequentially vertically. The position of already inserted elements can be changed by drag and drop. For a horizontal element arrangement a layout has to be used (e.g. table layout).

## 5.9.2 Design elements

The following elements are available to design the template:

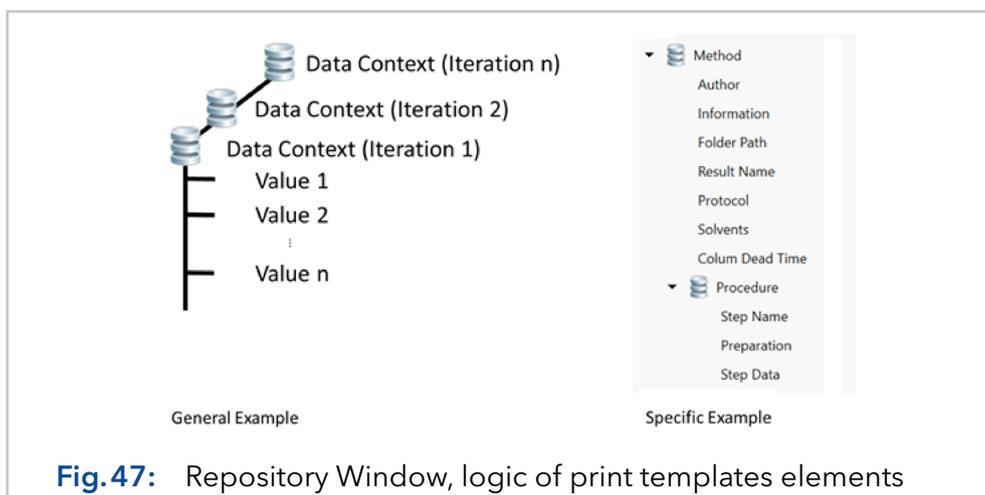
- **Text:** Display of formattable text. The values of an adjacent data context can be inserted as placeholders
- **Table layout:** Providing a layout table with configurable rows and columns
- **Data table:** Display of data tables (e.g. the results table)
- **Line** Display horizontal or vertical lines
- **Canvas layout:** Layout of elements, where the position is determined by the element distances to the layout borders
- **2d chart:** Display of data tracks
- **Page break:** Inclusion of a page break
- **Image:** Inclusion of an image file
- **Visual:** Display of an element provided as a graphic
- **Border:** Outlines an element placed inside

In order for elements to display data, they have to be linked to the corresponding data first. This data link is a general property of each element and is controlled via the "Value Path" property. The processing/display of the linked data is the responsibility of the respective element.

Data is provided to the template editor via data contexts. Each data context contains values (e.g. text values, data tracks or tables) as well as further subordinate data contexts.

Another property of data contexts is iterability. Iterability describes that there are different versions of a context, each containing different values.

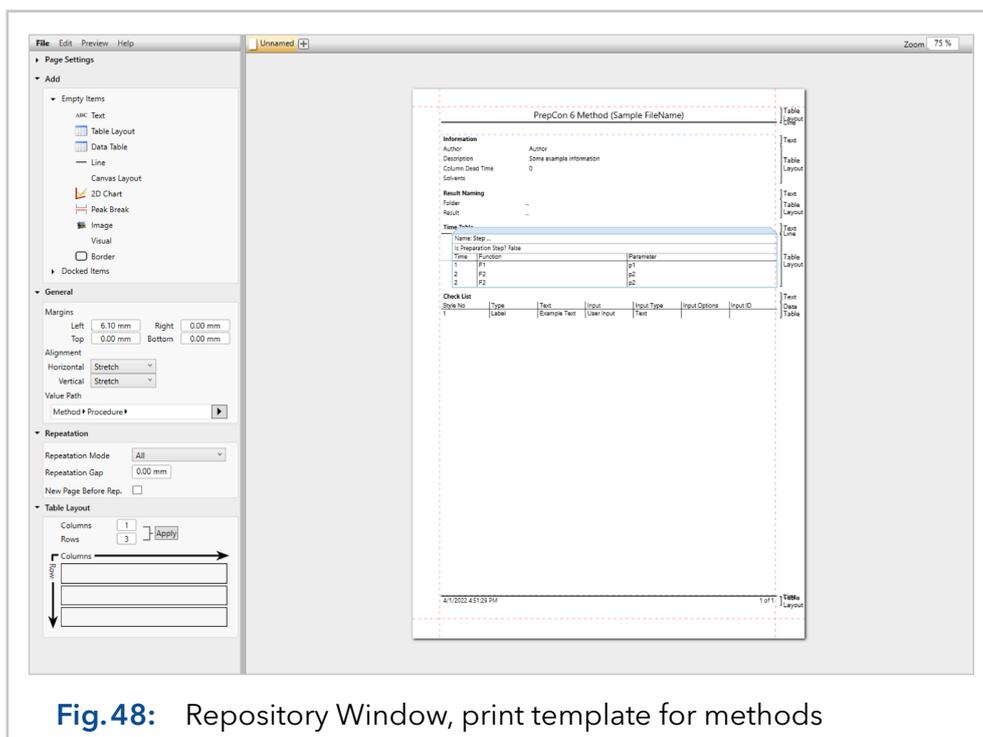
A data context contains the values 1 to n, where this data context has k-iterable versions. Another concrete example represents the data context of a method. The data context of the method has first value elements like the author, information and the checklist. Next, the method has a sub-data context for the procedure. The procedure is a sequence of n-steps. These n-steps are represented in the procedure data context by the iteration i.e. the first time-table step is iteration 1, the second corresponding iteration 2 and so on.



**Fig.47:** Repository Window, logic of print templates elements

If all method time table steps are to be mapped now, iterations have to be performed through the procedure context. Iterations are realized by the **Repetition** property of layout, which bind the respective data context. The layout is then inserted repeatedly according to the iteration of the data context when printing.

Fig.46 on page 49 shows a section of the standard method template. To print each procedure step, a table layout is first used and this is bound to the method procedure sub-context. The table layout contains three rows for the values "Step Name", "Preparation" and finally the procedure "Step Data". Repetition Mode: All" repeats the table layout for each procedure step. To open the table layout, the mouse should stand over the table and the key **1** has to be pressed on your keyboard as the mouse over text indicates.



**Fig. 48:** Repository Window, print template for methods

If the **Value Path** property refers directly to the basic data context (e.g. method, reconstruction or value fractions), it is referred to as an absolute value path. If this is not the case, it is a relative value path, which refers to the data context of the parent element (e.g. a layout).

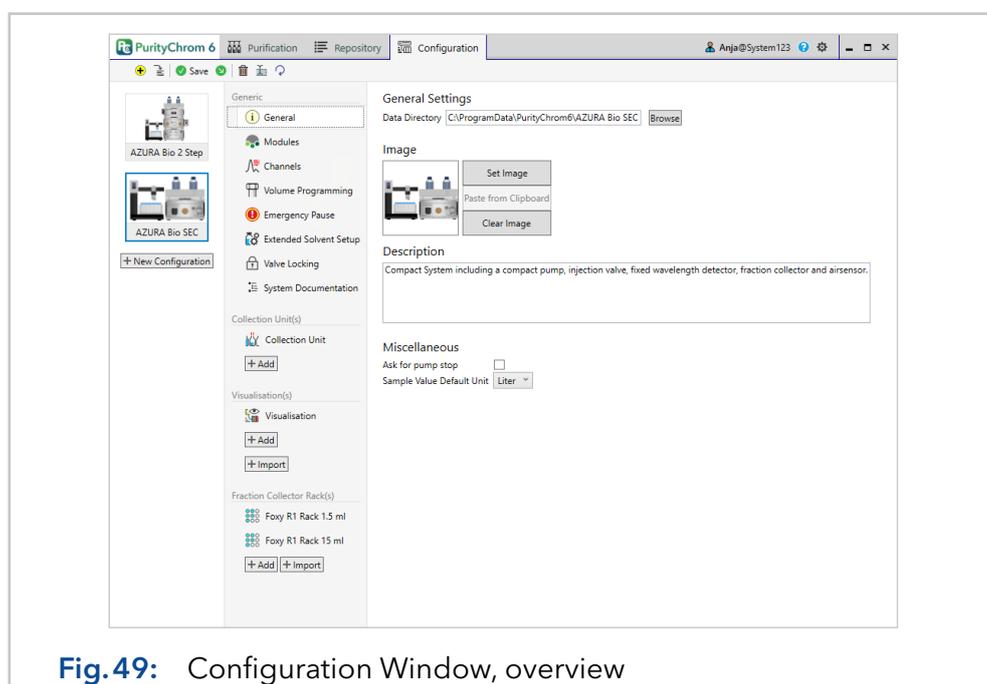
If the **Repetition** property is used with a layout, i.e. if the layout is to be repeated according to the iterations in the data context, the elements in the layout itself have to use a relative value path. Otherwise, the layout would be reinserted according to the iterations, but the data would always refer to the absolute path (i.e., the data context in iteration 1) for absolute paths.

## 6. Application part: Configuration

The **Configuration** application part of the software is used to configure the system that has to be controlled. Usually, only one configuration is used for the system to be controlled. Nevertheless, PurityChrom 6 can manage several configurations. However, since only one system can be controlled per time, the use of several configurations only makes sense if the system is used in different ways, e.g. if heating devices should not be used every time or a valve is used one time as inlet and one time as outlet valve.

### 6.1 Introduction to the editor

The Configuration window lists on the left side all configured systems, in the middle the steps necessary to configure a system, and on the right side, all settings made in the respective step.

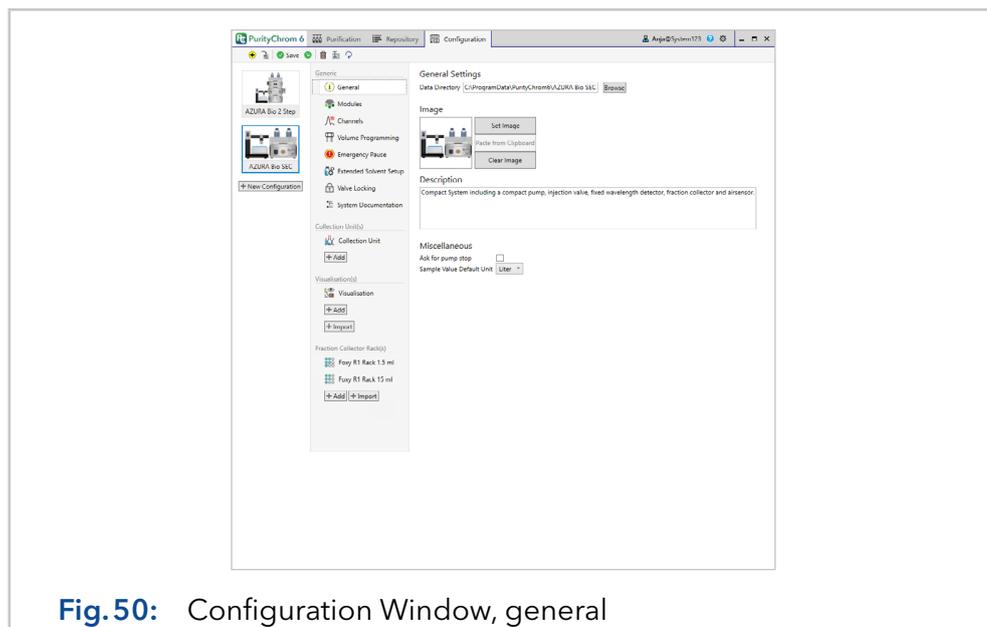


**Fig.49:** Configuration Window, overview

The title bar of the **Configuration Window** offers the following actions:

	<b>Create a new configuration:</b> Creates a new configuration.
	<b>Use Configuration Templates:</b> Creates a new configuration from a configuration template.
 Save	<b>Save:</b> Saves selected configuration.
	<b>Save as:</b> Saves currently selected configuration under a different name.
	<b>Delete:</b> Deletes currently selected configuration.
	<b>Rename:</b> Renames currently selected configuration.
	<b>Discard:</b> Discards/Resets all changes to the selected configuration.

## 6.2 Configuration part: General



**Fig. 50:** Configuration Window, general

All data associated with the configuration, such as methods, sequences, and result files, are stored under the given data path under **Data Directory** in the **General** tab. Choose a local location on your computer here. Using networks or servers is only recommended when working with a network or server supported by PurityChrom 6.

The image of the configuration is used on the start page of the **Purification** application part and will be also displayed in the list of configurations given on the left side of the **Configuration Window**.

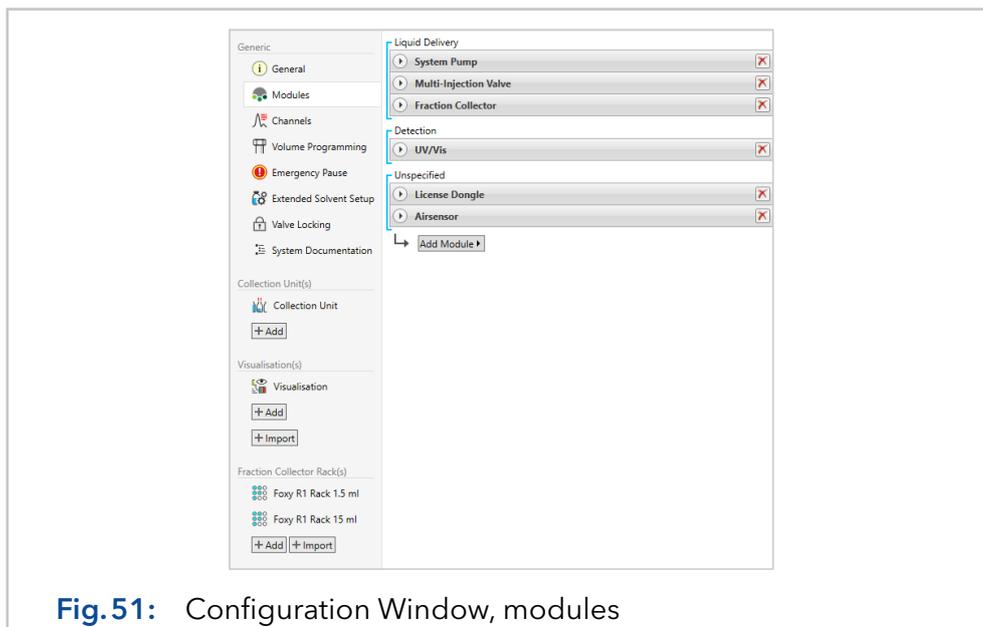
A short description of the configuration can be entered in the Description field.

Setting the check box **Ask for pump stop** results in a confirmation prompt in the **Purification Window** before the pumps are stopped by the command "Stop-All". Under **Sample Value Default Unit** the default unit that should be displayed for injection volumes in the sequence table and the schedule can be defined.

## 6.3 Configuration part: Modules

Modules are the individual components that provide PurityChrom 6 with functions. To use modules in PurityChrom 6, they have to be added to a configuration first. The modules are added to a configuration under the menu item **Modules**.

By clicking on the **Add Module** button, the available modules are listed. Another click on a module in the list adds it to the configuration.

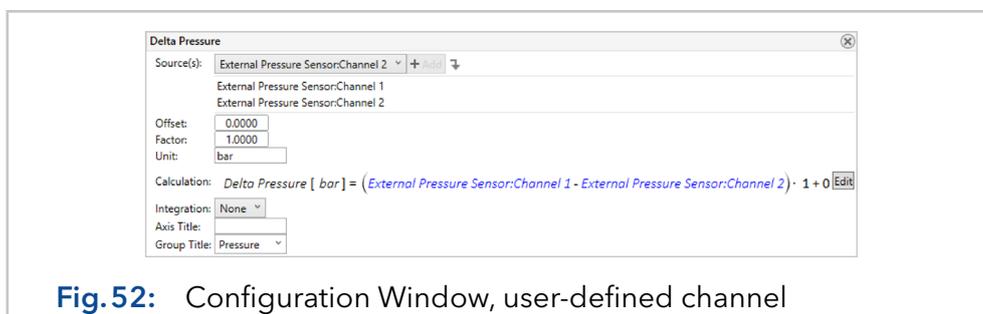


By clicking on the name of a module, its name can be edited. Details on the exact configuration of a unit can be displayed by clicking on . Clicking on removes the module from the configuration.

If the license requires a dongle, a matrix dongle module has to be added for each configuration.

## 6.4 Configuration part: Channels

The data channels are defined in the menu item **Channels** in the configuration. Channels can be added here via the button **Add Channel** which provides a list of the available channels depending on the configured modules. PurityChrom 6 distinguishes between hardware channels and logical channels. Hardware channels (e.g. pressure monitored by a sensor) are provided by devices. Logical channels can refer to one or more hardware channels and can be used to calculate with each other as required (e.g. delta pressure calculated by pressure signals of sensors adopted before and after a column). Fig. 52 shows the configuration interface of a logical channel as an example.



With a click on the channel name, the name of the channel can be changed. Under **Sources**, one or more hardware channels can be added to the channel for further calculations. **Offset** and **factor** can be adjusted so that the signal has the desired unit. The adjustment of the **Axis Title** and the **Unit** are used to identify the axis later in the graph. If there is no axis title given, only the unit will be displayed at the axis. Under **Calculation**, a formula can be used to specify how the sources are to be offset against each other.

Several sources can be added to a channel by selecting the respective channel under "Source(s)" and clicking "Add". To remove a source, select the corresponding source and press the Del key of your keyboard. If several sources are selected, they are first summed up. The calculation formula can be adapted via a formula editor under **Edit**.

The **Integration** dimension describes the standard basis for peak detection. There are four different options for integration dimensions:

- **Analysis time:** The time of the data point at process time
- **Chromatogram time:** The time of the data point in the chromatogram
- **Slice time:** The time of the data point at its data processing (recommended for precise integration results)
- **none:** choose this option to disable integration of the selected channel

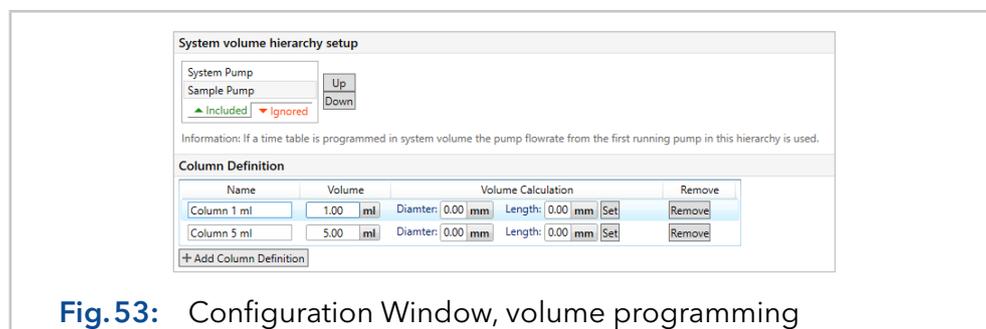
For a better understanding of the integration dimension, please refer to the concepts chapter in the appendix.

With **Group Title** several channels can be scaled to one axis. This is especially useful for a large number of channels in a system. For this purpose, write a name in the box and select the same entry for each channel that should use the same axis. The **Axis Title** or the **Unit** of the first channel of the group is used as the title for the merged axis.

## 6.5 Configuration part: Volume programming

Time table steps in the method procedure usually run in process time, but can also be programmed in volume or column volume. For a time table to be programmed in volume, this has to be defined in the configuration under Volume Programming first.

The system volume is defined hierarchically given by a list of pumps. According to this list, the hierarchical system volume is calculated from the flow rate of the first running pump. Fig. 53 shows the partial configuration of the volume programming. It can be seen that the configuration contains two pump modules, whereby both pumps are included in the system volume. The hierarchy here specifies that the flow rate of the system pump is primarily used. Only when the system pump is stopped, the flow rate of the sample pump is used to calculate the system volume.



**Fig. 53:** Configuration Window, volume programming

In addition to the system volume, a method can also be programmed in column volume, which is only a conversion factor of the system volume. To use the column volume, the used columns with their respective column volume have to be defined in the **Volume Programming**. The unit **column volume** is thus defined as ratio of system volume to the volume of the respective column.

## 6.6 Configuration part: Emergency pause

In the **Purification** application part, it is possible to execute an emergency stop, which first stops all units. The emergency stop can be triggered by the emergency stop button, but also automatically by a defined system status or a unit error.

In addition to the execution of the general stop of all units, other function executions can be configured. Programmed valve switches are a special case here, as they are switched back to the original state when the emergency stop is undone.

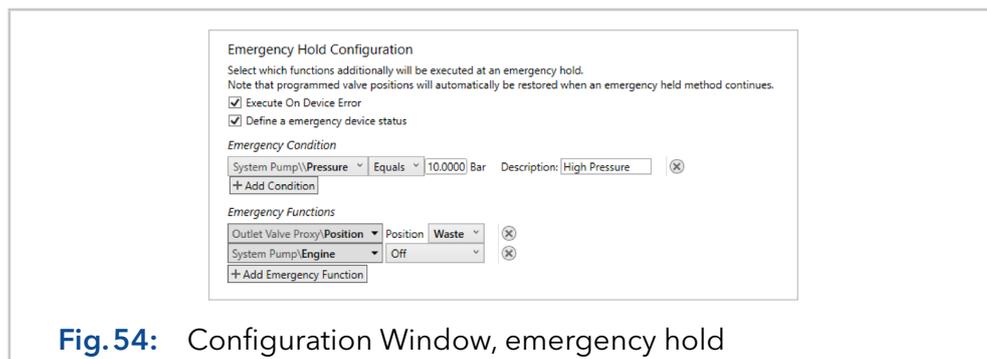
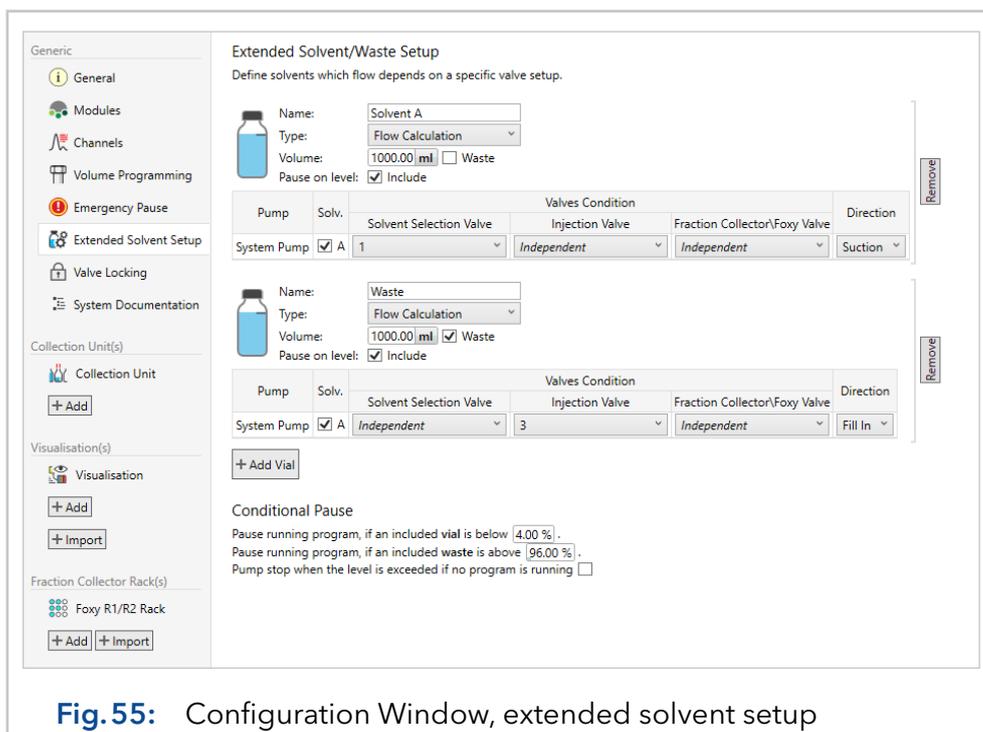


Fig. 54: Configuration Window, emergency hold

If the option **Define an emergency status device** is activated, system states can be defined under **Emergency Condition** for devices in the configuration that lead to the immediate execution of the emergency pause. If several system states are programmed, reaching a single operating state is sufficient to execute the emergency pause.

## 6.7 Configuration part: Extended solvent setup

In the **Extended Solvent Setup**, an assignment to pumps and valve positions as well as the reservoir size are defined for each reservoir and waste container. This is necessary for a correct calculation of the fill levels. Furthermore, in the lower part of the **Extended Solvent Setup**, threshold values can be defined from which running pumps are to be stopped automatically.



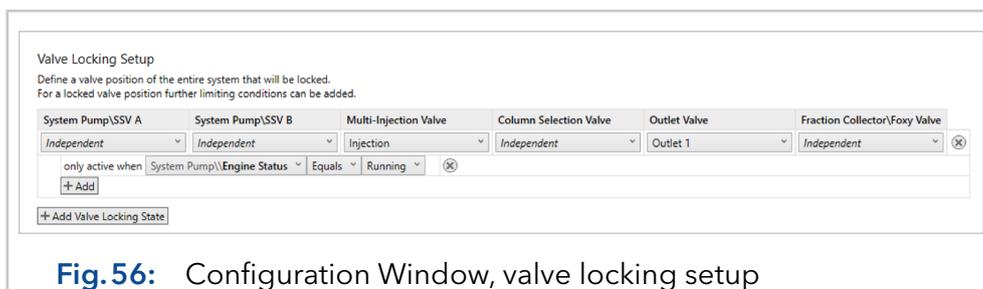
**Fig. 55:** Configuration Window, extended solvent setup

The option **Flow Calculation** under **Type** calculates the content of the container using the volumes delivered by the pumps. The **Module Status** option determines the content of the container through a connected device (level sensor, weighing plate).

**Pause on level** switches the automatic pause of PurityChrom 6 on and off when the threshold values defined under **Conditional Pause** are reached. The pump table should contain at least one pump channel for the reservoir. Under **Conditional Pause**, the minimum and maximum fill quantities in percent are defined which will pause the method automatically. Optionally, this solvent monitoring could be activated also for stopping the pumps outside of a method by choosing the option **pump stop when the level is exceeded if no program is running**.

## 6.8 Configuration part: Valve locking setup

In the **Valve Locking Setup**, certain valve positions can be prohibited. These prohibitions can also depend on the status of other devices, e.g. the status of the pump.

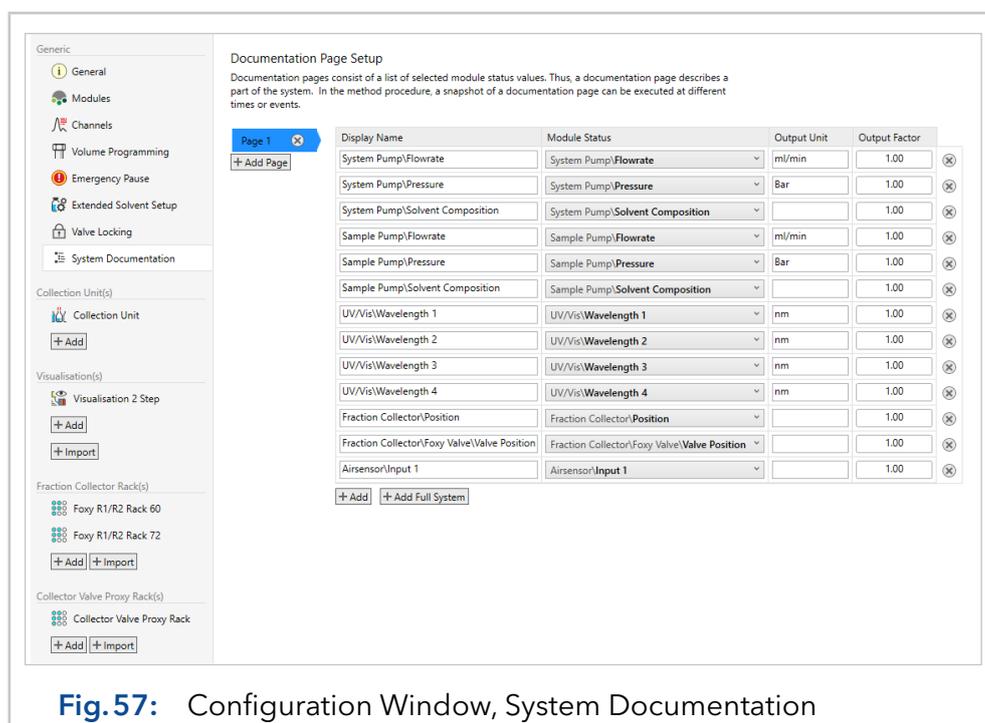


**Fig. 56:** Configuration Window, valve locking setup

## 6.9 Configuration part: System documentation

In this section, the user can define multiple documentation pages summarizing the device stati. Within a method, these pages can be used via the command Documentation\Page Snapshot to create documentation about the device stati at a certain time or event. The programmed page snapshots will be documented in the result file in a separate section see chapter 5.4.4 on page 41.

Use the **Add Page** button to define a new documentation page. The naming of this page can be changed by clicking on the default name and typing in the new one. Use the **Add** button to create a new entry for a device status or the **Add Full System** button to create a full list of device stati that should be documented on this page. In the column Module Status, a drop-down menu summarizes the device stati that are available for the respective system configuration. For deletion of a line, use the X Button on the right side of the respective line.



## 6.10 Configuration part: Collection unit

Each configuration can contain several collection units to set the fractionation conditions. Accordingly, it can be easily switched between fractionation via fraction collector, via outlet valve, or via an outlet valve combined with a fraction collector. Note, if you would like to use an outlet valve as a fractionation valve, you have to configure a **Collector Valve Proxy** in the **Module** subchapter of the configuration.

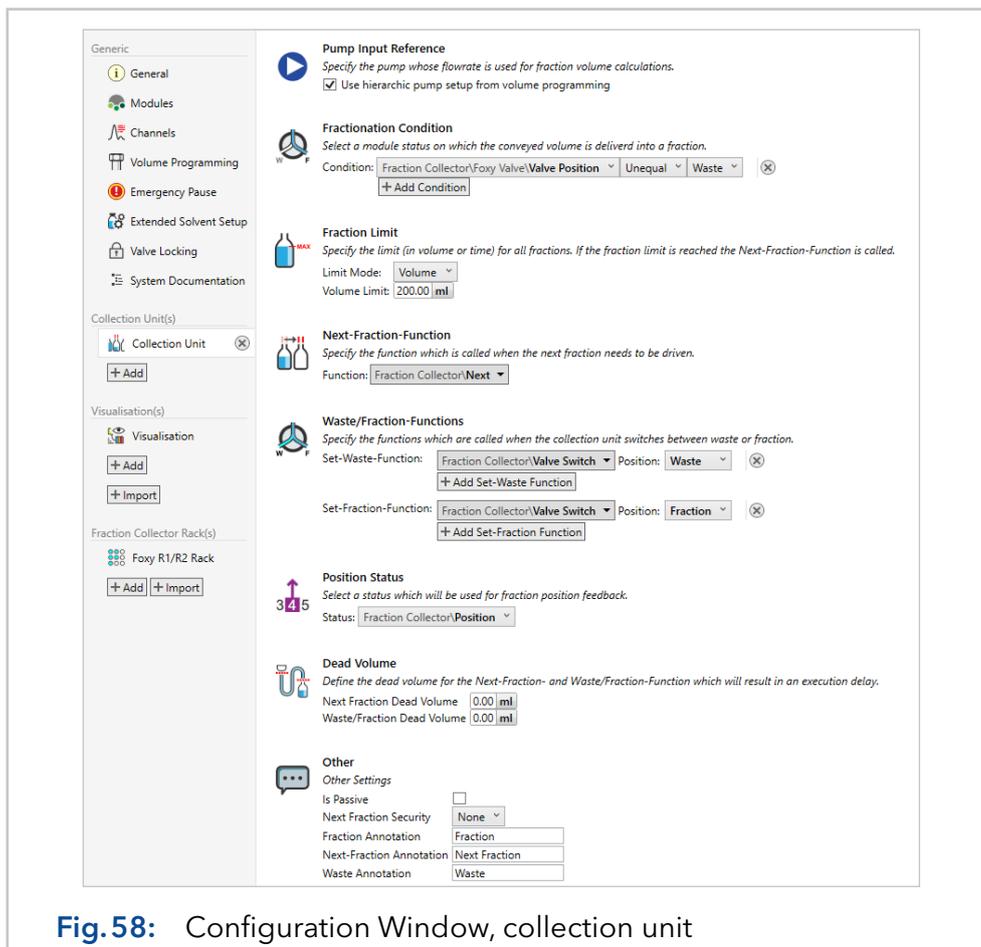


Fig. 58: Configuration Window, collection unit

Under **Pump Input Reference** a pump has to be selected whose flow rate is necessary for the calculation of the fraction quantities. Alternatively, the hierarchy from the **Volume Programming** tab can be used.

The **Fractionation Condition** specifies when the **Collection Unit** fractionates. Here, for example, several valve positions can be specified which have to be switched to be able to recognize the fractionation start. In the case of an outlet valve, the option **unequal waste position** has to be chosen.

Under **Fraction Limit** one of the following limit modes has to be chosen:

<b>Fraction Limit</b>	The maximum volume of one fraction tube defined in the used fractionation rack. A device designated as a collector has to then also be selected.
<b>Volume</b>	The maximum volume for one fraction.
<b>Time</b>	The maximum time after which the fractionation should be continued into the next tube or position.

This fraction limit is only a default value and can be changed by method settings later on.

The **Next-Fraction-Function** has to contain the action to be executed when switching to the next fraction. For example, a step of the fraction collector or the switching of a collector valve proxy.

The **Waste/Fraction-Functions** define the actions that have to be executed to change from waste to fraction and vice versa.

The **Position Status** has to contain the device and its status which defines the current fraction number.

With **Dead Volume** the dead volumes can be defined which should be considered during fractionation and result in an execution delay.

In the section **Other**, the following options can be configured:

<b>Is Passive</b>	If this option is activated, the Collection Unit can no longer be influenced by a method. (No Next Fraction, Waste/Fraction switching via the Collection Unit).
<b>Next Fraction Security</b>	This function ensures that already collected fractions are not contaminated by the current run.
<b>Fraction Annotation</b>	Text that will be annotated when switching from Waste to Fraction.
<b>Next-Fraction-Annotation</b>	Text that will be annotated when switching to the next fraction.
<b>Waste Annotation</b>	Text that will be annotated when switching from Fraction to Waste.

## 6.11 Configuration part: Visualisation

### 6.11.1 Concept

Several visualisations can be added to a configuration. A visualisation shows the various operating states of the individual devices and components in a simplified schematic representation. In addition, certain functions or commands can be executed.

## 6.11.2 Structure of the visualisation editor

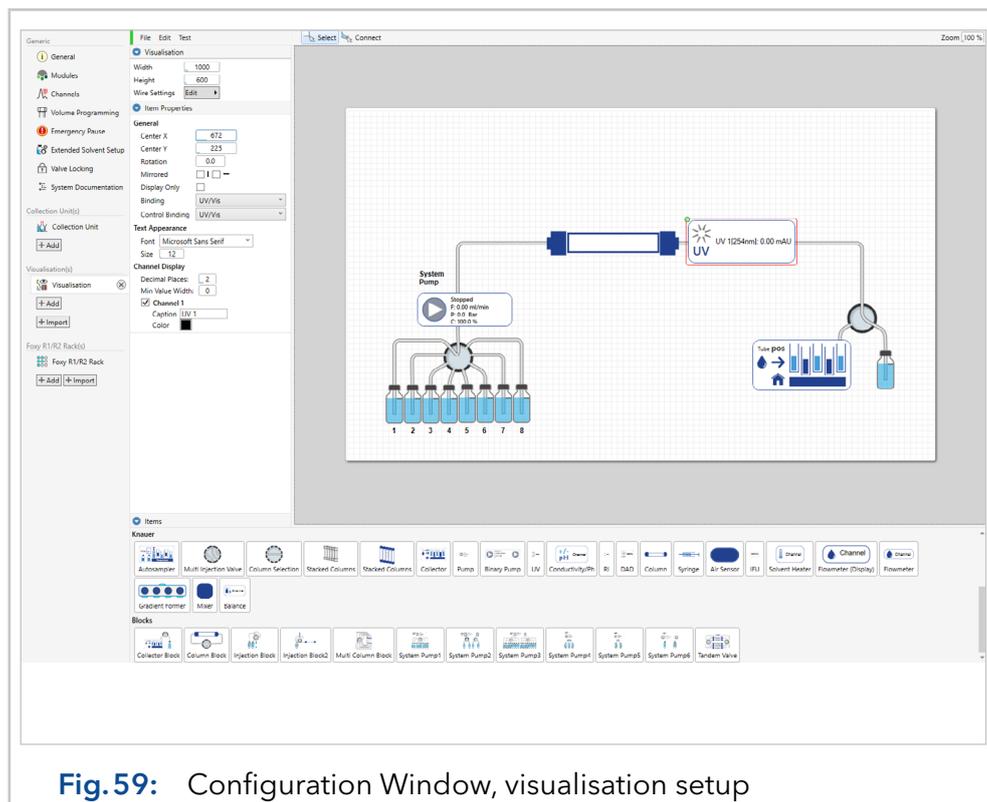


Fig. 59: Configuration Window, visualisation setup

The visualisation editor is divided into different areas. The actual visualisation area can be found on the right side of the window. There, various objects from the menu at the bottom of the window can be positioned by drag & drop and, if necessary, connected with the piping tool activated through the  **Connect** button.

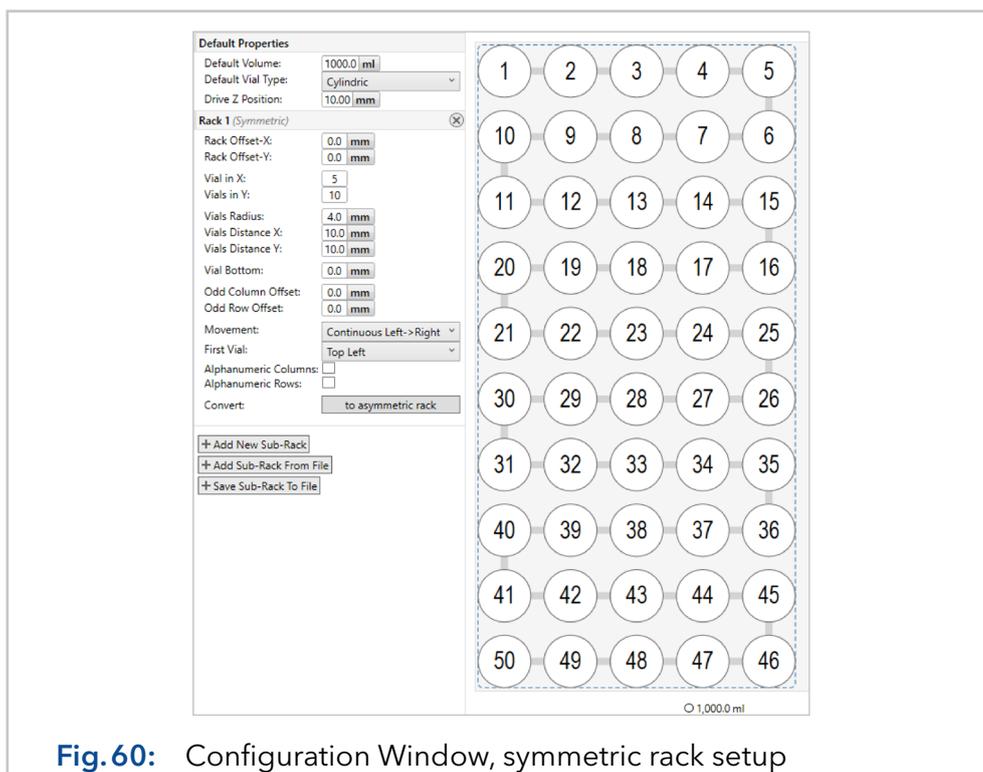
If a visualisation object positioned on the visualisation surface is selected with the left mouse button, its properties appear in the list of item properties on the left side of the window. Among other things, the visualisation object can be bound to a device from the configuration. This is necessary to assign the visualisation object to a device. The further configurable options depend on the visualisation object. For example, in the menu of a detector, you can select the data channels of the detector that should be shown in the visualisation object.

The visualisation can be tested via **Test** in the title bar of the editor. All functions are simulated here.

## 6.12 Configuration part: Collector racks

Each configuration can contain multiple fractionation racks. Which rack should be used for the next fractionation can be decided in the **Purification** window by the button  (Collector rack menu) or in a method unit via the command **load collector rack**.

Each fraction collecting device may contain one or more racks. Each rack in turn contains at least one sub-rack. Since some fraction collectors may contain multiple racks, a rack in PurityChrom 6 can be understood as the total "passable" area of a fraction collector and the sub-racks as the actual racks. Racks can be added to the configuration using the Add or Import buttons.



**Fig. 60:** Configuration Window, symmetric rack setup

In the Default Properties the following options are available:

<b>Default Volume</b>	The default volume for each tube in the sub-racks. The volume of each individual tube can also be set manually. See below.
<b>Default Vial Type</b>	The shape of the container in the rack visualisation. You can choose between cylindric and rectangular.
<b>Drive Z Position</b>	If a fraction collector has a Z-axis, a default value can be defined here for all containers in the sub-racks. This value can be set individually for each tube. See below.

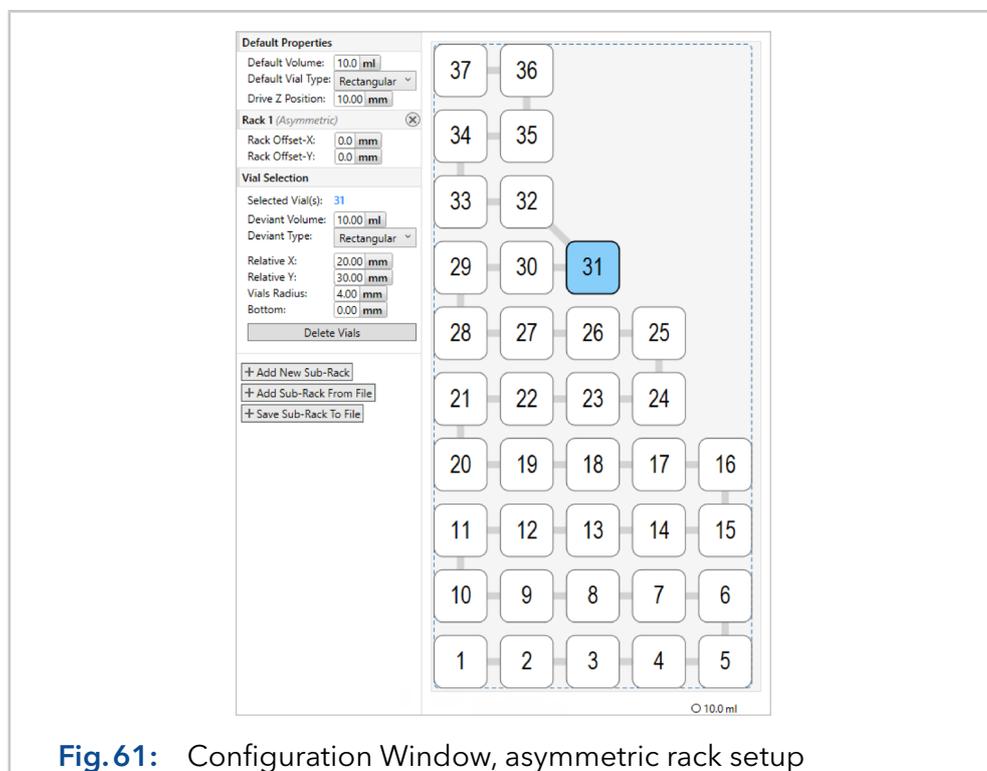
For each subrack the following properties can be configured:

<b>Rack Offset X</b>	The X position of the upper left corner of the sub-rack in the rack.
<b>Rack Offset Y</b>	The Y position of the upper left corner of the sub-rack in the rack.

In symmetrical sub-racks additionally the following options can be configured:

<b>Vial in X</b>	Number of vials in X
<b>Vial in Y</b>	Number of vials in Y
<b>Vial Radius</b>	Size of the vials inside the visualisation.
<b>Vials Distance X</b>	Distance between vials in X
<b>Vials Distance Y</b>	Distance between vials in Y
<b>Vial Bottom</b>	Z position of the mouthpiece
<b>Odd Column Offset</b>	The offset of odd columns.
<b>Odd Row Offset</b>	Offset of odd rows
<b>Movement</b>	The movement pattern of the collector through the rack.
<b>First Vial</b>	The position of the first vial in the rack.
<b>Alphanumeric Columns</b>	Columns are to be numbered alphanumerically.
<b>Alphanumeric Rows</b>	Rows are to be numbered alphanumerically.
<b>Convert</b>	Convert the symmetrical to an asymmetrical sub-rack.

By left-clicking on a vial in the rack visualisation, the maximum filling volume and the shape of this vial can be defined individually.



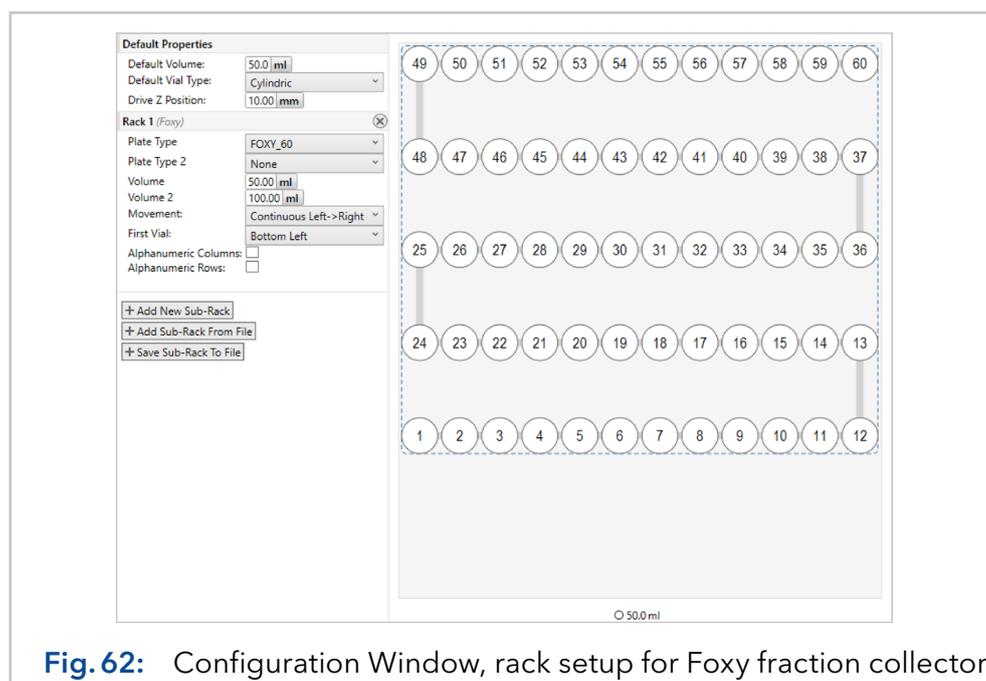
**Fig. 61:** Configuration Window, asymmetric rack setup

Asymmetric sub-racks can only be created by converting a symmetric sub-rack. Then only vials can be removed but not added. The following setting options are available for each fractionation tube:

<b>Deviant Volume</b>	Maximum vial volume.
<b>Deviant Type</b>	The shape of the vial in the rack visualisation. You can choose between round and rectangular.
<b>Relative X</b>	X Position in the sub-rack
<b>Relative Y</b>	Y Position in the sub-rack
<b>Vials Radius</b>	Size of the vial in the visualisation
<b>Bottom</b>	Z position of the mouthpiece

Additional sub-racks can be added to the rack via the option **Add New Sub-Rack**.

The teaching of fraction racks is supported by PurityChrom 6 as long as the fractionation device offers to teach. Because Foxy fraction collectors do not support the teaching option, preconfigured racks are installed with PurityChrom 6 and are available via **Plate Type**.

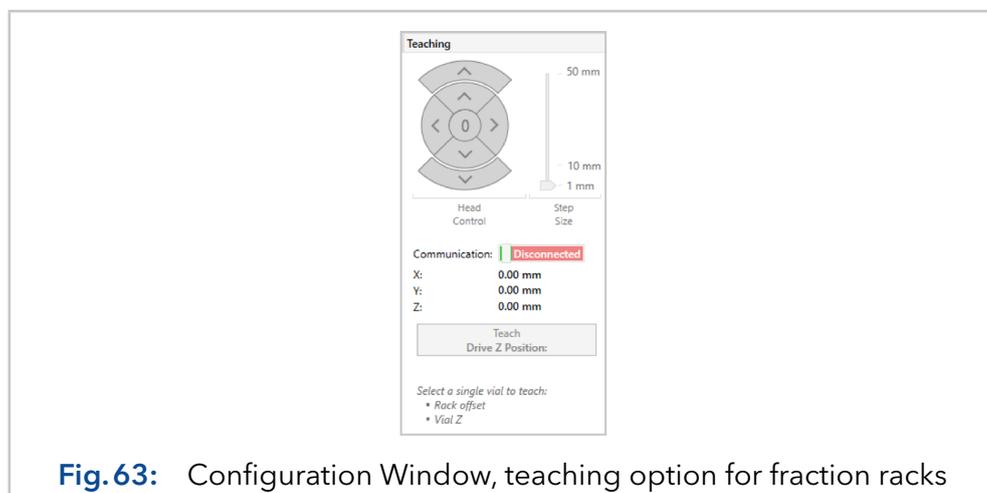


**Fig. 62:** Configuration Window, rack setup for Foxy fraction collector

Mention, that even in the case of a fraction collection via multi-position valve, a fractionation rack should be defined to monitor the filling levels of each collected fraction and to ensure the security settings of the collection unit. Therefore, a **Collector Valve Proxy** is needed as **Module** in the configuration.

## 6.12.1 Teaching of fraction racks

Fraction collectors that support individual rack geometries often have to be taught in order to achieve a match between the rack visualisation and the rack in the fraction collector. For this purpose, the teaching module is located on the right side of each rack.



To teach a rack, a connection to the fraction collector has to be established by activating the communication slide switch first. A marker  with the current position of the fraction collector now appears in the rack visualisation. Using the arrow keys of the teaching module, the marker and thus, also the fraction collector can be moved in X and Y directions. The length of the path per click can be adjusted with the slider of the teaching module.

To teach the rack position, the fraction collector is moved over a position in the rack and then the position moved to is marked in the rack visualisation by left-clicking. By clicking on **Teach**, the marker  is moved to the center of the fraction and the rack offsets are adjusted accordingly. If the fraction collector has a Z axis, this can be controlled using the outer arrow keys and also taught using the **Teach Drive Z Position** button. By marking any position and then pressing the **Go to** button, rack positions can be approached directly and the correctness of the rack with the approached positions of the fraction collector can be checked.

After completion of the teach-in process, the connection to the fraction collector should be disconnected again.

Finally, the configuration has to be saved to permanently adopt the taught-in values.

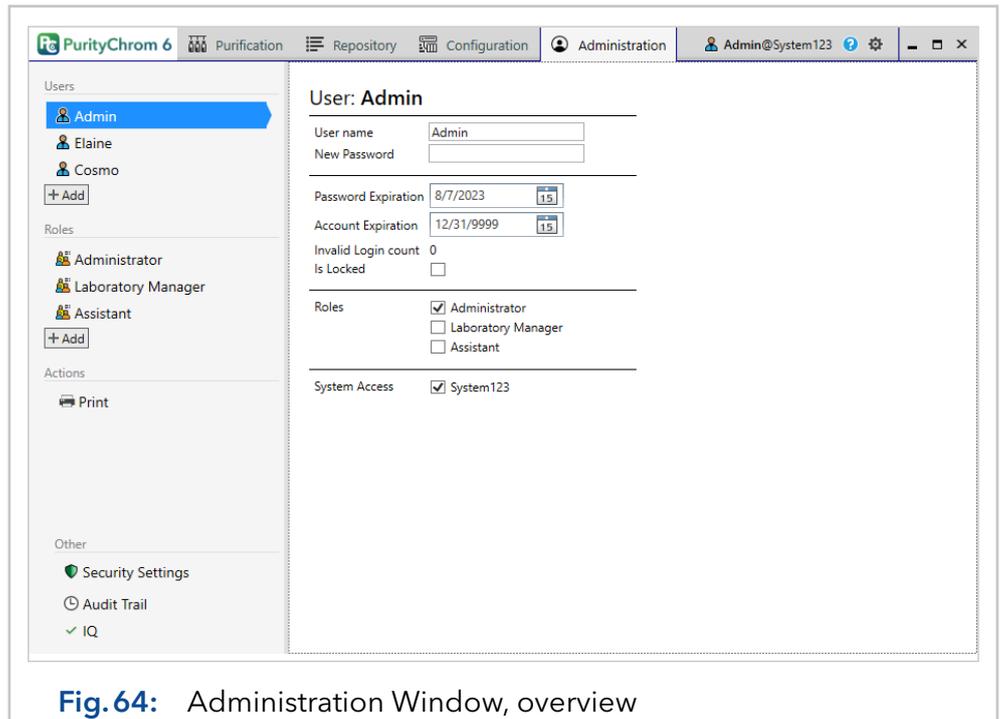


**Note:** Rack teaching is not supported by the fraction collector Foxy R1/R2.

## 7. Application part: Administration

### 7.1 Introduction to the editor

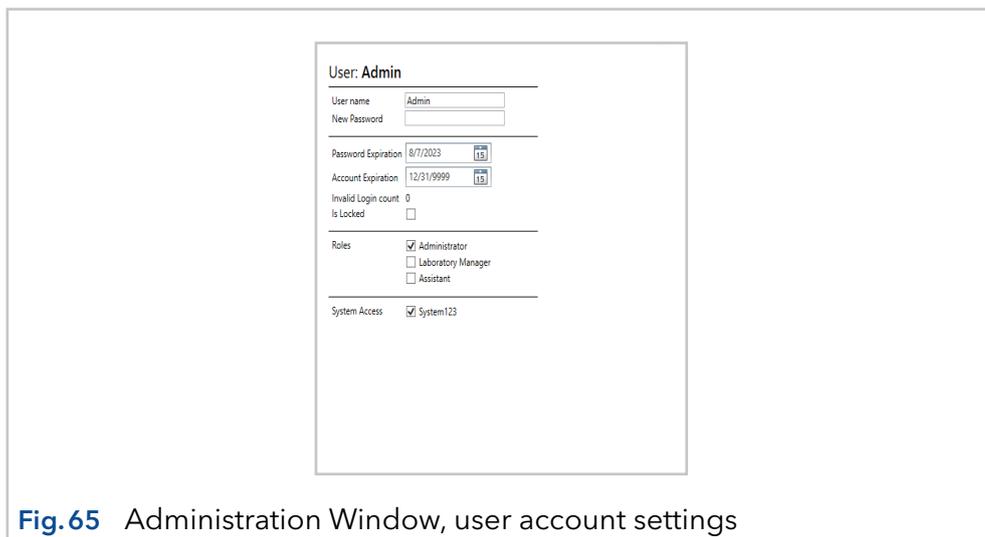
The **Administration** application part is divided into a list of users and roles on the left side and the general security settings, the audit trail and the Installation Qualification (IQ) at the bottom of the window. Furthermore, an overview of the user accounts and their privileges can be printed via the **Print** button on the left side. On the right-hand side are the details of the administration sub-item selected on the left-hand side.



**Fig. 64:** Administration Window, overview

### 7.2 User properties

The user name and password can be changed in the user properties. Under **Password Expiration** the expiration date of the current password is displayed. The date is automatically calculated based on the **General Security Settings** after each password change. However, it can also be set here manually for the current password.

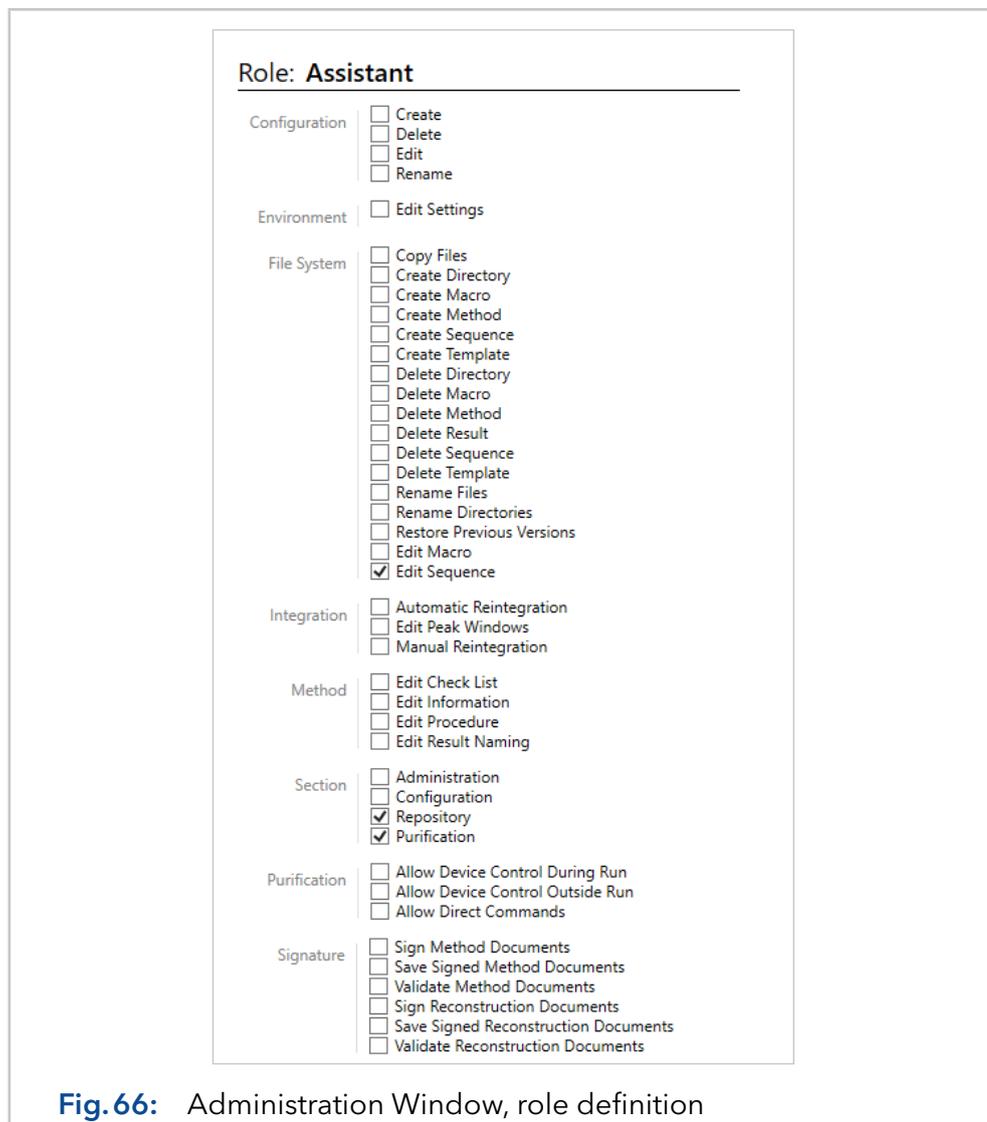


**Fig.65** Administration Window, user account settings

Under **Account Expiration**, an expiration date can be set for the user account. After this date, the user can no longer log in. If an account expiration date is set to 12/31/9999, the account will not expire. The **Invalid Login count** shows the user's invalid login attempts (wrong password) since the last successful login. The option **Is Locked** indicates whether the user account is currently prohibited to login. By removing the checkmark, the user account can be unlocked again. Under **Roles**, the user can be assigned to one or more roles. Under **System Access**, the authorization to login to a specific system can be given to the individual user.

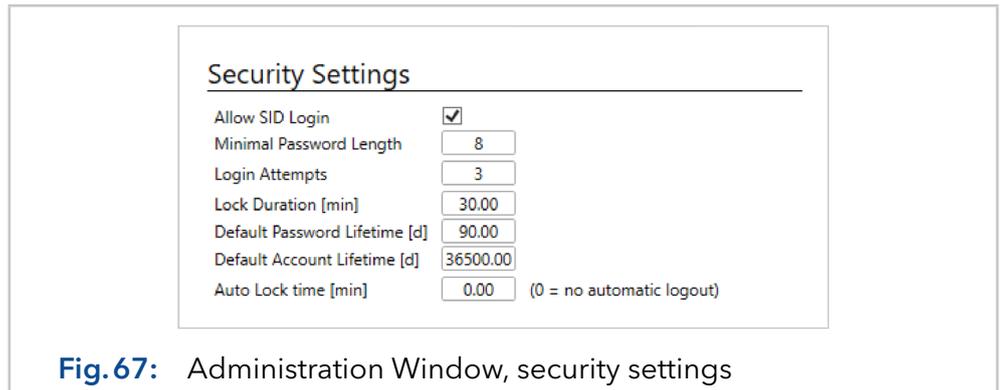
## 7.3 Role-properties

The properties of a role regulate access to the individual application parts, as well as the assignment of individual permissions. If a user belongs to several roles, the permissions of all roles are OR-linked.



**Fig. 66:** Administration Window, role definition

## 7.4 Security settings



**Fig. 67:** Administration Window, security settings

Under **Security Settings** the following settings are available:

The option **Allow SID Login** enables the user to link the Microsoft Windows user account to their PurityChrom 6 user account. If this option is used, a user who is not currently logged in with his Windows account is still able to log in to PurityChrom 6 by typing his PurityChrom login data in the login fields, but cannot use the Windows SID at this moment.

The **Minimal Password Length** defines the minimum length of a password and applies to all user accounts at the next password change.

**Login Attempts** determines after how many invalid login attempts (wrong password) since the last successful login the affected user account is automatically locked. The counter with the current failed attempts can be viewed in the user properties in the **Administration Window**. **Lock Duration [min]** defines the time a user account is locked after exceeding the number of invalid login attempts defined under **Login Attempts**.

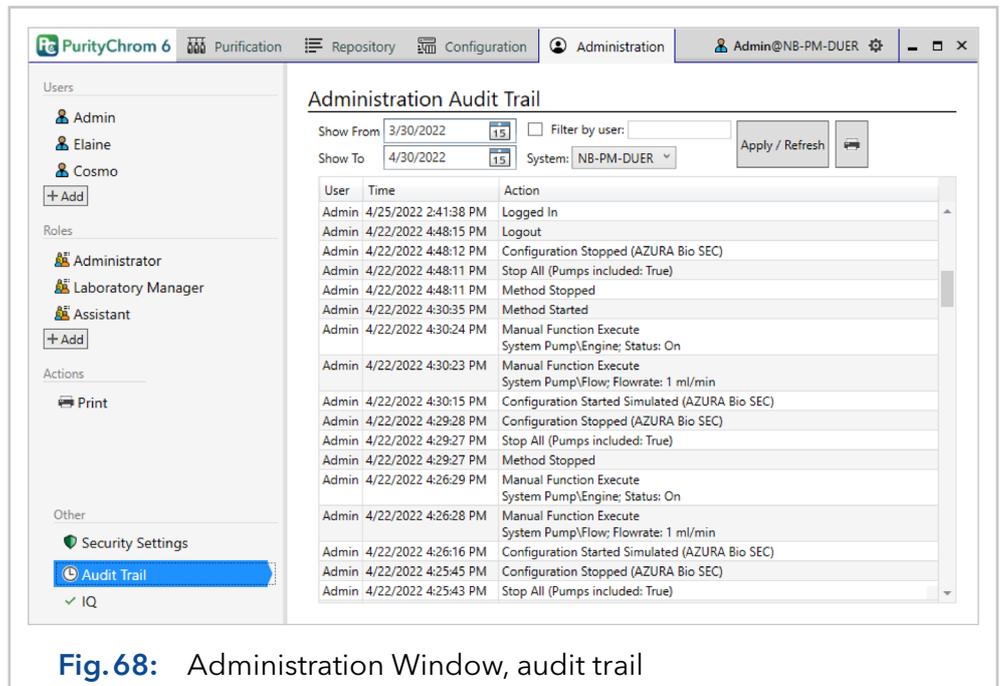
The **Default Password Lifetime [d]** specifies the interval in days at which users are prompted to change their password. This interval can be overwritten in the user settings and only applies after the next password change. The **Default Account Lifetime [d]** determines the time in days after which user accounts are automatically deactivated. This value is used when creating new users to calculate the account expiration value in the user properties.

The **Auto Lock time [min]** determines the time in minutes of user inactivity after which PurityChrom 6 is automatically locked and the user is asked for his login data again.

## 7.5 Audit trail

The **Audit Trail** summarizes all actions, the time when they were performed and the user who was logged in during this action. Accordingly, logins, logouts, the connecting to, and the manual device control of the system are logged in the audit trail. Note, that notifications that are given by the software outside of a method are not logged.

The information given in the audit trail can be filtered by date, user, and system. The chosen filter is activated by pressing the button **Apply/Refresh**. Via the printer button, the current view of the audit trail can be printed.

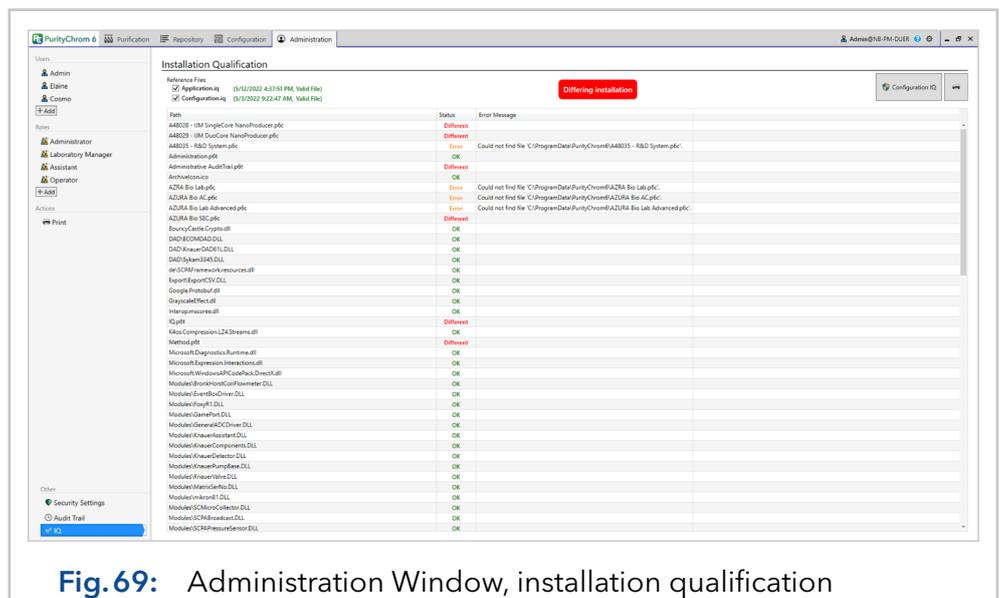


**Fig. 68:** Administration Window, audit trail

The audit trail is activated upon installation, but is only available if user administration was activated in the installation procedure.

## 7.6 Installation qualification

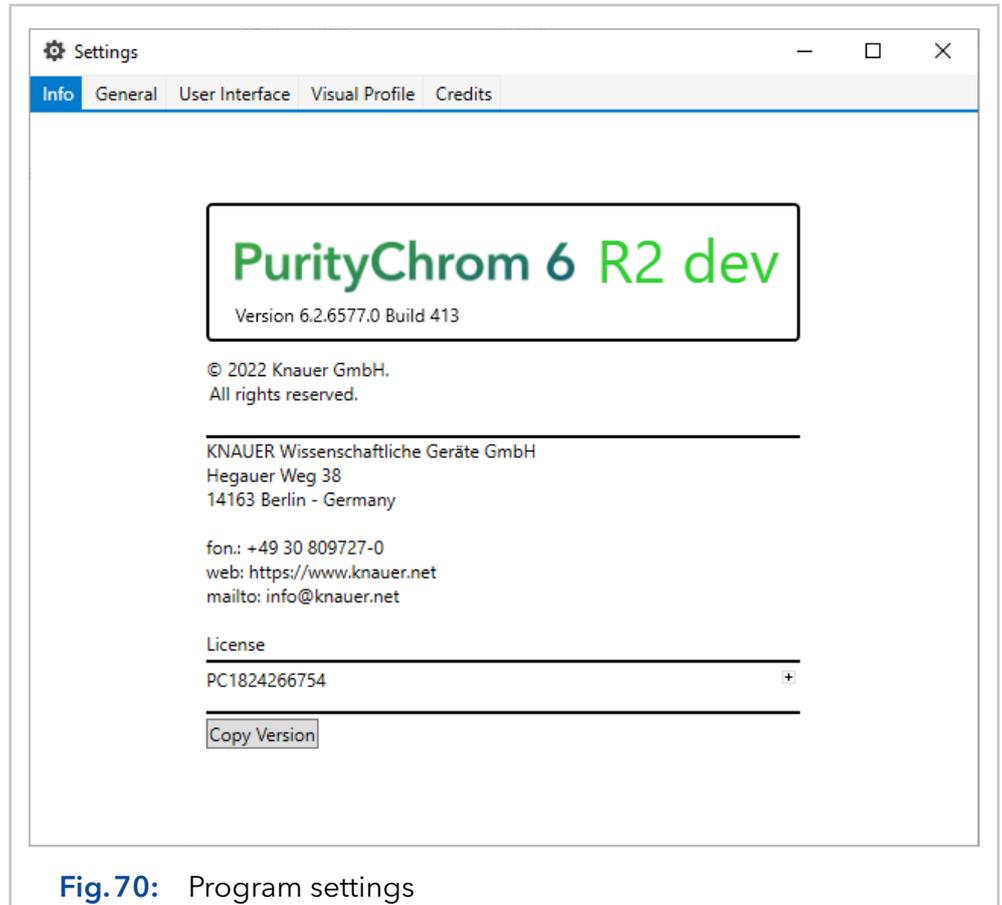
The Installation Qualification (IQ) can be checked by clicking on "IQ" at the bottom of the administration window. Choose "Application.iq" and/or "Configuration.iq" if you want to only check the software files or also the files related to your system configuration, respectively. If any software files, configuration data or their storage data paths were changed after the initial installation of PurityChrom 6 on your computer, this is indicated in the overview by entries highlighted in red and orange, as shown in the figure below. To print this overview, use the  button in the upper right corner of the window.



**Fig. 69:** Administration Window, installation qualification

## 8. Program settings

The software settings can be accessed via the gear icon  next to the user name at the top right in the title bar of the main window, if the user has the appropriate permission.



The window has several sub-chapters that are displayed on different tabs. The Info tab displays general information about the software version and the details of the license currently in use.

On the General tab you will find the following displays/setting options:

- **Working Directory:** Display of the data path where methods, sequences, result files and configurations will be saved.
- **Template Directory:** Specification of the data path in which the print templates are stored.
- **Device Simulation:** By activating this option, the device control is inactivated and the devices will be simulated. This mode is intended for training or testing.
- **Default Windows SID Login:** By activating this setting, the user is automatically logged in when the software is started, as a result of the recognition of the logged-in Windows user account.
- **Integrate Analysis-Process:** Integrate the analysis process or start it as a separate process.
- **Inter Process Communication (Ipc)-Mode:** Change method of the used processes between Transmission Control Protocol (TCP) and Memory-mapped files (MMF).

- **Display Latency:** Toggles the display of the latency in the runtime. The display consists of transport time, regularity and GUI response time in ms.

The following setting options can be found on the **User Interface** tab:

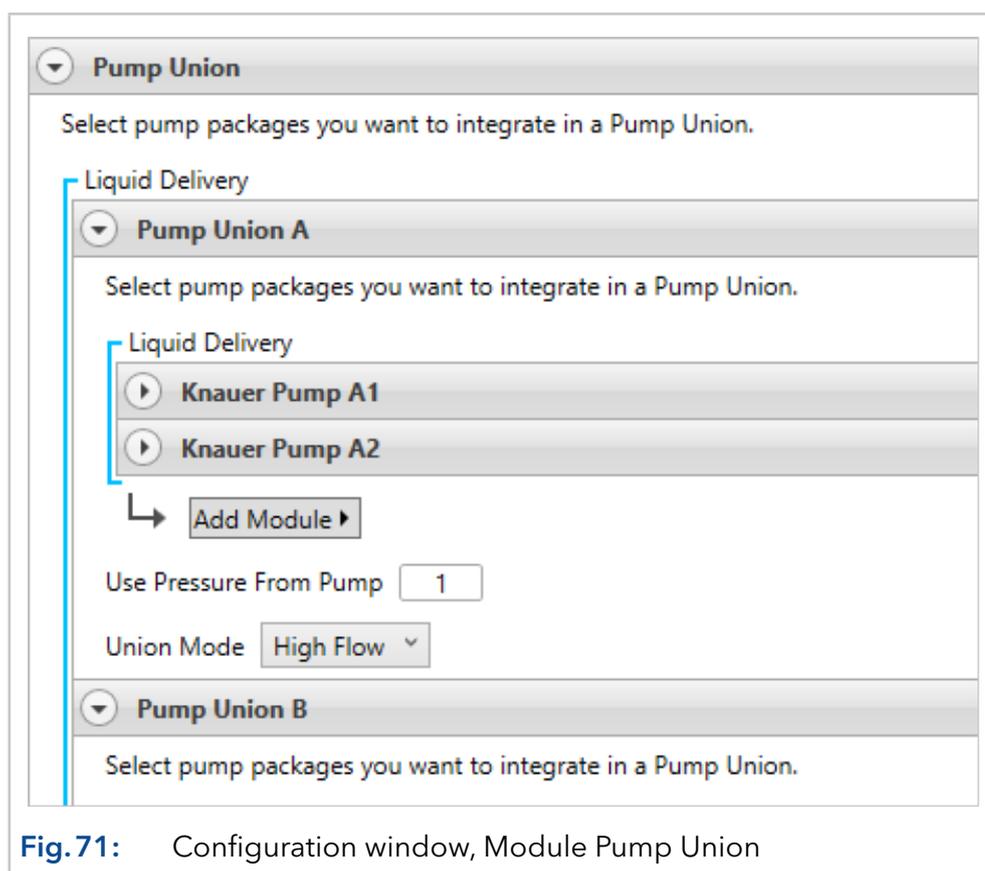
- **Drag Live Preview:** Immediate update of the function time when moved via the graphic function editor. If deactivated, the function time is updated after the completion of the movement process.
- **Curve Rendering:** The data tracks can be drawn with different methods, which differ in output quality as well as in drawing speed. The "Bitmap" mode is recommended, as it can best handle the drawing of larger amounts of data. The cleanest data traces are drawn in "Naive" mode, but this requires the most computing time. A mix of both is offered by the "Direct3d" mode. The output is very close to the "Naive" mode, with good speed - but there may be incompatibilities with the hardware.
- **Bitmap Antialiasing:** Applying an edge smoothing that is only active in "Bitmap" mode.
- **Optimized Point Merge:** With larger amounts of data, there may be several data points per pixel. By activating, only the min and max values are drawn per pixel, which significantly speeds up the drawing process of large amounts of data.
- **Baseline Ring Buffer Size:** The number of points stored to display the Baseline.
- **Acquisition Buffer Size:** If a value  $> 0$  is specified, the data track is displayed with a limited number of data points during the recording. The recommended value is the width of the data track display in pixels. To reduce the number of display data points, the two points with the smallest distance are combined at runtime.
- **Hide Baselines during acquisition:** Hides the channels that have not been started during a method step. Otherwise, channels that have not been started run in baseline mode.
- **Export/ Import Window Settings:** Export or import the purification window settings.
- **Do Print Preview:** Toggles the display of a visual preview upon pressing the print icon and selecting a print template. Change in this option is only visible after software reboot.
- **Use Sequence Table Editor:** If this option is activated, the sequence table is shown as explained in this manual. For PurityChrom 5 users, who were more accustomed to the former autosampler control file editor, a deactivation of this option enables a design of the sequence table similar to the former autosampler control file editor.

On the **Visual Profile** tab, the colors for data channels, gradients and the various annotations and representation of functions in the chromatogram window can be customized.

## 9. Special features

### 9.1 Pump Union for high flow applications

The module **Pump Union** enables the connection of multiple pumps to one pump system. Via the **Add Module** button, the modules for the individual pumps in the system can be added. The pressure signal will be later only displayed, monitored, and recorded for one of the pumps. Therefore, select the pressure channel that should be used under **Use Pressure From Pump** by inserting the number of the pump in the Pump Union module. As **Union Mode** choose either **High Flow** or **Software High Pressure Gradient** regarding the type of pump connection. The option **High Flow** defines multiple pumps as one pump in the software meaning the sent flow rate will be divided by the number of pumps and each of the pumps will flush with a portion of the total flow rate. With this option, high flow rates can be achieved. The option Software High Pressure Gradient enables the connection of pumps in a gradient system, e.g. with pump 1 flushing eluent A and pump 2 flushing eluent B.



**Fig.71:** Configuration window, Module Pump Union

Figure 72 shows an exemplary system configuration consisting of 2 pumps flushing eluent A and 2 pumps flushing eluent B. Such a system can only be configured with pumps of the same type as here in the example with 4 KNAUER Pump P 2.1L with 500 ml pump head. In such a case, pumps A1 and A2 as well as pumps B1 and B2 are connected via high flow mode. Both pump systems (A/B) are then connected via a Software High Pressure Gradient Mode to enable the programming of A/B gradients. The maximum pressure of such a system is related to the maximum pressure of one pump, whereas the maximum flow rate is related to the sum of the maximum flow rate of all A (or B) pumps.

Respectively, in the given example, the maximum flow rate would be 1000 ml/min. If the user sets the flowrate to 800 ml/min and defines a gradient of A/B = 80/20, then pump system A has to deliver 640 ml/min and thus, pump A1 will deliver 320 ml/min and pump A2 also will deliver 320 ml/min, while pump system B has to deliver 160 ml/min and thus, pump B1 will deliver with 80 ml/min and pump B2 also will deliver 80 ml/min.

## 9.2 Stacked injections

Stacked injections are an easy way to increase productivity and improve efficiency in preparative chromatography. With this approach, multiple injections are nested in one batch run for maximum throughput and minimum downtimes between peak collections. This technique saves time as well as a solvent but can only be applied to isocratic methods.

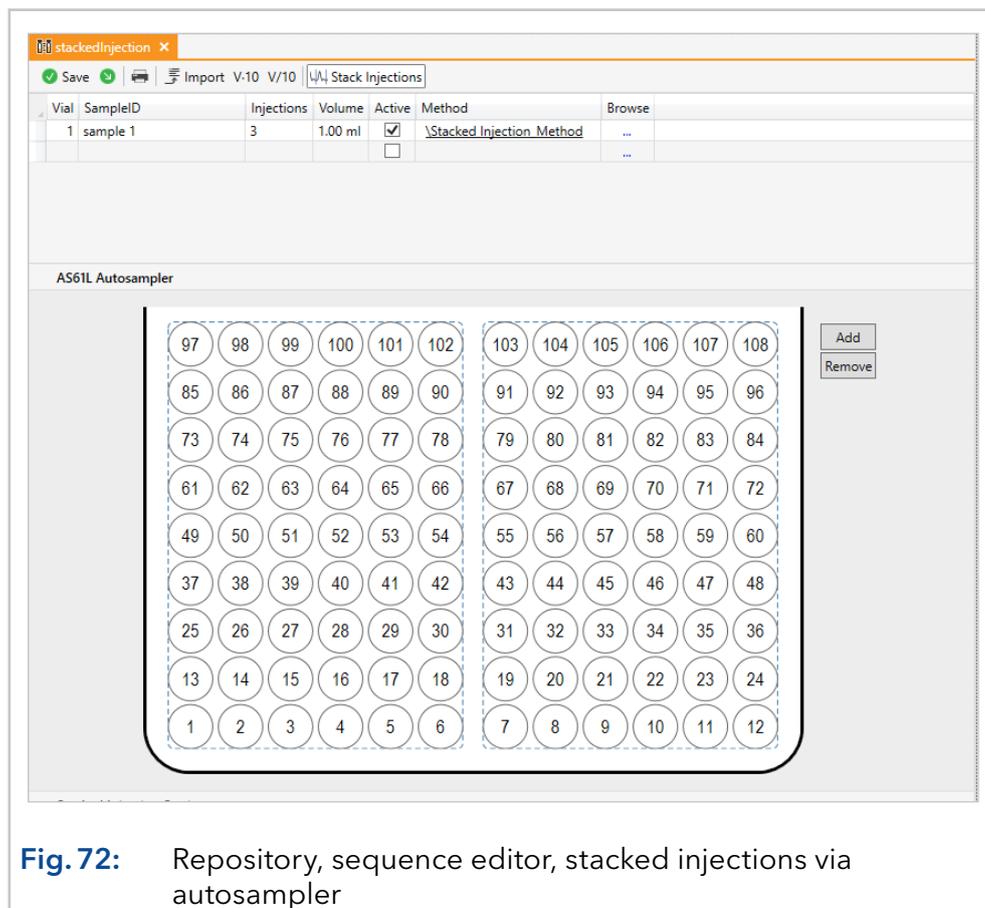
### 9.2.1 Stacked injections: General

The stacked injection settings will be displayed on the bottom of the window and include the following parameters:

Cycle Time	Time between each injection
Load Before Inject	Time for preparation of the next injection. It is advised to use a command as load step. The actions defined in the command will be executed prior to the injection depending on the defined time.
Break Into Single Cycles	If activated, the procedure is divided into single runs in the scheduler (Purification Window) and later saved in individual chromatograms. If deactivated, one chromatogram will be created for all injections.
Preparation Step	Method step that should be performed only once at the beginning of the stacked procedure including all initial parameters and chromatogram start.
Load Step	Method step that includes the sample loop loading procedure or any actions necessary before injection
Inject Step	Method step that includes the injection.
Cycle Step	Method step that should be performed during stacked injection procedure including the fractionation. This step is prolonged according to the number of injections.
Finish Step	Method step that should be performed only once after finishing the whole stacked injection procedure.

## 9.2.2 Stacked injections via autosampler

The stacked injection procedure via an autosampler can be defined in the sequence editor after activating the Stack Injections button .



**Fig. 72:** Repository, sequence editor, stacked injections via autosampler

In the upper sequence editor part all to be injected samples and volumes are defined. All samples will be injected after each other in a stacked mode in one method. In the lower part all parameters concerning the stacked injections can be defined.

## 9.2.3 Stacked injections using a sample pump

Using a sample pump samples can be injected directly onto the column or a sample loop can be filled before injecting it through the loop.



**Note:** The volume setting in the sequence editor does not influence the volume injected via the sample pump. The volume is defined by the method. It is advised that the load step is used for defining the injection time and thereby the injected volume.

## 10. How to

### 10.1 Recommendations for backups

It is recommended to regularly back up the PurityChrom 6 working directory, the template directory, the user database and the data directories of the configurations. Any backup program or Windows Explorer can be used for this purpose. However, it has to be ensured that PurityChrom 6 is not started during the backup. By default, the working directory, data directory and the template directory are located in C:\ProgramData\PurityChrom6, while user database and general software settings are stored in C:\ProgramFiles\PurityChrom6.

The data from the **Repository** (the data is stored in the data directory of the configuration) can also be backed up during the runtime of PurityChrom 6 using the archives option.

If a MySQL server is used for user administration, the contents of the database should also be backed up regularly. The procedure for this can be found in the documentation of the database server.

The backup interval should be based on the frequency of use of PurityChrom 6.

# 11. Appendix: Concepts, structure & ways of working

The concepts used by PurityChrom 6 are shown in the following. The exact technical understanding of individual concepts is important for certain questions, such as the installation concept for administrators. Furthermore, the exact concept description can also be used for concept validation for potential users, as well as for technically interested users in general.

## 11.1 Installation concept and user administration

The PurityChrom 6 installation directory (by default C:\Program Files\PurityChrom 6) contains the software files, the license and a file with the basic software settings (ApplicationSettings.dat). If PurityChrom 6 is installed in the standard Windows directory for applications (C:\Program Files), the executing user needs administrator privileges to be able to write to this directory. All files in the application directory are only written during the installation or configuration process. During regular application execution, only read permissions are required.

The encrypted writing of the basic software settings is done via the PrepConConfiguration.exe application, which is executed after the installation procedure has been completed. However, the configuration can also be changed after installation by starting PrepConConfiguration.exe. To do so, the user needs write access to the application directory. The basic software settings include:

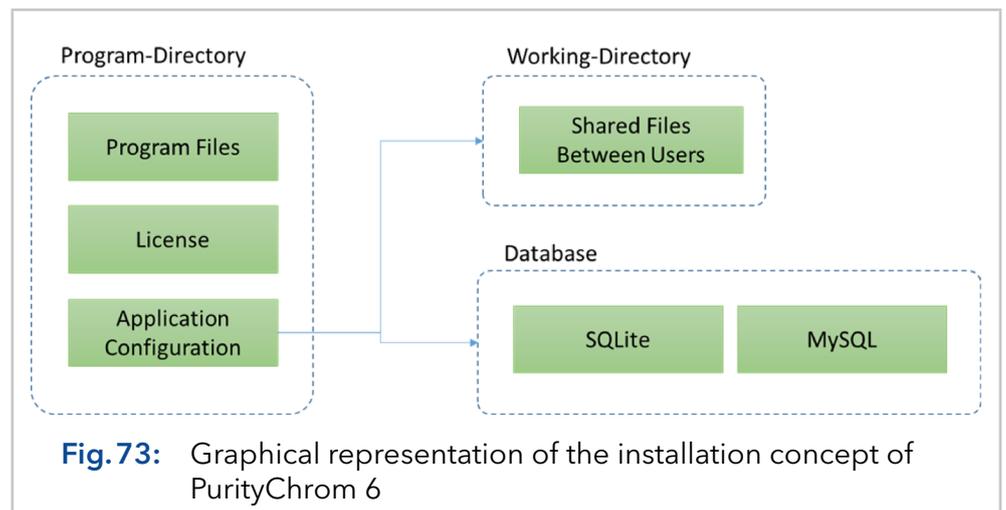
- License selection.
- Selection of the working directory.
- Selection of the authentication mode (only with the user name, local database, global database).
- Selection of the file system (local file system, via PurityChrom 6 file service).

PurityChrom 6 has its own user administration which, depending on the software setting, accesses a local SQLite database or a MySQL server. The login of a user in PurityChrom 6 is initially independent of the current Windows user, i.e. a specific PurityChrom 6 user can log in from the context of different Windows users. The basic requirement here is, that the same PurityChrom 6 installation and the working directory is accessible to every Windows user. For this reason, the working directory should not be located in a user folder (e.g. User\Documents). In an environment where each Windows user is assigned to a PurityChrom 6 user, the SID of the Windows user can be linked to an associated PurityChrom 6-User. In this case, the PurityChrom 6-User optionally no longer needs a password for authentication. At this point, it should be noted, that PurityChrom 6 is always executed under the context of a user from the point of view of the operating system. This means, that when a user changes in PurityChrom 6, the operating Windows user does not change.

With the PurityChrom 6 user administration, certain functions can be activated or deactivated for individual users, including the deletion of files via the PurityChrom 6 interface. Now there is the case that some PurityChrom 6 users are denied the right to delete data.

This can collide with the permissions of the Windows user, who may be able to delete files. In this case, a PurityChrom 6 user could not delete a file from the PurityChrom 6 interface, but could from the Windows-Explorer. The same problem occurs with write permissions. In a GMP-compliant environment, for example, the user has to ensure that files cannot be deleted without an audit trail entry. If a file is deleted in the Windows Explorer, it is not GMP-compliant. The solution to this problem lies with the user and can be solved, for example, by making backup copies.

At this point, however, it should be mentioned that in PurityChrom 6 all file accesses are abstracted so that they can also be redirected through a service. The redirection of file accesses through a service can be configured in the software settings. The file service can thus run on the system on which PurityChrom 6 is also running, only with higher permissions than the Windows user. For example, the service could have delete permissions for a directory, but the Windows user himself does not. Depending on the privileges, a user could still obtain write access by the service via PurityChrom 6.



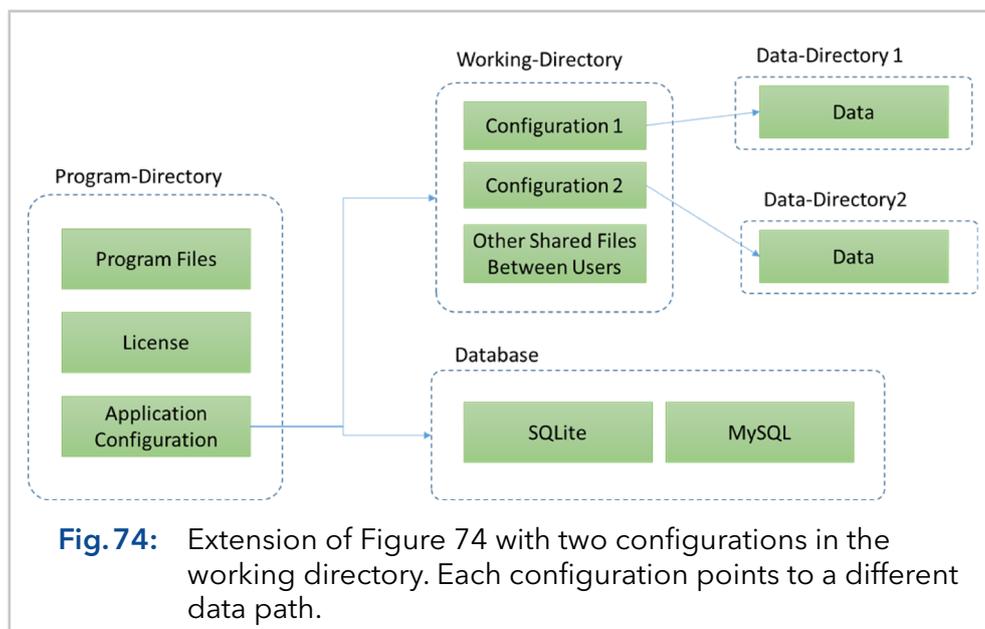
**Fig. 73:** Graphical representation of the installation concept of PurityChrom 6

The application directory (by default C:\Program Files\ PurityChrom 6) is described by the installation routine and contains the software files, the license and the basic software settings (ApplicationSettings.dat). Furthermore, the software settings contain references to the working directory and the user administration. Data files to which all PurityChrom 6 users have access are stored in the working directory (by default C:\Program-Data\ PurityChrom 6).

## 11.2 System-related data: Configuration

PurityChrom 6 is used for system control. Before a system can be controlled, it has to be configured. The system configuration (abbreviated to "configuration") contains, among other things, a list of the devices/modules to be controlled and a reference to a data directory. All data related to a system are stored in this data directory. This primarily includes methods, sequences and result files.

Fig. 74 on page 79 shows an example of two configurations in the working directory, which in turn points to their own data directory. To ensure that all users have access to the data directory, it should not point to a user directory.

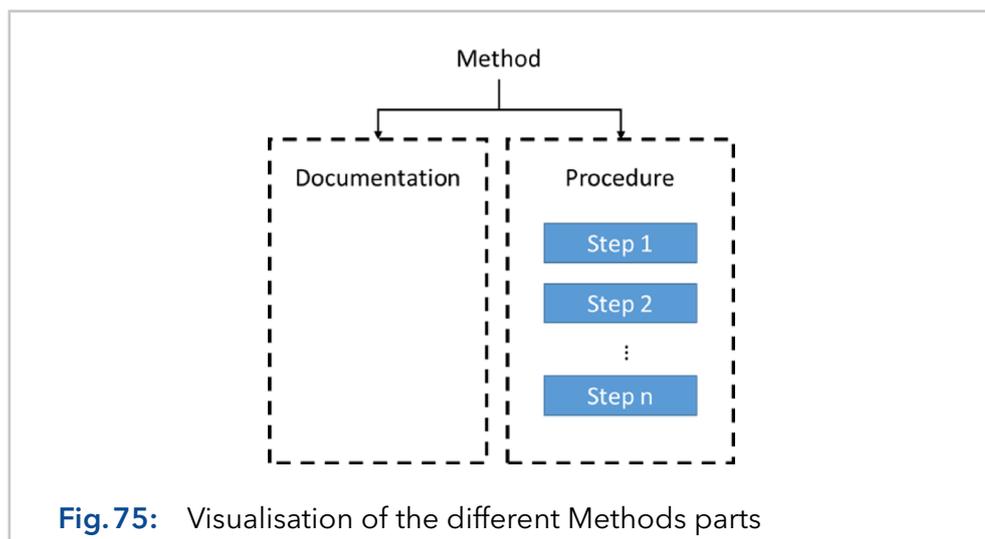


### 11.3 Working with sequences, methods and result files

A system is usually controlled with the intention of generating a product. How exactly the system has to be controlled to obtain a single product is described in a method. The execution of a method delivers a result file in relation to the data, in which the execution is documented. If several methods are to be executed consecutively, they can be combined into a sequence.

In addition to the documentation values (such as author, used solvents, file name and check list), a method itself contains a procedure part in which the exact function calls are listed (Fig. 75 on page 80). The procedure can be seen as a sequence of steps. The next step is processed when the current step is finished. The processing of the entire method ends when the last step of the procedure has been processed. In PurityChrom 6, three step types are defined: The command step, the time-table step and the macro step.

The command step is defined in such a way that a list of functions is executed once when the step is started and then waits for a condition to be fulfilled. The time table step is defined by a time-based function execution. The term time here is based on the physical definition and means a sequence of events with a unique and irreversible direction. Both process time and volume can be used as a time for a time table step. Macro steps represent sub-procedures. Recurring command or time table steps can be stored in macros and used in several methods.



**Fig. 75:** Visualisation of the different Methods parts

The method itself is based on the definitions of the configuration. Accordingly, the method depends on the configuration and can initially only be executed in the context of the configuration being used. If a method is also to be executable under another configuration, the functions have to be mapped onto the modules of the new configuration first.

## 11.4 The runtime environment

The Purification program part is the runtime environment of PurityChrom 6, which actively controls the devices and processes sequences and methods. Purification is initially inactive during a regular PurityChrom 6 start. To activate the purification, the configuration to be started has to be selected beforehand. The purification takes care of the following functions:

- Communication with the modules/devices.
- Display of the system visualisation.
- Loading and processing of methods and sequences.
- Control of the devices by the user through menus and visualisation.

The runtime environment is executed separately from the user interface in a separate subprocess. Communication takes place using memory-mapped files or TCP/IP. In the case of a crash of the graphical user interface, the runtime environment can continue to be executed in this way. This purification concept also opens up the possibilities for using PurityChrom 6 without a graphical user interface (e.g. PurityChrom 6 as a component in a process).

## 11.5 Driver concept

PurityChrom 6 works with an abstract concept for devices/modules. Every module that is integrated has to implement an abstract basic interface. Independent of the device type, a module has to provide the following properties:

- Executable functions.
- Programmable functions for the procedure with a graphical interface.

- Status values.
- Configuration interface.
- Data tracks.

This abstract definition also allows devices to be integrated into PurityChrom 6 whose type is still unknown to PurityChrom 6. Based on the basic types, e.g. pump, detector, etc., further types have been defined, e.g. a gradient pump. PurityChrom 6 also uses the definition of specified types for special functions. If PurityChrom 6 recognizes that it is e.g. a pump driver, this type of knowledge is used to utilize e.g. the current flow rate for level calculations.

By using the basic type, the functionality used by PurityChrom 6 is limited to the execution of functions from the procedure and from the visualisation, as well as to the generic evaluation of the status information (e.g. in status queries). Visualisation elements for modules as well as special menu entries can also be integrated into PurityChrom 6 by libraries.

## 11.6 Print Templates

The printout is designed using print templates. Every possible file that can be printed via a print template is listed as a data source in the template editor. Each data source of a template provides basic values for the template, such as texts, tables and data tracks. The provided values can then be arranged in the template.

If a template is used whose data source is not available, the corresponding elements are not printed. Even though several data sources can be used in one template, the standard application is that a separate template is created for each file type, which accordingly only refers to the respective data source.

## 11.7 Fractionation concept

A standard feature in preparative chromatography is the collection of fractions. For this, a currently conveyed volume has to be classified as to whether it is a value fraction or waste. This classification can be done manually in PurityChrom 6, by thresholds or by a data analysis. Physically, the fraction volume has to be separated from the waste volume, which can be done, for example, by switching a valve. If different fractions are produced in a process, these have to be further separated spatially, where this task is performed by a fraction collector, for example.

Abstracting this fractionation process, one can make the following assumptions regarding the system:

- There is a conveyed volume by a pump.
- A condition has to be defined where the system is in a fractionation state.
- Functions have to be performed to direct the pumped volume to the waste.
- Functions have to be executed to direct the pumped volume into the fraction.

- A function has to be executed to switch to the next fraction.
- The system has to be informed of the current position of the fractionating vial.

All the assumptions listed previously are combined in PurityChrom 6 in a module, the **Collection-Unit**. Different collection units can be defined for a system to be controlled. The advantage of this concept is that the system functionality of fractionation is always presented to the user on the same interface (collection unit), regardless of the device types used.

If the collection unit is to collect the conveyed volume, the function **Set Fraction** can simply be programmed. The collection unit will then execute the configured functions and monitor the delivered volume of the referenced pump. When the fraction limit is exceeded, the collection unit will execute the command for the next fraction. Similarly, to end the fractionation process, the **Set Waste** function is called.

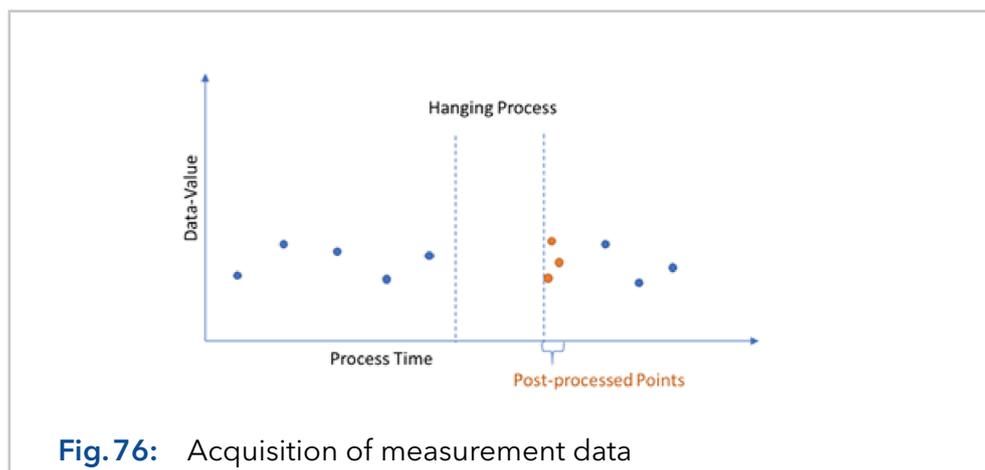
This concept also allows more complex system structures to be described. If, for example, several valve switches are required to direct the pumped volume into the fraction, all these switching commands can be specified accordingly in the collection unit configuration.

## 11.8 Acquisition of measurement data

In the configuration, logical channels are defined, which are based on the hardware channels from the device drivers. In short, a hardware channel can be described in such a way that it transfers a finished measured data portion (slice) to the superior system by triggering an event. In this context, a slice is a part of a measured signal. A module can determine for itself how many measured values a slice contains. For example, an AD converter channel returns a single floating point number as a slice, a diode array detector a whole spectrum. A logical channel can refer to several hardware channels and calculate them with each other.

PurityChrom 6 processes time-related measurement data, i.e. a slice also has to have a time component - in this way, non-equidistant slices can also be described. However, the time component of a slice does not have to be returned by the hardware channel itself, because PurityChrom 6 "stamps" each incoming slice with the current process time. This stamping process is done by appending a time component to the finished slice of the hardware channel. This attached time can now be used to display the data and do calculations. The stamped process time is in the end the time with which PurityChrom 6 has "noticed" one or more slices.

It is important to understand what this process timestamp means and how to Interpret this time component. Assuming that the analysis process is currently busy, e.g. because the computer is hanging or a module is blocking the process, it may be that slice processing is also hanging and thus many slice events have to be post-processed. In this case, a similar process time would be stamped on the delayed processed slices.

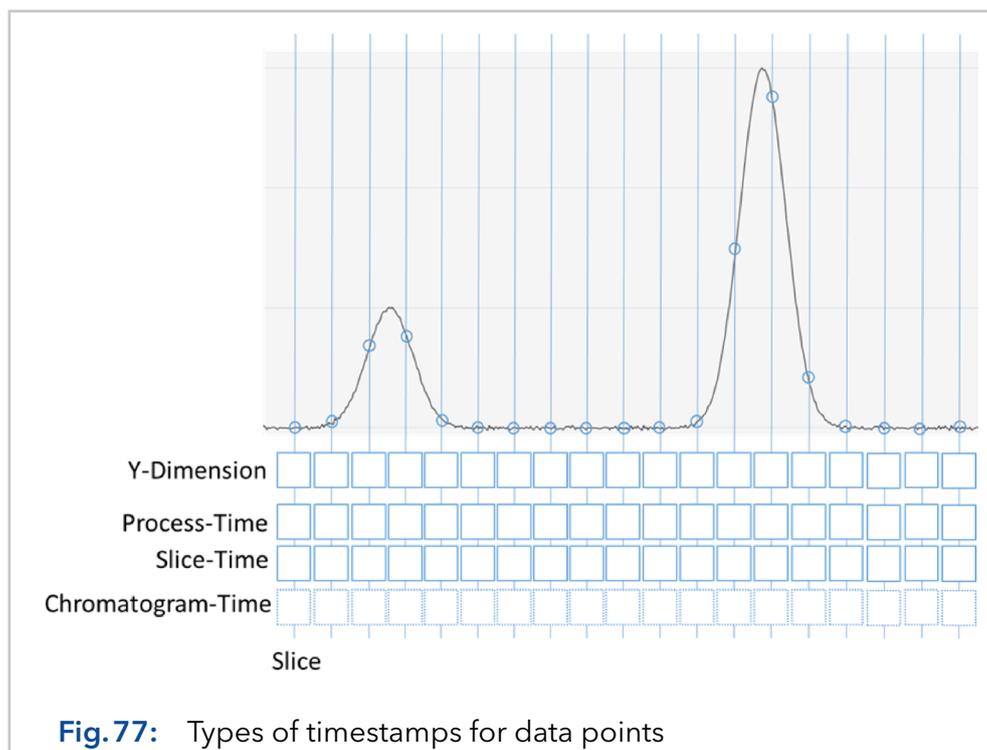


The process time indicates when a slice was handled in the process. Even if you have a hardware channel that delivers equidistant data due to the measurement process, it will never be possible to stamp it equidistantly because there is always a small delay in processing (or sometimes a larger one due to a hang-up). If the data points are processed on time, which should be the case in practice, then the process time distance between two nearby data points corresponds approximately to the data point distance with which the data were actually measured.

The process time value is not suitable for data interpretations that are strongly dependent on the processing time. If, for example, the area of a peak in the measurement data of the AD converter is to be calculated, the process time of the slices would not be used for this question. If the process time were nevertheless used as a time component for this calculation, an error would be artificially implanted.

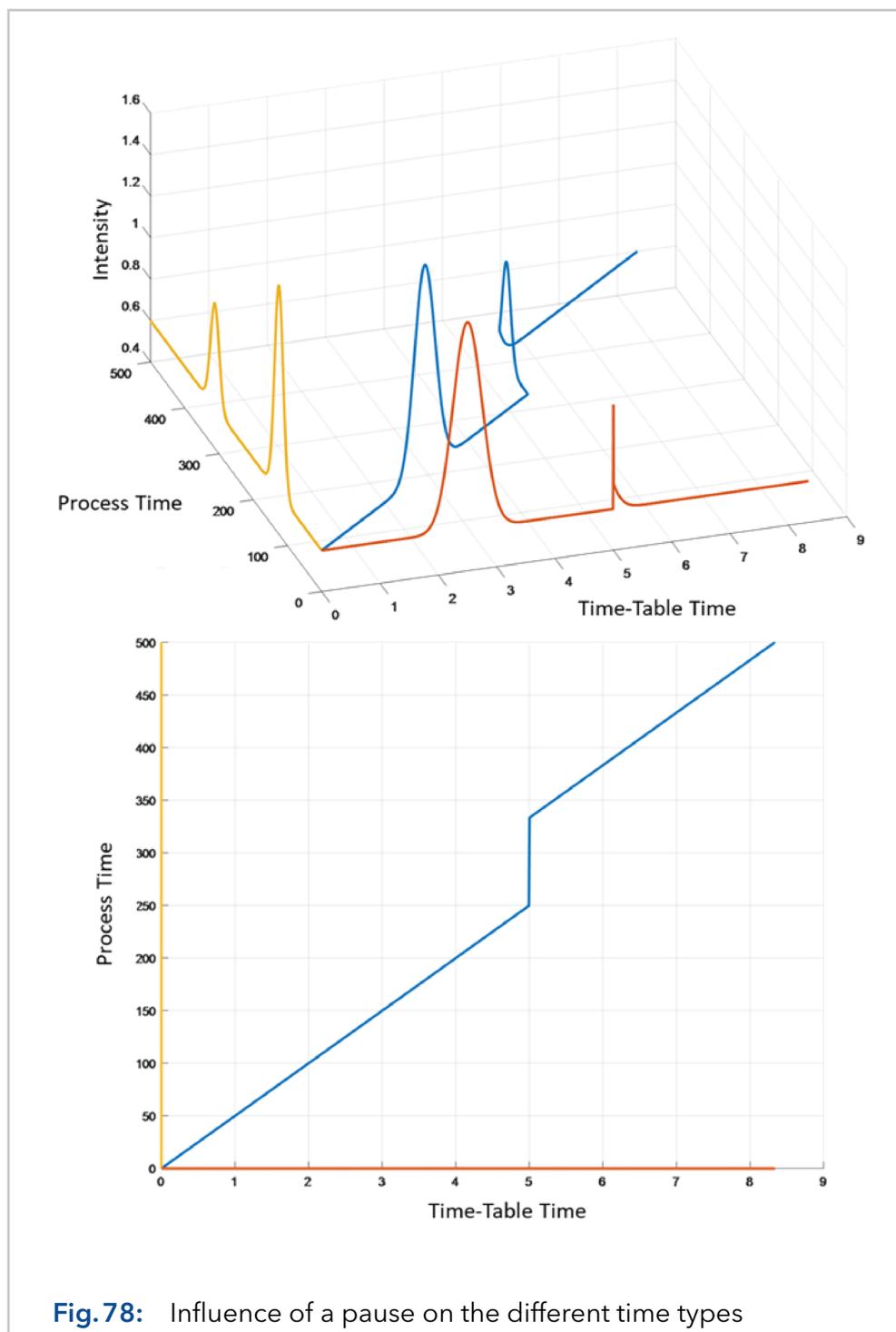
Each data point is thus stamped in PurityChrom 6 with the process time and with the time of the "real measured data point distance". If a chromatogram is recorded within a time control file, the chromatogram slices are further stamped with the time table time for the third time. The reason for this third stamping is that the time of the time-table file is assumed by PurityChrom 6 to be 'Non-Predictable'. A volume can also be used for the time base of a time control file. This volume could be, for example, the calculated pumped volume of a pump or a volume measured using a flow sensor.

Fig.77 on page 84 visualises a chromatogram consisting of several slices. Each slice contains the measured value (Y-dimension, blue circle on the chromatogram), as well as the three attached time dimensions. The Y dimension and one of the three time dimensions are used for the display or for calculations.



The nice thing about storing the time table time for each slice is that if the time difference for nearby points is different from zero, i.e. the time table "time" is no longer linear to the process time in spots, you can reconstruct this again. An example of a use case is the pause function, which stops the method processing. Stopping the time table means that all slices are stamped with the same time table time during the pause, so the chromatogram stops. However, no data is lost, because the chromatogram can be viewed with the process time, as this continues to run even during a time table pause. In this way, it is possible to see what happened during the pause in relation to the data track.

Fig. 78 on page 85 shows a chromatogram that was recorded in a time table that was paused at 5 time table time units for a certain duration. The blue chromatogram was plotted over two time components (process time and time table time), the orange one over the time table time only and the yellow one over the process time. In the top view, it is easy to see how several data points have the same time table time, while the process time continues to increase.



## 11.9 Evaluating measurement data / peak detection / integration

Integration in PurityChrom 6 describes the evaluation of a data track to calculate the limits of peaks. Once the peak boundaries have been calculated, further peak-specific values such as the area or symmetry values are calculated.

At runtime, the online integration is used to calculate the current peak situation (start, end, valley...) of a channel during the analysis. Based on the calculated peak situation, actions can be programmed in the procedure. Usually, the peak situation is used as a trigger for fractionation functions.

If the operator wants to collect a separated substance, the time window in which the substance (and thus the peak in the data) is expected is required as prior knowledge. This time window can be programmed in the procedure editor as a function of a time table (Sampling Window). Within the sampling window, further functions can be programmed for a new peak situation, e.g. peak start.

In this context, peak means a local increase in the measurement data. For fractionation, the beginning and the end of the peak are of particular interest. Basically, there are various detection options for the peak limits. For example, a threshold value can be defined, above which a peak start is assumed. The disadvantage of the simple threshold function, however, is that it is dependent on the current signal background. Thus, for robust use, it has to be ensured that the background is always at a similar level.

Another method of peak detection, which is also used as the basis of the integration function, is the definition of a threshold value in the first time derivative of the measurement data. The first time derivative describes the change of the signal per time unit. Thus, large changes in the measurement signal lead to large deflections in the first derivative. A major advantage of using the derivative is that the threshold value defined in this way is independent of the signal background. For example, a signal increase from 100 mV to 200 mV per second leads to a derivative of 200 mV/s - 100 mV/s = 100 mV/s, but a signal increase from 1000 mV to 1100 mV per second also leads to 1100 mV/s - 1000 mV/s = 100 mV/s.

Since the first derivative amplifies the signal noise, it is common to calculate it over several points to achieve smoothing. Specifically, the first derivative is calculated via a convolution operation using a derivative convolution kernel of length  $N$  ( $N \in \{5, 7, 9, 11, 13, 15\}$ ). I.e. to calculate a point in the derivative,  $N$  data points are needed. This has the consequence that at runtime the derivative follows the signal by  $\lfloor n/2 \rfloor + 1$  points. For a peak start detection, four data points are needed out of the first derivative, which calculates the delay at  $n=5$  to  $\lfloor 5/2 \rfloor + 1 + 4 = 7$  points. With a data point interval of 500 ms, this would be a delay of 3.5 s.

In relation to the system, there is a volume from the detection point to the exit point of the liquid. Collection unit functions can be delayed by this system-specific dead volume. However, since the delay of the peak detection is known, this delay time is also taken into account.

In PurityChrom 6, the same algorithm is used for online peak detection and for recalculation (offline) to detect peak situations.

## 11.10 Documentation / Audit-Trail

PurityChrom 6 documents relevant user actions in the general audit trail associated with user management. Relevant user actions include logging in and out, manual device control, and starting and stopping methods. In addition to the general audit trail, each result file contains its audit trail in which all relevant actions during a run and post-processing are documented.

Since the used method has a direct influence on the result file, it is additionally documented in the result file. A version history is stored in the method itself, whereby previous versions can also be restored. The version history is not documented in the result file, only the used version.

Science with Passion



Latest KNAUER instructions online:  
[www.knauer.net/library](http://www.knauer.net/library)

**KNAUER**  
Wissenschaftliche Geräte GmbH  
Hegauer Weg 38  
14163 Berlin

Phone: +49 30 809727-0  
Fax: +49 30 8015010  
E-mail: [info@knauer.net](mailto:info@knauer.net)  
Internet: [www.knauer.net](http://www.knauer.net)