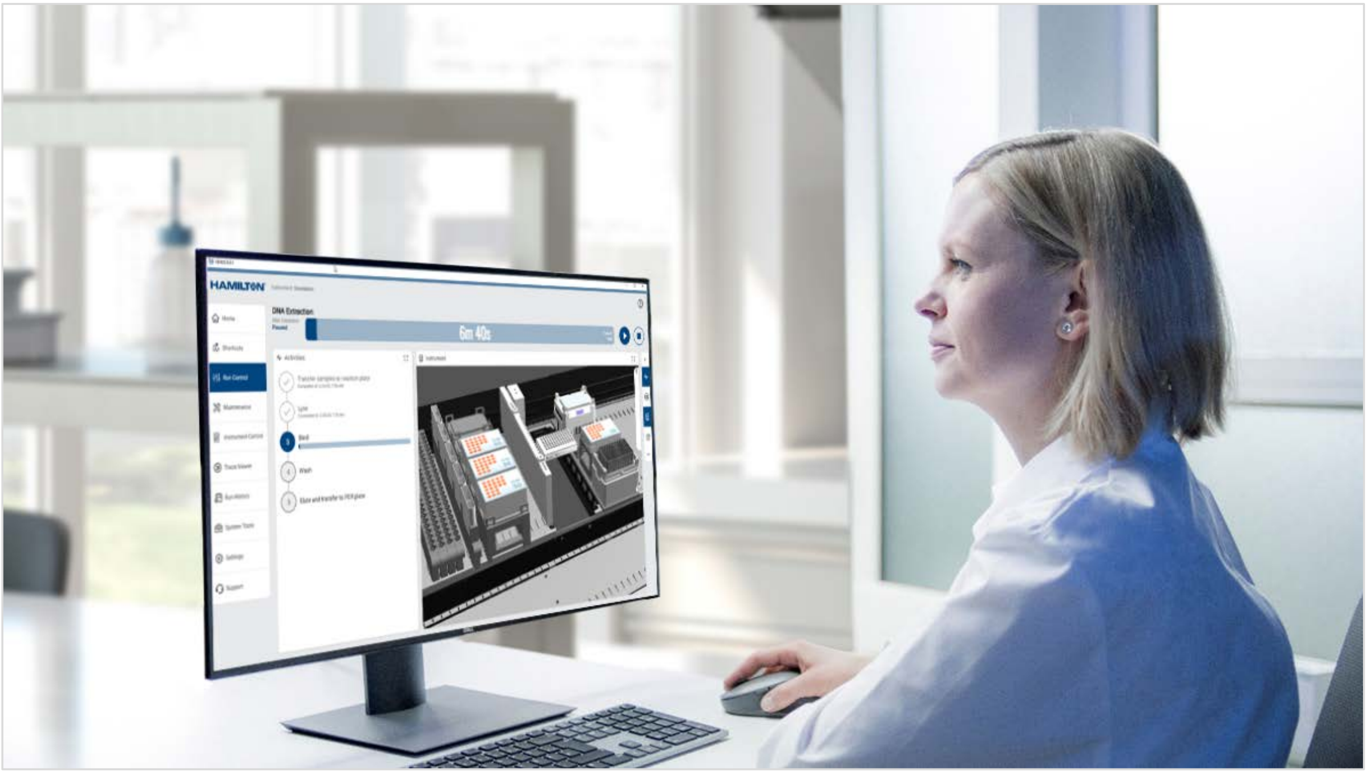
VENUS Software Programmer's Manual



Product Number 111008367_00

Important notice

Table of Contents

PART I: General and Installation Information

1 General Information

This Programmer’s Manual is designed to help in getting the most out of the Microlab STAR, Microlab STAR^{V}, and Microlab VANTAGE.

Please read through the corresponding Microlab instrument Operator’s Manual before beginning to operate the instrument. This manual should be read with particular attention, it contains important information about the instrument.

1.1 About this Manual

This manual refers to the Hamilton VENUS Software release base package, in the following summarized as VENUS Software. Hamilton’s Microlab STAR, Microlab STARlet, and Microlab STARPlus are in the following summarized as Microlab STAR; Hamilton’s Microlab STAR^{V} and Microlab VANTAGE are in the following summarized as Microlab VANTAGE.

This manual is a programmer’s guide and therefore is intended for advanced users only. The manual describes mainly VENUS Software in depth, enabling a programmer to program the instrument. This manual is divided into three parts.

Part I: "**General and Installation Information**", provides an introduction to VENUS Software.

Part II: "**The VENUS Software**", introduces VENUS Software through the editors.

Part III: "**Working with VENUS Software**", offers a guide to programming of the Microlab STAR/VANTAGE pipetting tasks, accessories and third-party equipment.

Warnings and Notes are included in this manual to emphasize important and critical instructions.

ATTENTION

Any warning or important information will be accompanied by this symbol. Read these items carefully.

NOTE

Information which can be useful but not essential to the task at hand is given.

[...] Push buttons with corresponding descriptions

“...” Description for all kinds of entry fields, control fields, checkboxes, lists, etc.

___ References to Manuals, Figures, Sections, etc.

1.2 VENUS Help and Other Microlab Instrument Manuals

A detailed software reference for can be found in the VENUS Help Function. This tool will answer any question regarding details of the VENUS Software.

Total Aspiration and Dispensing Monitoring (TADM) is an additional safety tool helping to improve the reliability of pipetting processes. Its functions and programming methods are described in the TADM Manual.

The VENUS Dynamic Scheduler is a software tool for organizing and controlling the workflows of a laboratory equipped with Hamilton and other manufacturers' instruments.

VENUS Database Plus is a software tool that allows tracking over multiple runs which is also used when data is put in operation from a remote server.

Guidelines on how to operate the Microlab STAR or Microlab VANTAGE instrument correctly and safely are described in the corresponding Microlab instrument Operator´s Manual.

Further manuals for specific applications, standard solutions, workstations, features, etc. come together with the dedicated Microlab STAR or Microlab VANTAGE instrument. Refer to these specific manuals for the respective application.

The manner in which the Microlab instrument and its components must be serviced is described in the corresponding Microlab instrument Service Manual. This manual will be made available to Hamilton-trained Field Service Engineers.

1.3 Use of the Microlab STAR, Microlab STAR^{V}, and Microlab VANTAGE

The Microlab STAR, Microlab STAR^{V}, and Microlab VANTAGE are robotic pipetting workstations used for pipetting liquid samples in an automated process suitable for medium to high throughput with a high degree of flexibility in pharmaceutical, veterinary, genomics, proteomics, cellomics, and other related fields.

The Microlab STAR, STAR^{V}, and VANTAGE are classified as a general laboratory instruments and are not *in vitro* diagnostic devices.

NOTE

Please refer to the corresponding Microlab instrument Operator's Manual to know the main safety regulations, main hazards involved, electrical and biological properties as well as how to operate the product.

Perform test runs: i) with deionized water and ii) with the final liquids, prior to routine use. Test for all the liquid classes to be used.

An interruption to the power supply during a run may cause loss of data. If data loss is unacceptable, use an Uninterruptible Power Supply.

To avoid computer breakdowns, ensure that there is always enough storage capacity on the hard drive.

ATTENTION

Opening the front cover during a run will lead to a system abort and may cause data loss.

1.4 VENUS Software

The software is a Windows™-based, menu-driven interface, allowing the user to define deck layouts and methods as well as to run the Microlab instrument.

VENUS Software allows the user to program and run different methods for aspirating and dispensing liquids, including control to the accessories such as a wash station or a vacuum system.

NOTE

Each programmed method has to be validated by the programmer.

Note that the methods developed for one particular instrument and configuration do not automatically fit to another instrument or configuration. It might be necessary to make adaptations due to differences in hardware. Please validate imported methods on the target instrument as well.

1.5 Version History

Version	Release Date	Description
00	06/2023	First Release

2 VENUS Software Installation

2.1 Overview

NOTE

VENUS Software is validated for the following Windows™ Operating Systems:

- *Windows 10 (64-bit)*
- *Windows 11 (64-bit)*

ATTENTION

Administrator access to the computer must be allowed before any software installation.

ATTENTION

Disable all antivirus programs before installation. Scripts used in the installation process can interfere with the virus protection software.

2.2 VENUS Software Installation Procedure

ATTENTION

As a safety precaution, before upgrading the system with the new software version, make sure to create back-up copies of the methods, user-defined labware, liquid classes, etc. The easiest way to do so is to export the method/workflows into a package file (refer to Section 5.1.4 Method Export and Import).

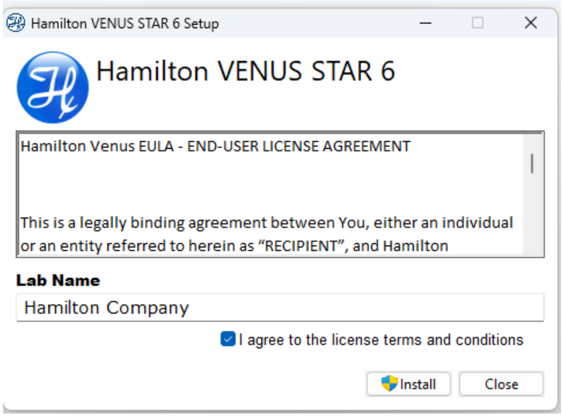
NOTE

The chosen settings (except the selected instrument) are configurable via the Configuration Editor at a later time, if necessary.

1. To back-up methods, etc., rename the Hamilton sub-directory to HAM or any other name that can easily be recalled. After doing so, proceed with the installation.
2. Insert the VENUS Software supplied on a USB Stick into a USB Port.
3. If the installation does not start automatically, select the **Venus.Vantage.Installer.exe** or **Venus.STAR.Installer.exe** program file found in the USB stick. Follow the instructions during installation carefully.

2.2.1 Installation - End User License Agreement (EULA)

1. The installation requires the user to agree with the license terms and conditions to get started.



2. Click [**Install**] to proceed with the installation.

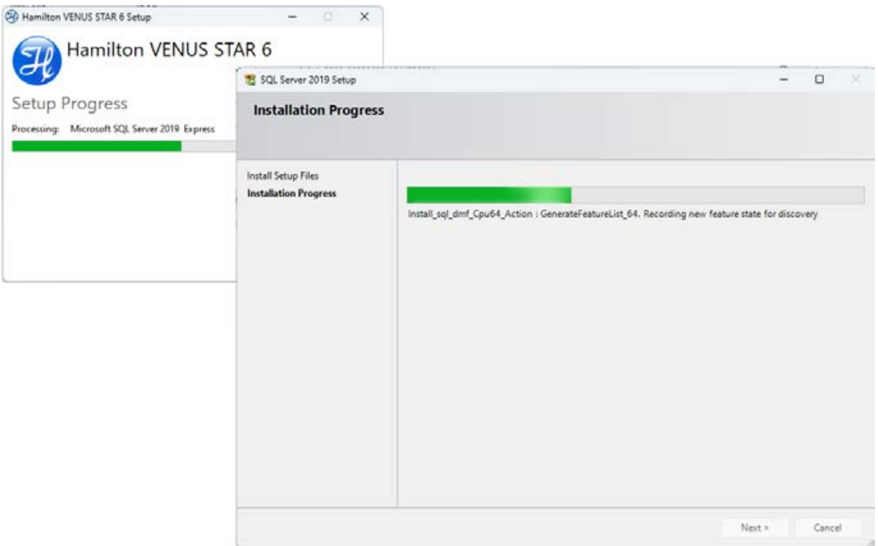
ATTENTION

The installation may vary depending on the operating system used. Some necessary software prerequisites (delivered with VENUS Software) may require a reboot of the PC during installation.

If a software prerequisite is already part of the Operating System, it will not be installed twice.

2.2.2 Installation - Microsoft SQL Server

Using VENUS Software requires access to a Microsoft SQL Server. If the "Microsoft SQL Server 2019 Express Edition" is not installed at this point, a prompt will be displayed during the SQL Server installation.



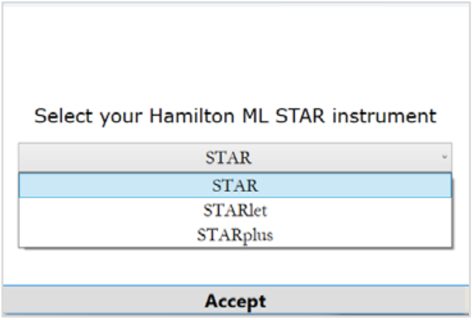
ATTENTION

"Microsoft SQL Server 2019 SP2 *Express Edition" has a limited storage capacity of 10GB.*

Make sure that the SQL Server is maintained on a regular basis within routine use.

2.2.3 Installation - Instrument Selection

1. If installing VENUS for Microlab STAR, during the installation process a prompt will be displayed requesting to indicate the Microlab instrument type: STAR, STARlet or STARplus.



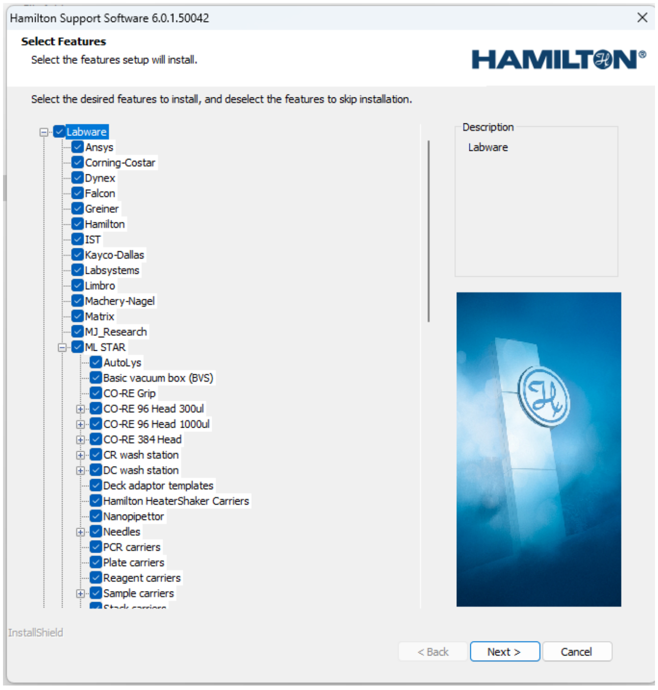
If installing VENUS for Microlab STAR^V or VANTAGE, no instrument selection will be prompted during the installation.

2.2.4 Installation - Support Software

At the end of the VENUS Software installation process, the Hamilton Support Software will start automatically to install additional labware and libraries.

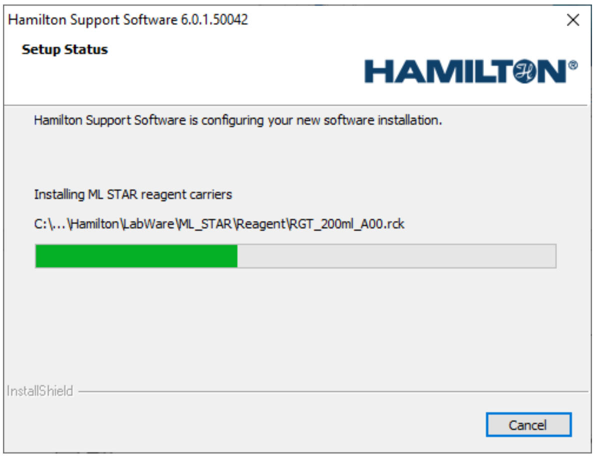
1. Select the labware, libraries and methods which are needed for the daily work by ticking the appropriate boxes.

2. Click [Next >] to continue.

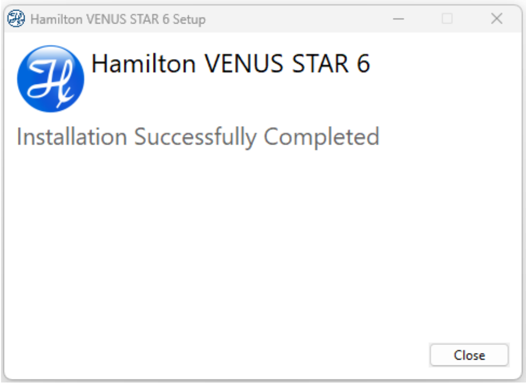


The Hamilton Support Software can also be started through the Hamilton Method Editor > Tools menu.

3. Click [Install] to complete the installation of the selected features for the Hamilton Support Software. The installation may take a few minutes.



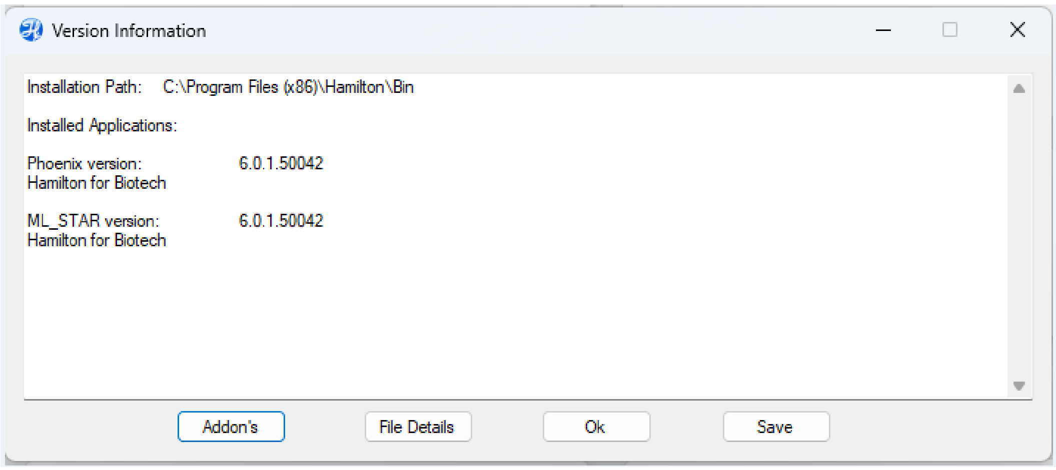
→ A final prompt will notify that the installation is completed.



Finding Version Information after Installation

1. Open the VENUS software by clicking on the desktop icon

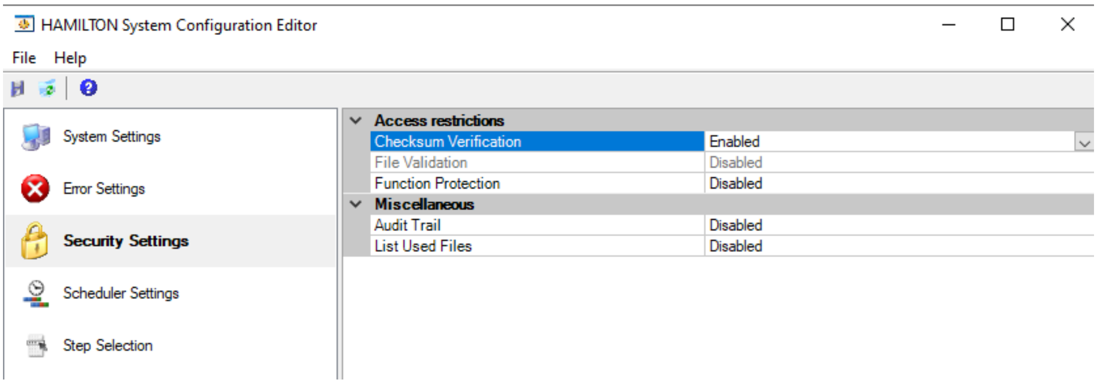
2. Click on System Tools and open the Hamilton Version tool.



2.2.5 21 CFR Part 11 Security Settings

The security requirements can be defined in the System Configuration Editor that can be accessed from the System Tools page of the VENUS software.

1. Open the VENUS software by clicking on the desktop icon
2. Click on System Tools and open the System Configuration Editor
3. Open the Security Settings section



- **Restrict Functionality by User logon (Function Protection)**

  Enable the "Function Protection" option to restrict access to different software functions based on the Windows user logged on. For more information about the controlled access functionality, refer to Section 2.4 Authentication Systems. The access rights of the current user are monitored.

  By installation default, it is set to: **OFF**

ATTENTION

If the option 'Restrict functionality by user logon' is used, access rights must be defined after software installation. See Section 2.4 Authentication Systems.

- **Record (all) File Names in the Runtime Trace** (Listed Used Files)

  Enable the "Listed Used Files" option to include in the run trace (Log File) all the linked file names such as deck layout, liquid classes, labware, etc. at the end of each method run.

  By installation default, it is set to: **OFF**

- **Use File Checksums to Validate Files** (Checksum verification)

  Verify the checksum of the method and of all the linked files such as the deck layout, liquid classes, labware, etc. If a file is corrupted or it has been externally manipulated, the software will detect it. Enable/disable checksum verification of files.

  By installation default, it is set to: **ON**

- **Enable Versioning and Validation of Files (File Validation)**

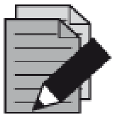
  By installation default, it is set to: **OFF**

- **Enable Viewing of File History (Audit Trail)**

  When enabled, this setting can be set to ‘Always’ to force the audit trail of all saved changes.

  By installation default, it is set to: **OFF**

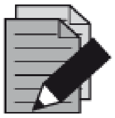
2.3 Oracle Database Installation



NOTE

The Installation of an Oracle Database requires the installation of the VENUS Database Plus 1.1 package. VENUS Database Plus 1.1 is a tool for tracking samples over multiple runs or multiple instruments throughout the lab. Database plus can be used on both local and remote servers. It allows easy access to run data, reliable sample tracking and customized reporting.

To set up a database connection with an Oracle server, the Oracle server has to be installed and connected first. Also, a local client has to be installed (up to version 11g).

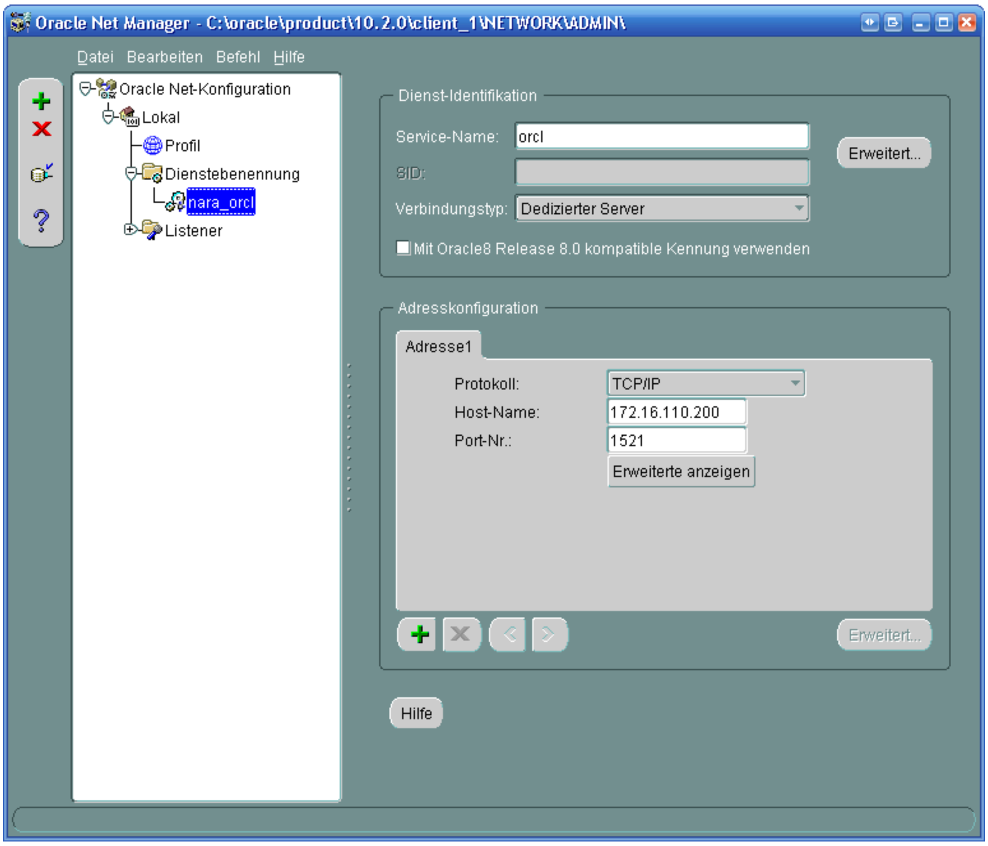


NOTE

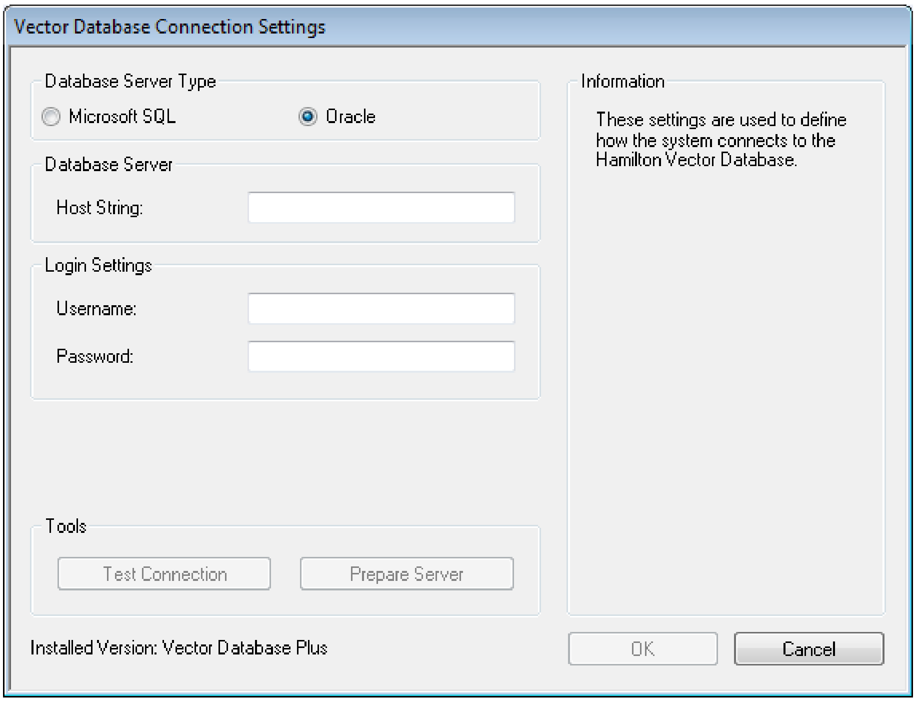
For Oracle installation and server set-up, please refer to the appropriate Oracle information: www.oracle.com

Example for Oracle Services (TNS name orcl)

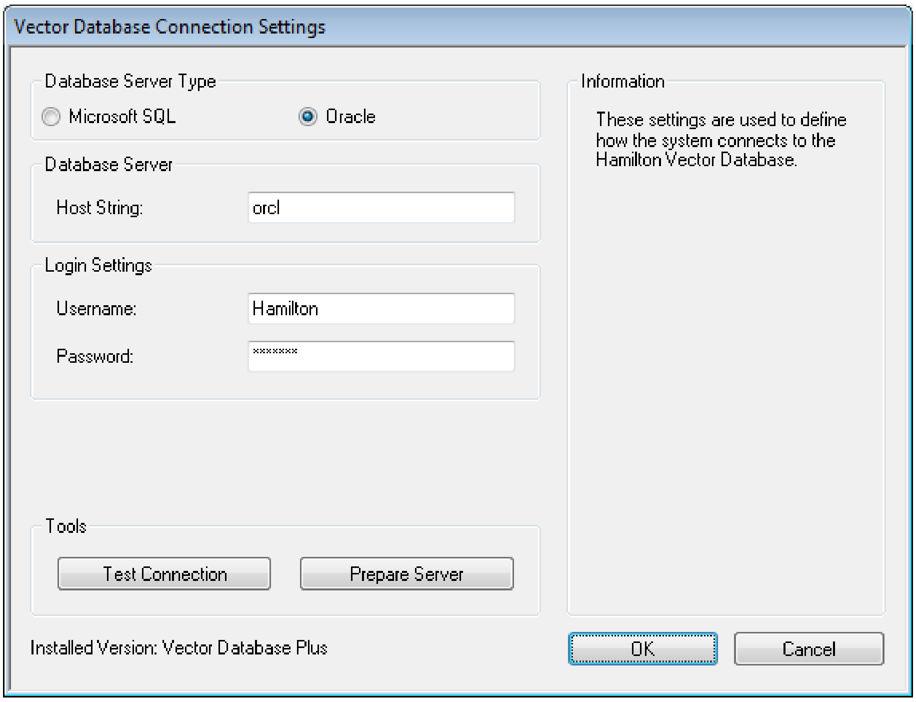
1. Open the Vector Database Connection Settings found in the System Settings of the Hamilton System Configuration Editor and switch the database server type to Oracle.



2. Enter the TNS name of the server into the Host String text field and specify a username and password.



3. Type in the settings specified in the image below.



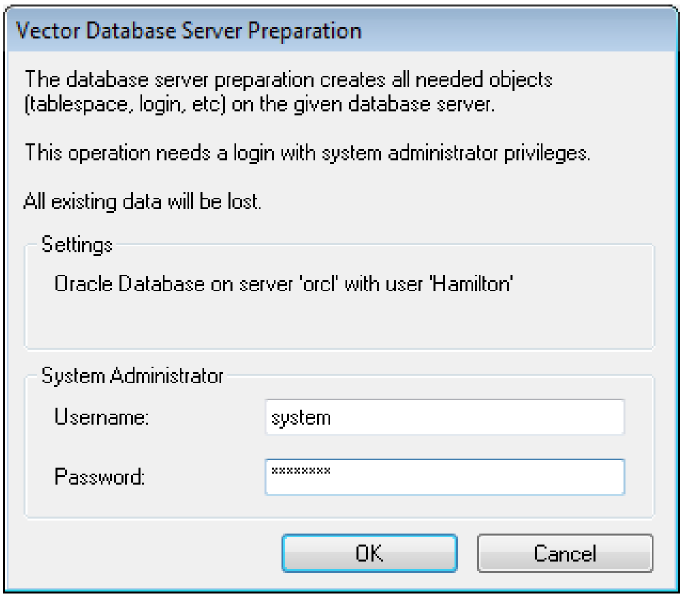
<u>Values taken out from the example screen above:</u>

TNS Name: **orcl**
Username: **Hamilton**
Password: **p@ssword:B7402**

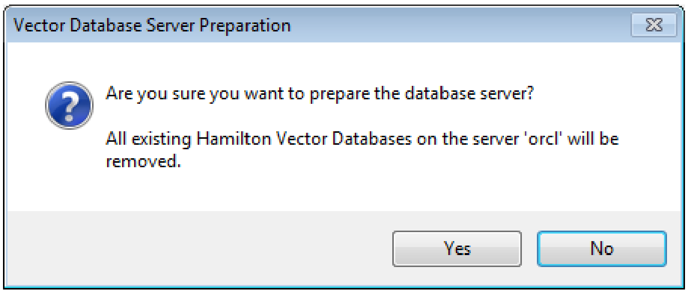
Alternatively, a host string can be used if there is no TNS name matching to the host string, for example:

(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=172.16.110.200)(PORT=1521)))(CONNECT_DATA=(SIC=orcl)))

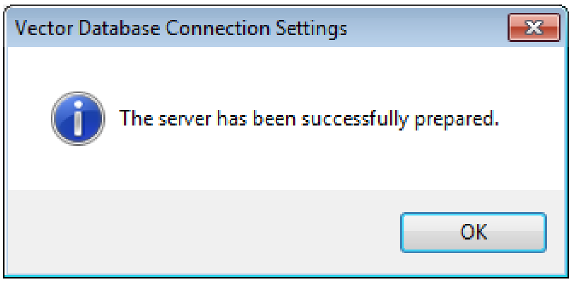
4. Prepare the server for the connection. Enter the username and password of the server administrator.



5. Confirm to build up the database server.



6. The dialog message should be prompted immediately after.



7. The database connection can be checked using the **[Test Connection]** Button found in the "**Vector Database Connection Settings**".

2.4 Authentication System

Hamilton provides an authentication for the Hamilton software based on Windows user groups.

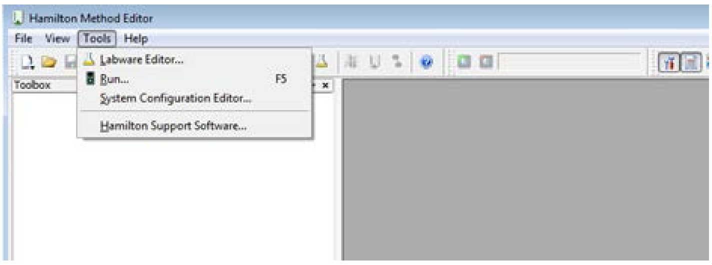
When enabled, the access to different VENUS software functions will be restricted depending on the Windows user and the group the user belongs to.

ATTENTION

*Access rights must first be defined for the "**Lab Service**" before any changes can be made in the System Configuration Editor. Refer to the <u>Section 2.4.2 VENUS user groups in Windows</u>*

2.4.1 Enabling the Authentication System

1. Open the System Configuration Editor from the VENUS home screen > System Tools page or from the Method Editor > Tools menu.

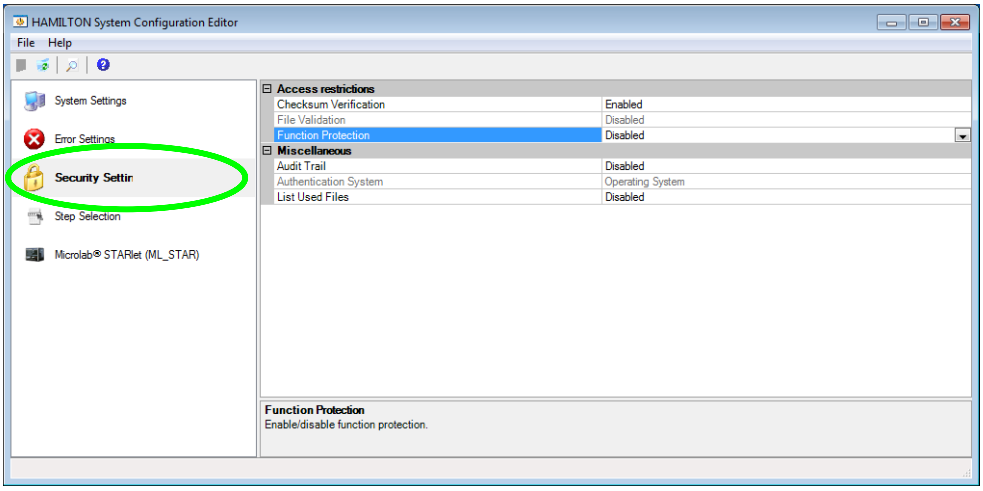


2. Proceed to “**Security Settings**” and specify the settings shown below.

3. Click the “**Function Protection**” entry to enable it.

4. Select the desired authentication system.

5. Click the “**File → Save Changes**” Menu, to save the changes made.

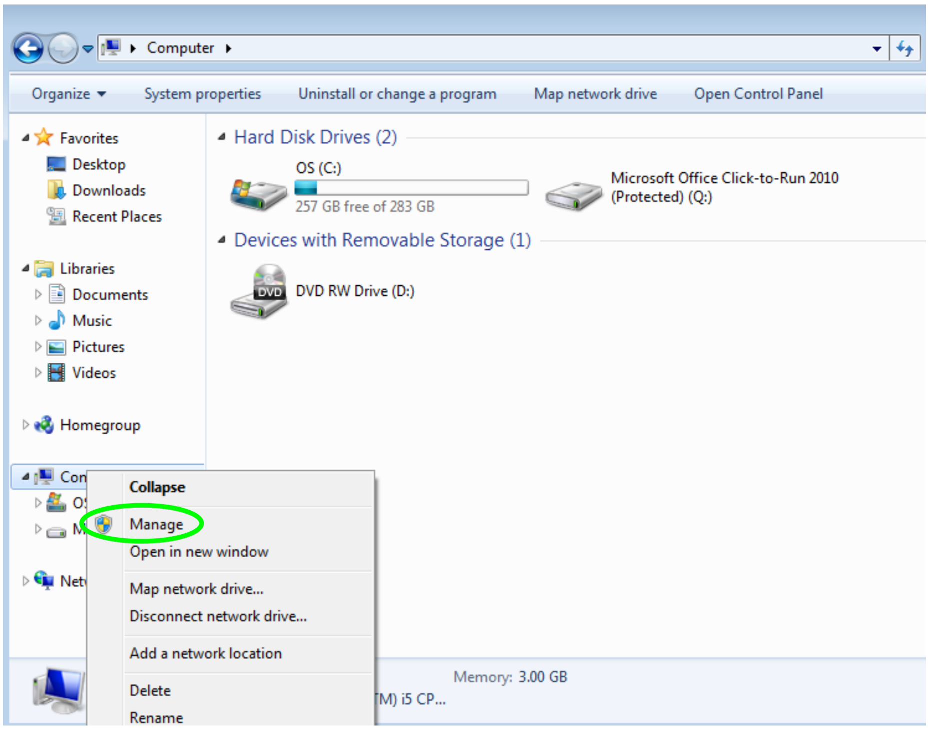


2.4.2 VENUS user groups in Windows

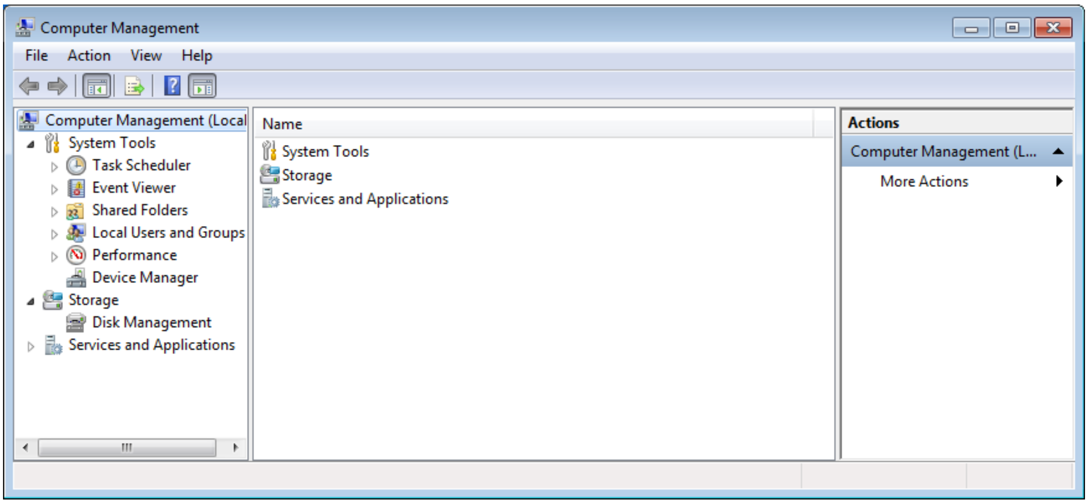
VENUS Software has five different levels of access authorization.

Group	User	Authorization
"Lab Operator"	Routine User	Operators may run any method
"Lab Operator 2"	Routine User	Operators 2 may run any method and move elements on the deck
"Lab Method Programmer"	Method Programmer Laboratory Manager	Method Programmers may modify method and labware definitions
"Lab Remote Service"	LAN Manager PC Administrator	Remote Service
"Lab Service"	Service Engineer Laboratory Manager	Hamilton-trained Field Service Engineer

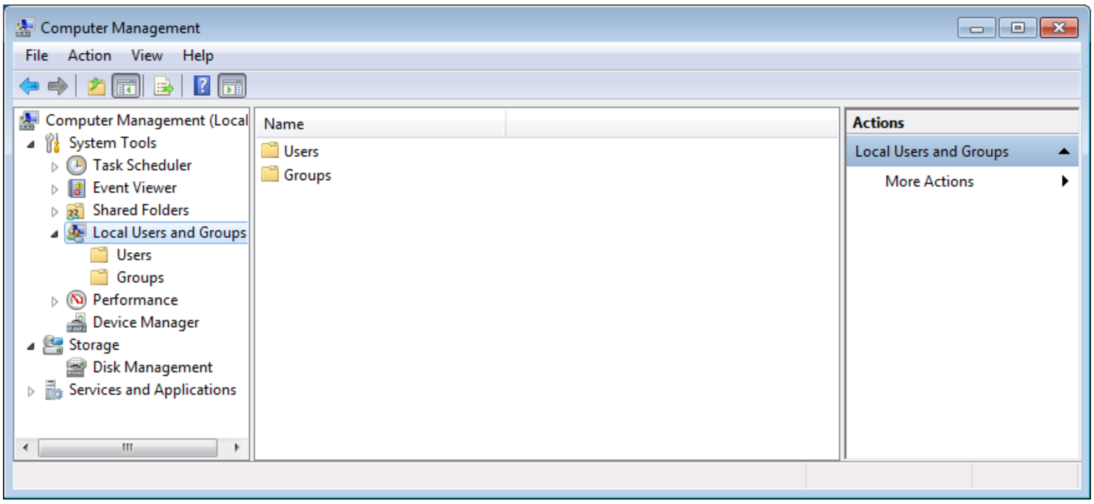
1. Start by navigating through the explorer. Right-click on "**Computer**" and select "**Manage**" from the list.



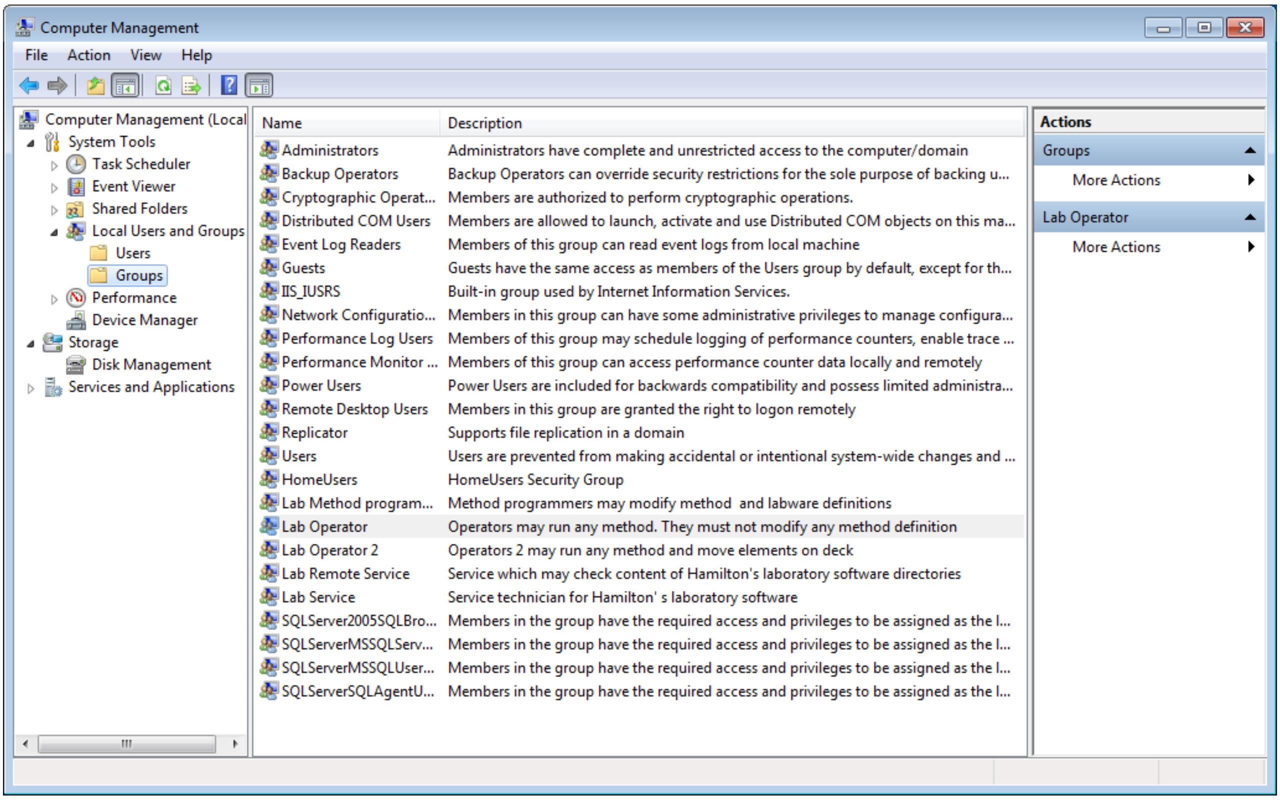
2. The “**Computer Management**” Window will be displayed.



3. Open “**Local Users and Groups**”.

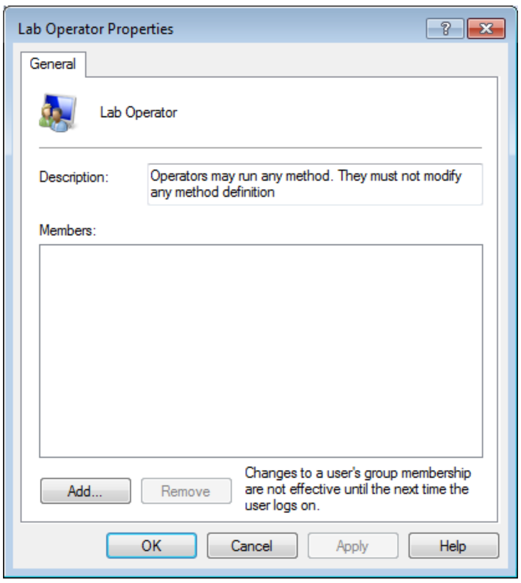


4. Click on "**Groups**" to display the list of defined groups. Groups such as "**Lab Operator**", "**Lab Operator2**", "**Lab Method Programmer**" and "**Lab Service**" can be found on the list.



Here, all users engaged in routine laboratory work (members of the "**Lab Operator**" group) must be keyed-in to obtain the appropriate access rights.

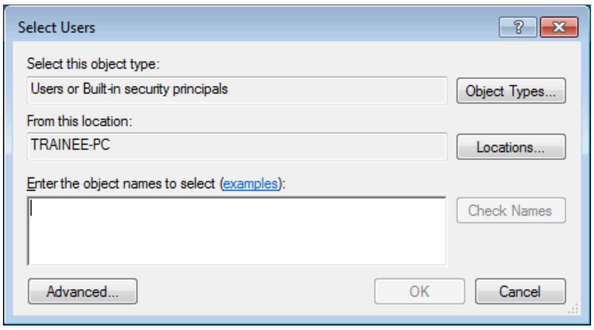
5. Double-click on "**Lab Operator**". The "**Lab Operator Properties**" window will appear. Click the **[Add]** Button.



ATTENTION

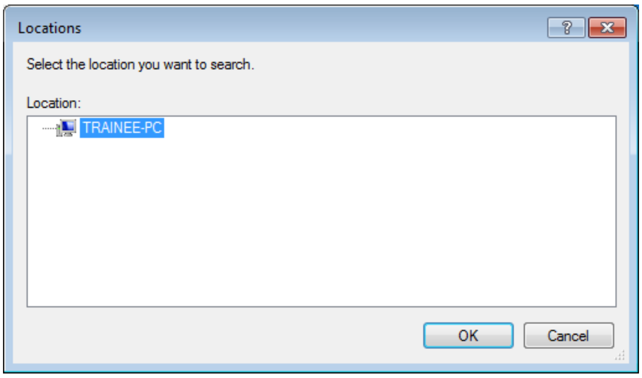
Decide if the user is to be defined only locally or also for the domain (when the PC is in a network environment).

6. On the “**Select Users**” Screen, click the **[Locations…]** Button.



Select the correct “**Location**” source for the laboratory and click **[OK]**.

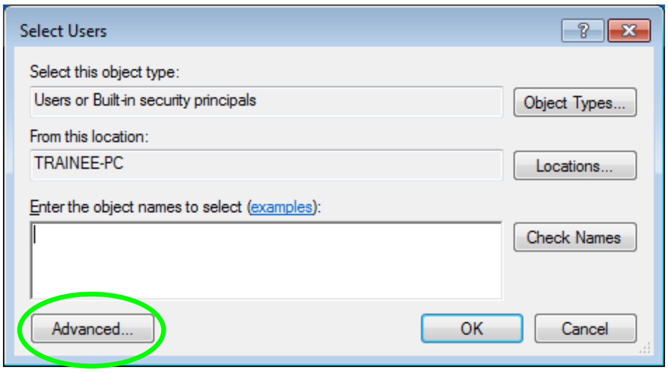
7. The entry “**From this location**” in the “**Select Users**” Window is then filled in.



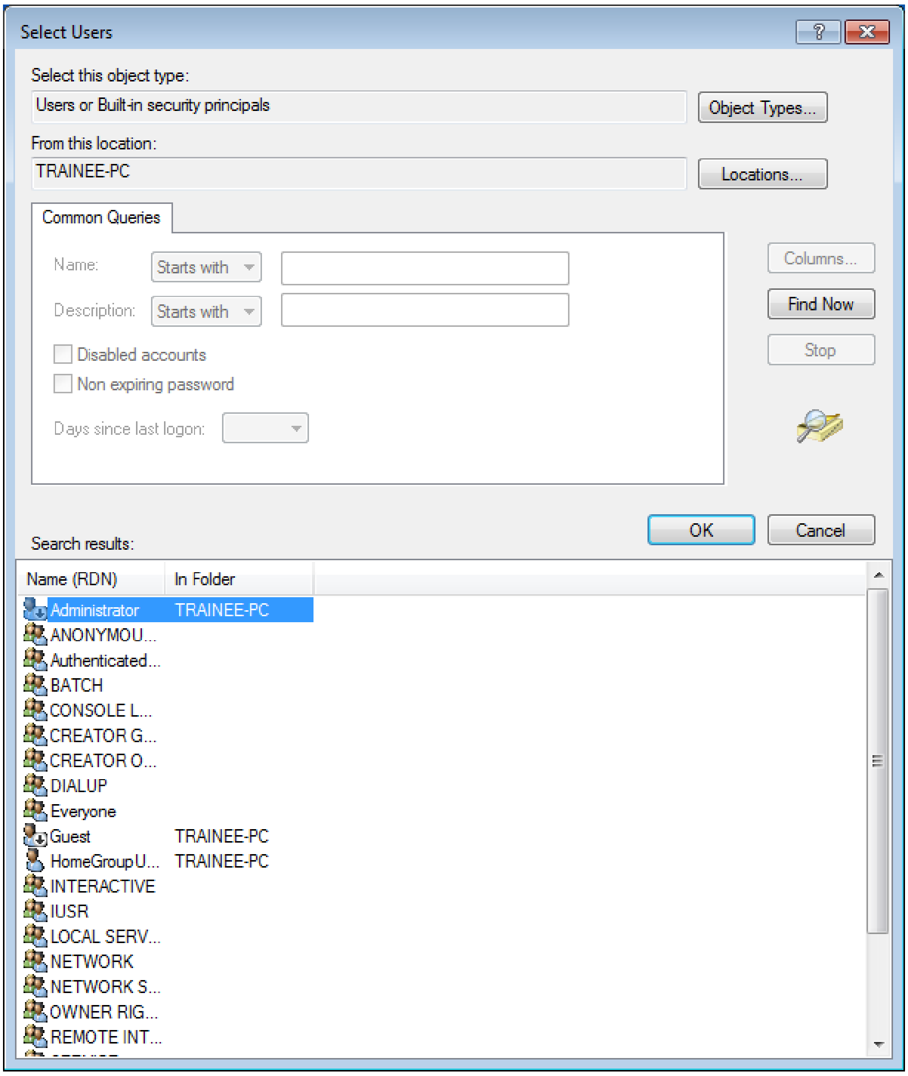
NOTE

*The content on the “**Locations**” Screen will be different for each customer location.*

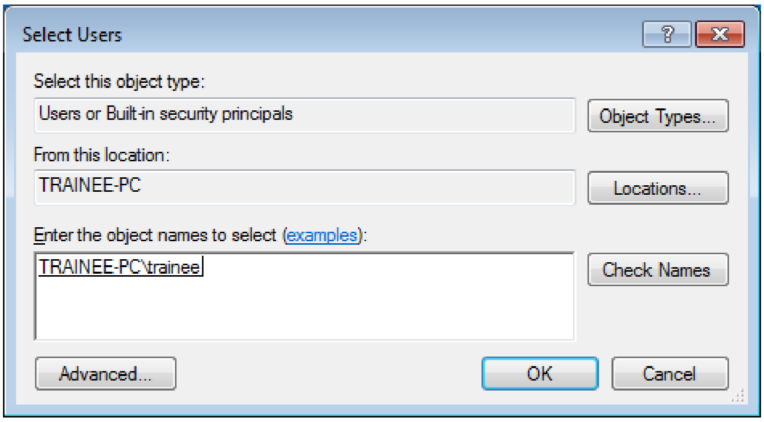
8. Click the **[Advanced]** Button.



9. Either click **[Find Now]** or type the complete name or the beginning of a name in the "**Common Queries**" Text Field, this will search for a matching name. Click **[OK]**.

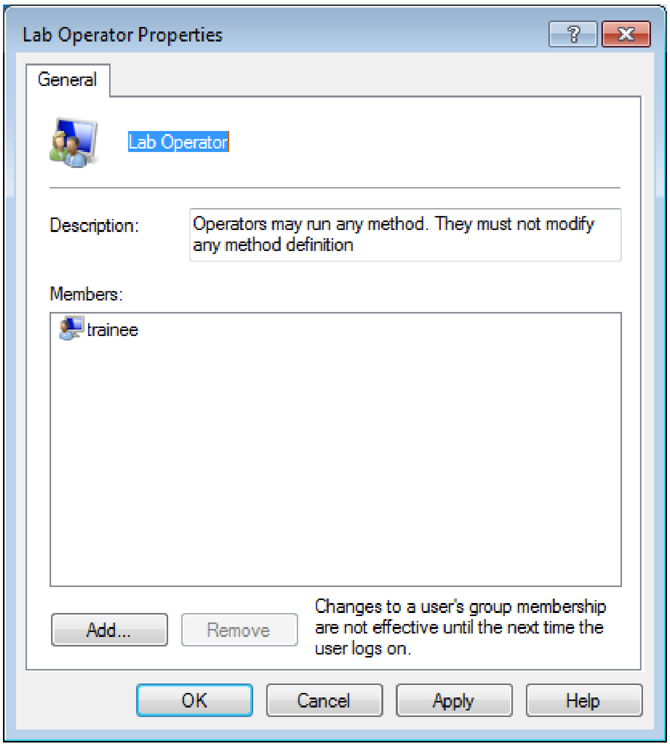


10. Once the name is found, select **[OK]**, and the person will be added to the "**Select Users**" screen.



11. Repeat the previous steps to add additional names. Once all required names are included, click **[OK]**.

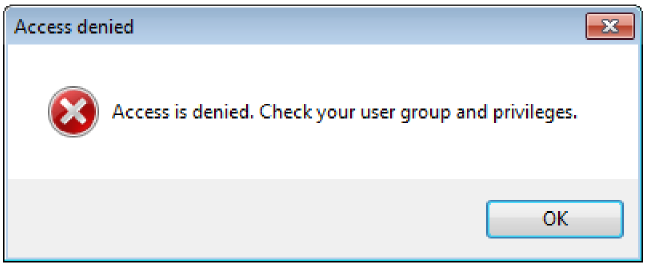
12. The selected user will then be added to the members list. Click **[OK]** to finish.



13. After defining the access rights, log off and then log on again. Check that the installation was successful by logging-on with every specified user.

ATTENTION

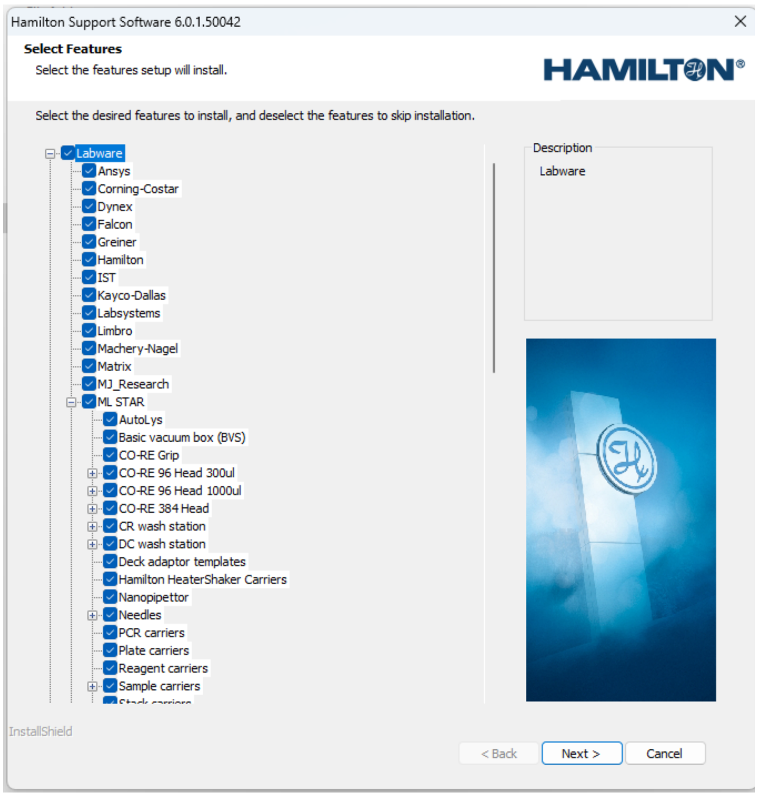
Unregistered users cannot operate the VENUS Software – not even LAN administrators. The following error message is displayed in the case of unauthorized access attempts:



2.5 Adding Labware and Libraries

If not all labware and libraries needed for the daily work are installed, or if there is a requirement for additional labware or libraries, add elements using either the Method Editor, the Deck Layout Editor or the HSL Method Editor. To add labware or libraries, follow the steps below:

1. Select "**Tools → Hamilton Support Software...**" from the menu bar.
2. Select "**Modify**" on the welcome screen, then **[Next >]**.



→ A dialog box which lists all installable components in a tree view appears.

3. Click on a “⊞” sign to open a branch and mark / unmark the required components by ticking the boxes.
4. Click **[Next >]** to install the selected items.

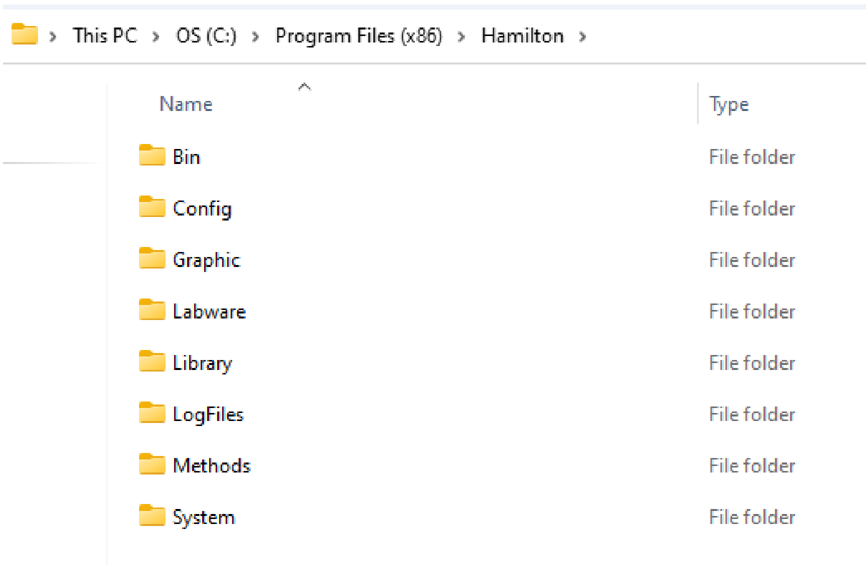
Group	Content
Labware	Labware (e.g. plate carrier, tip carrier, sample carrier)
Methods	Methods
HSL Libraries	HSL Libraries
Service Enhancements	System Debugger for HSL
Old Labware	Labware delivered by older versions of Hamilton Vector / VENUS Software → possibly needed for upgrades

NOTE

A checkbox next to a plus sign does not mean that all possible selections in the tree structure below are also checked.

2.6 File Structure

The installation creates the following default directory structure.



The file consists of the following directories listed in the table shown below.

Directory	Contents
Bin	Executables, DLLs
Config	Configuration files
Graphic	Icons and bitmaps for the software
Help	Help Files
LabWare	Labware definitions
Language	
Library	HSL libraries of commonly used additional functions
LogFiles	Communication traces Method traces (*.trc) Sample tracking result and report files
Methods	Methods (*.med, *.hsl) Deck layouts (*.lay)
System	HxElementCounter.mdb
C:\Barcodes	In addition, this directory is used by the sample tracking to store Microlab AT-like barcode and register files

PART II: The Microlab Instrument VENUS Software

3 Hamilton System Configuration Editor

The Microlab STAR, STAR^{V}, and VANTAGE are available in several configurations. Use the Hamilton System Configuration Editor to configure the VENUS Software to a specific instrument configuration. The primary configuration will be performed by the Hamilton-trained Field Service Engineer during the instrument installation.

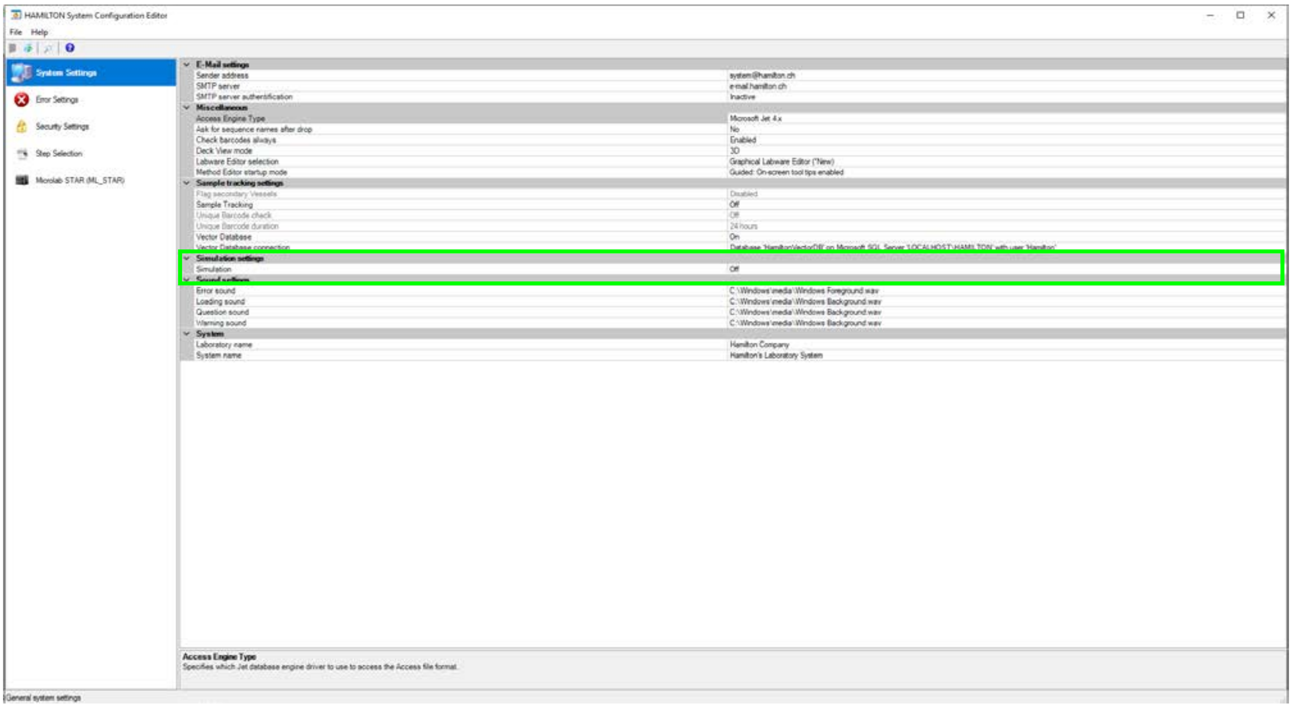
The simulator runs with the same configuration as the instrument. If the VENUS Software is installed on a PC without a connected instrument, it is possible to configure the simulator manually to the instrument configuration, for which the method is intended to be written. Some configuration options are not instrument-dependent and have to be edited in any case.

To open the "**Hamilton System Configuration Editor**", start VENUS and navigate to "**System Tools → System Configuration Editor**". It can also be accessed from the Method Editor >Tools menu.

Or double-click the "**Hamilton.HxConfigEditor.exe**" in the directory "**HAMILTON\bin**".

The "**Hamilton System Configuration Editor**" opens as shown in the image below.

When the "**Hamilton System Configuration Editor**" has been activated, the first decision to be made – prior to the run – is whether to run the method with a real instrument or to simulate the run without using the instrument. Run simulation is a tool for testing whether a method has been programmed correctly. The Run Simulator creates a method trace file with pipetting pattern and other useful information.



To switch the run mode, select "**System Settings" → "Simulation Settings**":

3.1 System Settings

In the system settings, specify the following:

- Sender address and server for the outgoing emails
- Data access engine
- Settings for the flagging of secondary vessels and for sample tracking
- Settings for the unique barcode check
- Database connection
- Setting for instrument simulation
- Some sound settings
- Names of the laboratory system

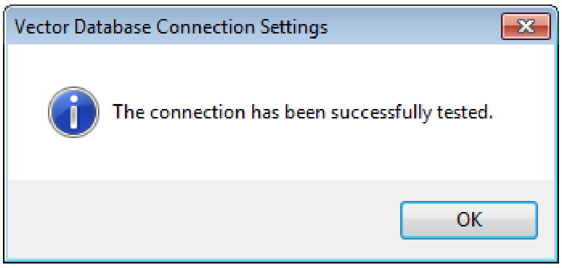
E-Mail settings	
Sender address	system@hamilton.ch
SMTP port	587
SMTP protocol security	None
SMTP server	e-mail.hamilton.ch
SMTP server authentication	Inactive
Miscellaneous	
Access Engine Type	Microsoft Jet 4.x
Ask for sequence names after drop	No
Check barcodes always	Enabled
Deck View mode	3D
Run control selection	Standard Run Control
Runtime settings	
Progress changed interval	1 Seconds
Simulation	On
Sample tracking settings	
Flag secondary Vessels	Disabled
Sample Tracking	On
Unique Barcode check	Off
Unique Barcode duration	24 hours
Vector Database	On
Vector Database connection	Database 'HamiltonVectorDB' on Microsoft SQL Server 'LOCALHOST\HAMILTON' v
Sound settings	
Error sound	C:\WINDOWS\media\Windows Foreground.wav
Loading sound	C:\WINDOWS\media\Windows Background.wav
Question sound	C:\WINDOWS\media\Windows Background.wav
Warning sound	C:\WINDOWS\media\Windows Background.wav
System	
Laboratory name	Hamilton Company
System name	Hamilton's Laboratory System

3.1.1 Vector Database Connection

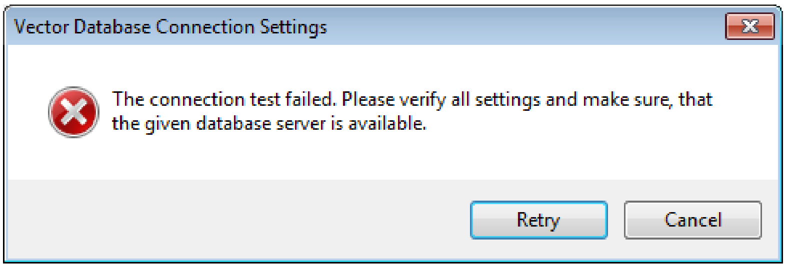
Opening the "**Vector Database Connection**" allows the user to specify the connection to a database or to test the existing settings. These settings are used to define how the system connects to the "**Hamilton Vector Database**".

View mode	3D
Sample tracking settings	
Flag secondary Vessels	Disabled
Sample Tracking	Off
Unique Barcode check	Off
Unique Barcode duration	24 hours
Vector Database	On
Vector Database connection	Database 'HamiltonVectorDB' on Microsoft SQL Server 'L(

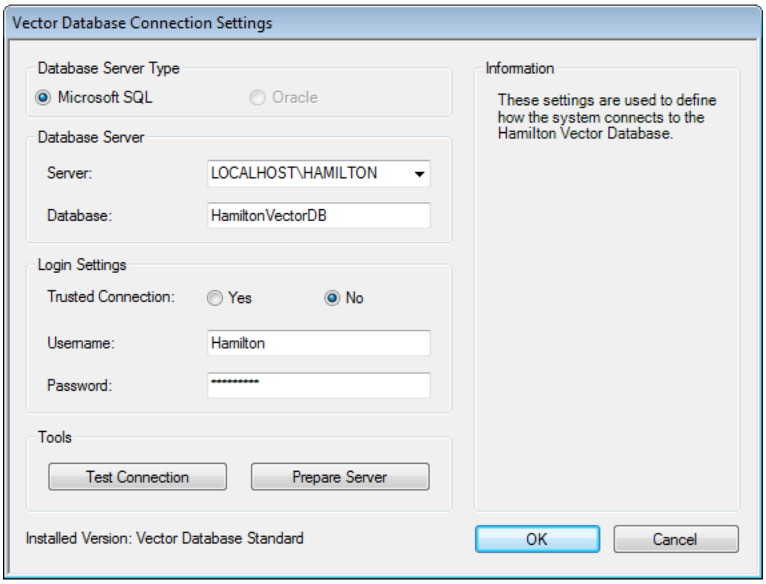
The **[Test Connection]** Tab allows checking of the existing connection. The message dialog shown above will appear when the connection is successful. End with **[OK]**.



A non-working connection will prompt a message dialog shown in the image below.

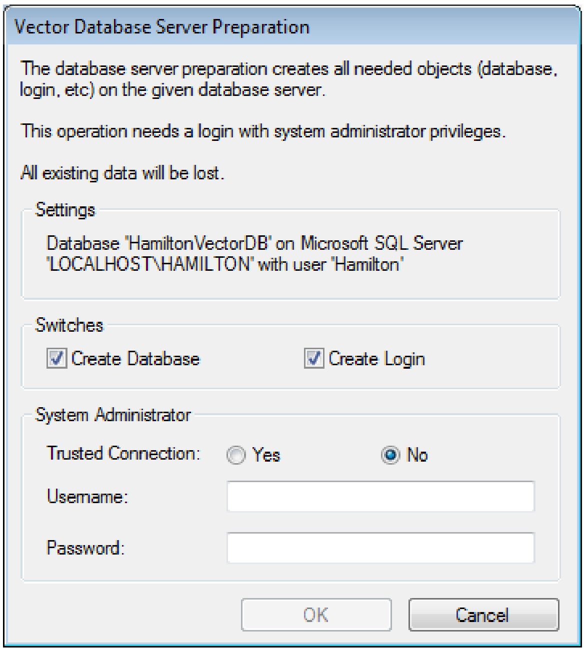


When this occurs, check the settings. The default values are as follows:



Server:	**LOCALHOST\HAMILTON**
Database:	**HamiltonVectorDB**
Trusted Connection:	**No**
Username:	**Hamilton**
Password:	**p@ssword:B7402**

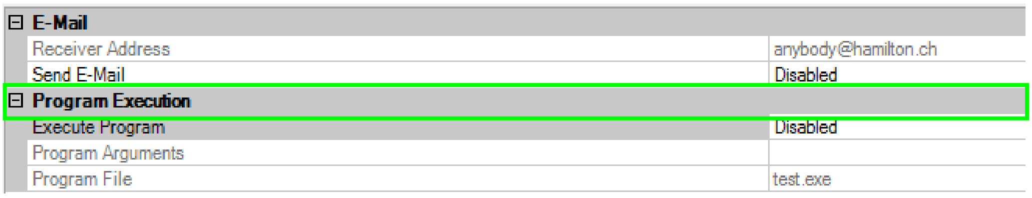
The “**Vector Database Server Preparation**” Screen allows the user to connect to a different server or database. Please make sure that the server to be used is local; otherwise use the “**Hamilton Vector Database Plus**” which allows the connection to a remote server.



Default Settings: Username: **sa**

Password: **HAMILTON:db43**

3.2 Error Settings



The “**Error Settings**” allows a specification of:

- An e-mail address to where a message will be sent in case of an error
- A program which will be executed in case of an error

3.3 Security Settings

In the “**Security settings**”:

- Enable/disable verification of file integrity by checksum.
- Enable/disable file validation (if function protection is enabled)
- Enable/disable function protection
- Enable/disable the audit trail, i.e. a protocol of methods that are run
- Enable/disable the protocol of the files used by methods

Access restrictions	
Checksum Verification	Enabled
File Validation	Disabled
Function Protection	Enabled
Miscellaneous	
Audit Trail	Always
Authentication System	Operating System
List Used Files	Enabled

3.4 Step Selection

The “**Step Selection**” Menu has several sub-sections in accordance with the enabled functions. They allow the including or excluding of method steps such as the “Move” Commands selectable in the method editor.

Please switch to the “**Instrument**” Tab (e.g. Microlab STARlet) to set the visibility of the instrument-related steps (e.g. Single Steps).

3.4.1 General Steps Settings

Here, it is specified which steps will be available in the toolbox for the programmer. These steps are always displayed as soon as a method has been loaded or newly created.

General steps	
General steps	
Abort	Step visible
Adjust Sequences	Step visible
Array: Copy	Step visible
Array: Declare / Set Size	Step visible
Array: Get At	Step visible
Array: Get Size	Step visible
Array: Set At	Step visible
Assignment	Step visible
Assignment with Calculation	Step visible
Begin Parallel	Hidden
Comment	Step visible
Communication Port: Close	Hidden
Communication Port: Open	Hidden
Communication Port: Read	Hidden
Communication Port: Write	Hidden
End Parallel	Hidden
Error Handling by the User	Hidden
File: Close	Step visible
File: Open	Step visible
File: Read	Step visible
File: Set Position	Step visible
File: Write	Step visible
If, Else	Step visible
Loop	Step visible
Loop: Break	Step visible
Return	Step visible

3.4.2 Smart Steps Settings

The “**Smart Step Selection**” Section allows the including or excluding of Smart Steps in the toolbox of the method editor.

Microlab® STAR Smart Steps	
1000µl Channel Needle Eject	Step visible
1000µl Channel Needle Pick Up	Step visible
1000µl Channel Needle Wash Settings	Step visible
1000µl Channel Pipette - Aliquot	Step visible
1000µl Channel Pipette - Pooling (n-1)	Step visible
1000µl Channel Pipette - Replica (1-n)	Step visible
1000µl Channel Pipette - Simple (1-1)	Step visible
1000µl Channel Tip Eject	Step visible
1000µl Channel Tip Pick Up	Step visible
Advanced Load Settings	Hidden
Load	Step visible
Load and Match	Step visible
Unload	Step visible

3.4.3 Data Handling Steps Settings

Data Handling Steps	
Generate Mapping File	Step visible
Import Worklist	Step visible
Remove Labware	Step visible
Set Labware Barcode	Step visible
Update Job Status	Step visible

The “**Data Handling**” Section allows the enabling or disabling of the following functions:

- Generate a Mapping File
- Import Worklist
- Remove labware
- Set Labware Barcode
- Update Job Status

3.4.4 Custom Dialog Steps Settings

The “**Custom Dialog Steps**” allow the enabling or disabling of the function:

- Custom Dialog

Custom Dialog Steps	
Custom Dialog	Step visible

3.5 Microlab Instrument Settings

3.5.1 Barcode Settings

The “**Barcode settings**” allows:

- Selection of a plate to assign the barcode to if multiple plates are on the same carrier position
- Enabling or disabling of the generation of a barcode trace file (C:\Program Files\HAMILTON\LogFiles\barcodes_N.txt)
- Selection of the barcode types which are allowed on the instrument

The storage location of the barcode is dependent on the Operating System and the language settings.

Barcode settings	
Ask for labware dialog	On
Barcode trace	Enabled
Used 1D barcode types	
Codabar	Enabled
Code 128 (Subset B and C)	Enabled
Code 2 of 5 Interleaved	Enabled
Code 39	Enabled
Code 93	Enabled
ISBT Standard	Enabled
JAN/EAN 8	Enabled
UPC A/E	Enabled
Used 2D barcode types	
Aztec	Enabled
Data Matrix	Enabled
EAN/UCC Comp	Enabled
GS1 DataBar	Enabled
Maxi Code	Enabled
Micro PDF 417	Enabled
PDF 417	Enabled
QR Code	Enabled

3.5.2 Communication Settings

The “**Communication settings**” Header is only active if “**instrument**” Mode has been selected. In “**Simulator**” Mode, there is no communication port to select.

Communication settings	
Communication interface	USB
RS232 Com port	COM1
Used Instrument	

3.5.3 Instrument Configuration

Instrument configuration	
Automatic locking of front cover	Disabled
Check for free deck	On
Default waste (for old methods)	Waste
Speed up cLLD detection	Off
Teaching with 1000µl Channel no.	8
Teaching with 5ml Channel no.	4

The following can be selected in the "**Instrument configuration**" Menu:

- "**Automatic locking of front cover**" Option
- Check for free deck position before a carrier is loaded
- "**Default waste**" location (accept the default for new methods)
- Speed up cLLD detection (remember liquid height) if sample tracking is enabled
- 1000µl-pipetting channel to teach labware positions
- 5 mL-pipetting channel to teach labware positions

3.5.4 Maintenance Settings

The maintenance procedures can be enabled here. The user of the system is periodically requested to execute preventative maintenance. Hamilton recommends switching on the maintenance control to ensure that the system works reliably. The maintenance procedures are described in the corresponding Microlab instrument Operator's Manual.

Microlab STAR Maintenance Settings

Maintenance settings	
Maintenance control	Optional
Maintenance processes	
00 Daily maintenance	Warning
01 Weekly maintenance	Mandatory
05 Verification (not yet installed)	Disabled
07 Daily Morning Maintenance Nanopipettor	Disabled
08 Start Up Maintenance Nanopipettor	Disabled
09 Decontamination Maintenance Nanopipettor	Disabled

Microlab STAR^V / VANTAGE Maintenance Settings

Maintenance settings	
Maintenance control	Optional
Maintenance processes	
00 Daily maintenance	Mandatory
1001 Track Gripper maintenance	Mandatory

3.5.5 Simulator Settings

Simulator configuration	
1000µl Channel: number of channels	8
1000µl Channel: raster	9mm
5ml Channel: number of channels	0
5ml Channel: raster	18mm
Autoload	1D Reader
Camera Channel	Not available
CO-RE 384 Head	Not available
CO-RE 96 Head	Not available
Gel Card Gripper	Not available
iSWAP	Not available
Labware Handling Capper station 1	Not available
Labware Handling Capper station 2	Not available
Labware Handling Channel: number of channels	0
Labware Handling Channel: raster	18mm
Labware Handling Channel: type	Not available
Nano pipettor	Not available
Pump Station 1 (node HW)	Not available
Pump Station 2 (node HU)	Not available
Pump Station 3 (node HV)	Not available
Punch Card Gripper	Not available
Puncher	Not available
Simulation	Off
Simulator Delay	0%
Status Light	Available
Temperature-controlled Carrier 1	Not available
Temperature-controlled Carrier 2	Not available
Tube Gripper	Not available
Wash station 1	Not available
Wash station 2	Not available

"**Simulator configuration**" allows the selection of the built-in option with which the simulated instrument should run. Make sure that the devices to be used by the system are activated ("**Available**") in the simulation configuration to simulate a method without any error messages.

3.5.6 Step Selection Settings

The "**Step selection**" Section allows the including or excluding of Single Steps in the method editor toolbar.

Step selection	
Commands	
1000µl Channel Aspirate	Step visible
1000µl Channel Aspirate (Single Step)	Hidden
1000µl Channel CO-RE Grip Get Plate (Single Step)	Hidden
1000µl Channel CO-RE Grip Move Plate (Single Step)	Hidden
1000µl Channel CO-RE Grip Place Plate (Single Step)	Hidden
1000µl Channel CO-RE Grip Read Barcode (Single Step)	Hidden
1000µl Channel CO-RE Grip Transport	Step visible
1000µl Channel Dispense	Step visible
1000µl Channel Dispense (Single Step)	Hidden
1000µl Channel Dispense on the Fly (Single Step)	Step visible
1000µl Channel Get Last Liquid Level (Single Step)	Hidden
1000µl Channel Move To Position (Single Step)	Hidden
1000µl Channel Start Needle Wash (Single Step)	Hidden
1000µl Channel Tip Eject (Single Step)	Hidden
1000µl Channel Tip Pick Up (Single Step)	Hidden
1000µl Channel Wait For Needle Wash (Single Step)	Hidden
5ml Channel Aspirate	Step visible
5ml Channel Aspirate (Single Step)	Hidden
5ml Channel CO-RE Grip Get Plate (Single Step)	Hidden
5ml Channel CO-RE Grip Move Plate (Single Step)	Hidden
5ml Channel CO-RE Grip Place Plate (Single Step)	Hidden
5ml Channel CO-RE Grip Read Barcode (Single Step)	Hidden

NOTE

TADM settings are only available if the VENUS TADM feature is enabled. For more information, contact a local Hamilton Representative.

VENUS TADM records and monitors pressure curves for every individual pipetting channel. It defines customized tolerance bands for acceptable pipetting limits and detects a wide variety of pipetting errors and sample conditions. TADM features help in customizing liquid classes and monitor processes at the same time provides the opportunity to look inside the pipetting channel. With VENUS Software, the TADM test aspiration pressure curves are analyzed, to indicate pipetting performance.

4 VENUS Software Tour

VENUS Software allows programming the Microlab STAR or Microlab VANTAGE instrument according to the needs of the workflow and assay. Like any other software from Hamilton, Hamilton-specific terms are used. Some terms are explained in short summaries to make programming easier.

The second part of this section shows an example on how to create programs.

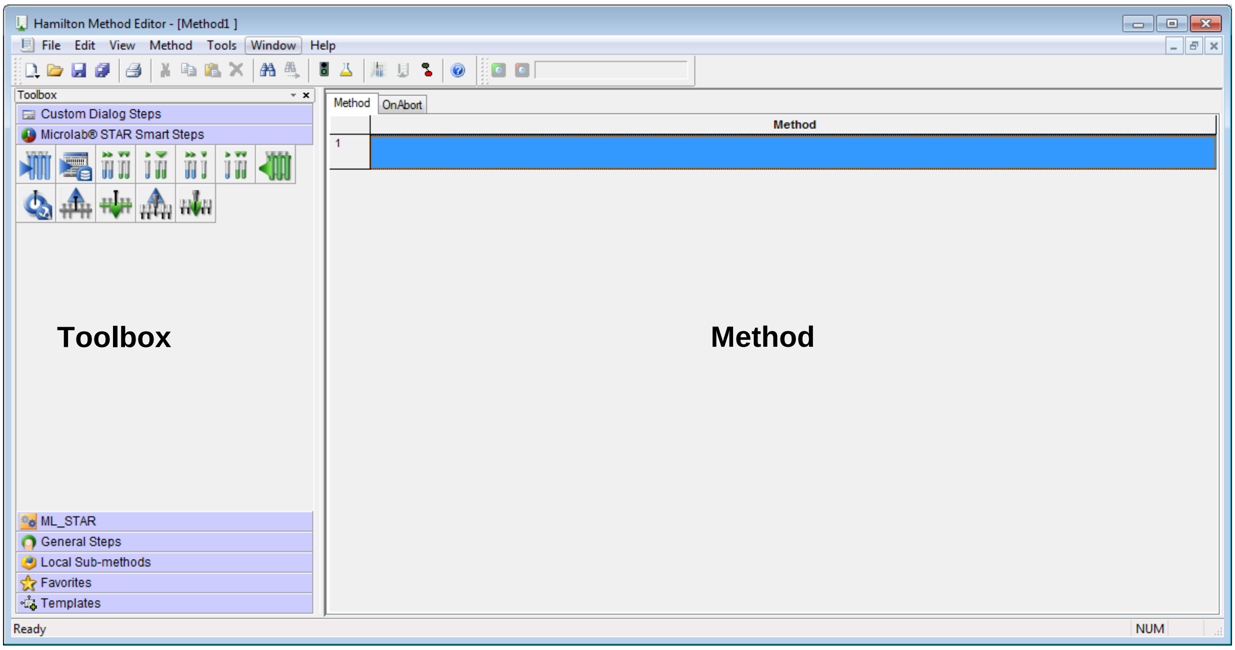
NOTE

Hamilton offers basic and advanced VENUS software training courses. Visit the Hamilton Company website or contact a local Hamilton Representative for more information

4.1 Overview

The master editor called the "**Method Editor**" combines several sub-editors, which helps optimize the workflow and pipetting results. There are two ways of displaying a method.

The first option is to use the step-oriented method editor. Here, the method is created by adding steps from the toolbox to the step-oriented "**Method Editor**".

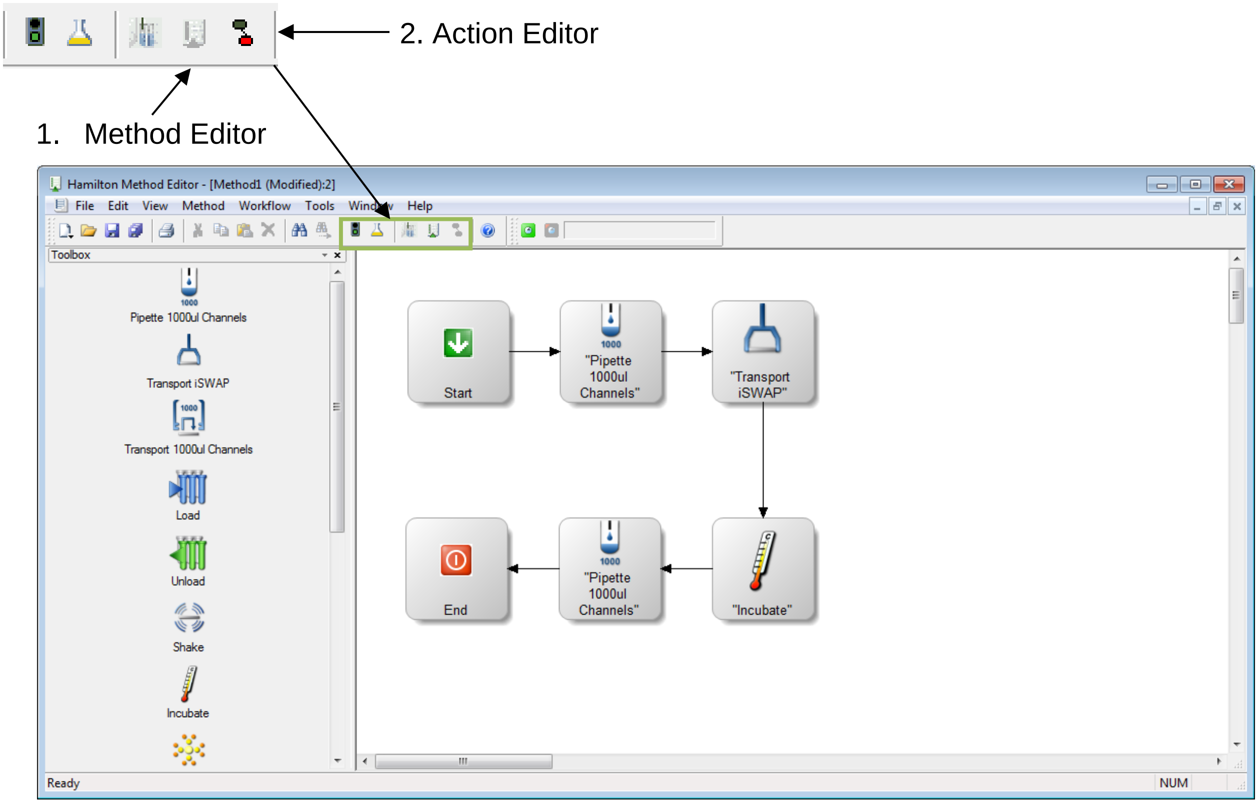


The second option is through the "**Action Editor**". The Action Editor offers a more visual representation of the method.

The selected actions in the toolbox can be dragged-and-dropped to the editor window. Arrows show the chronological order of the process.

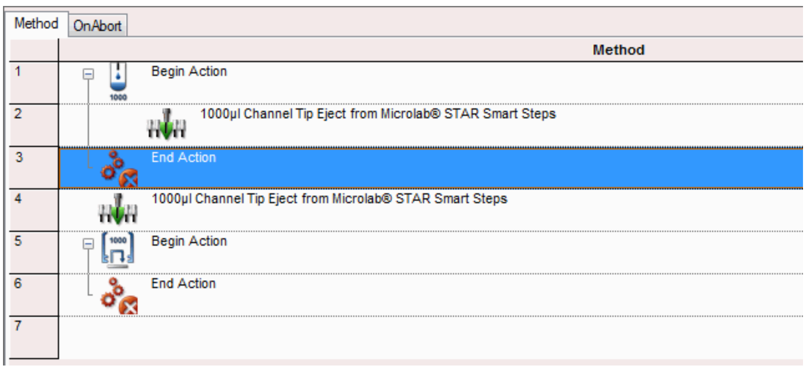
Both editors use the same set of data. The editors, however, use different ways to display data.

To switch between both views, use the following icons:

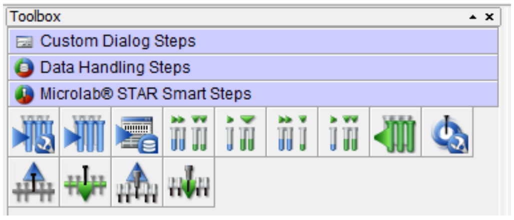


4.2 Methods

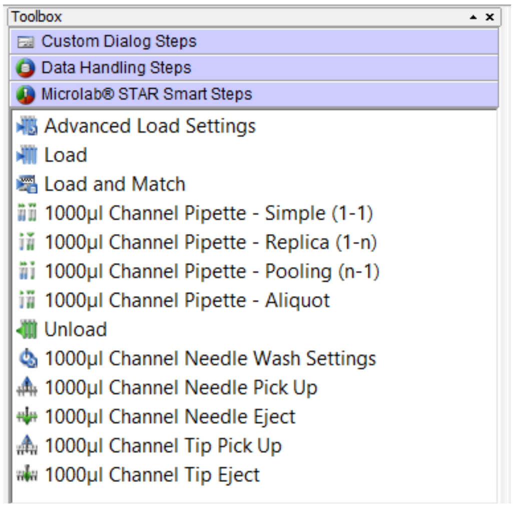
Programming Microlab STAR or Microlab VANTAGE instruments always means creating new or adapting existing methods. A method is a list of instructions for the instrument, appearing as actions, steps, transport steps, loops, user dialogs, etc.



The user software offers various standard step libraries such as the general steps, the pipetting steps, etc. while the instrument software offers different levels of programming.



Smart Steps combine tasks. For example, combining a complete pipetting task with a plate copy, aliquoting, pooling, etc. These commands are best for beginners to become familiar with the system. They incorporate a guided programming wizard, predefined error recoveries and customized recovery strategies.

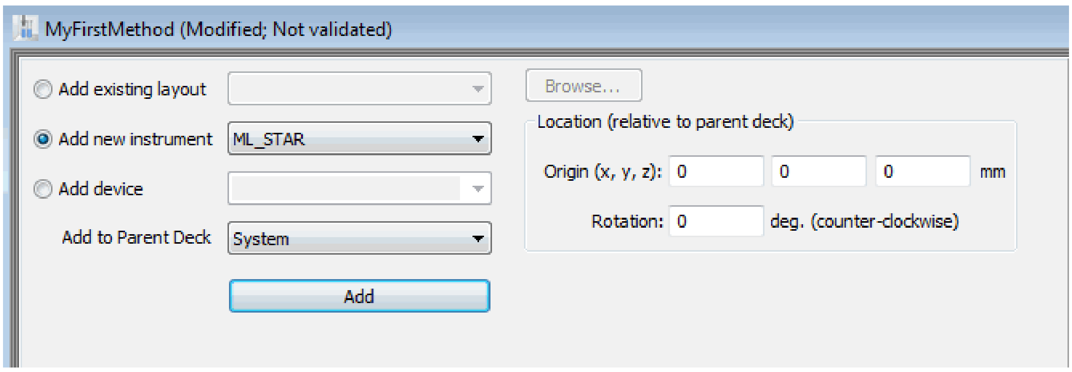


Easy Steps are for pipetting and plate handling. They offer a wider range of settings and possibilities to handle errors than the smart steps. “**Easy Steps**” Icons have yellow backgrounds.

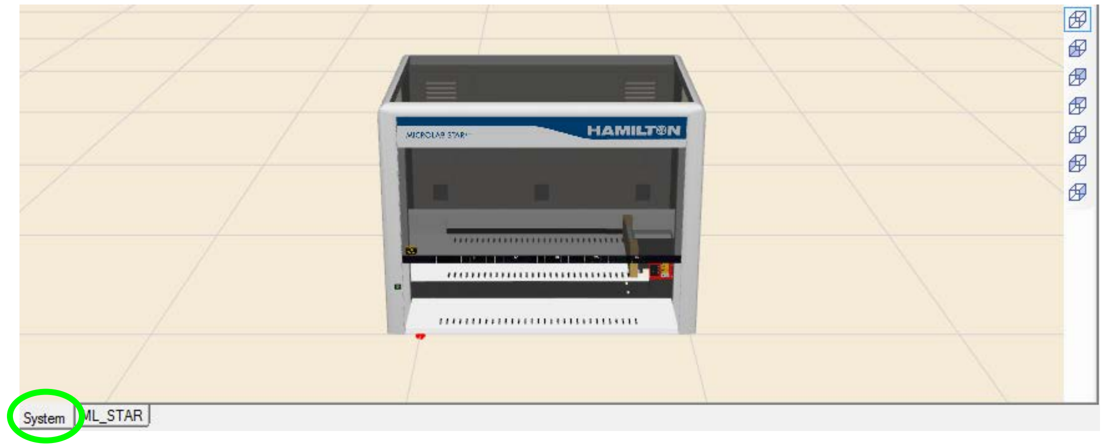
Single Steps are used when highest flexibility of the system is required. These commands allow even the most complex pipetting and plate handling tasks. Single steps have the suffix “**Single Step**”.

Aside from the standard step libraries, additional libraries for advanced programming are also available.

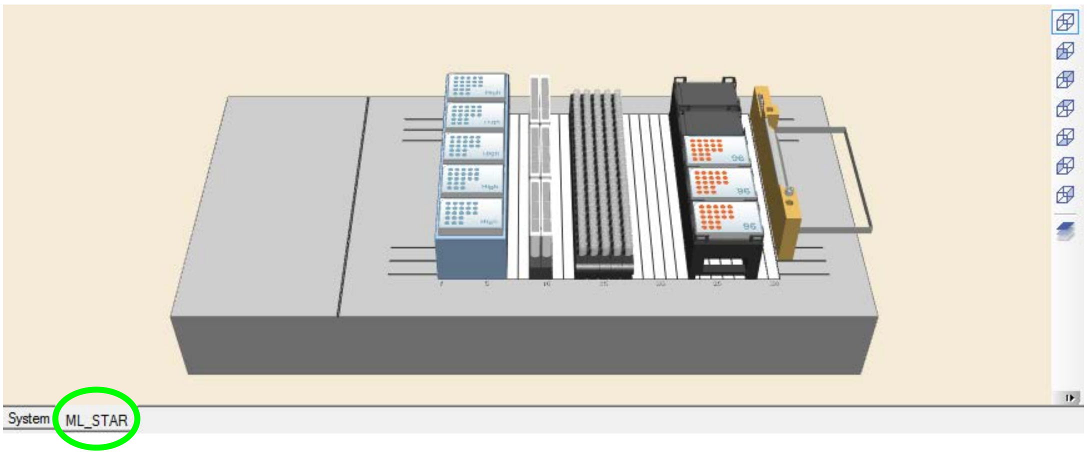
Each method is linked to a **System Deck** which is an empty environment upon opening. Now, the programmer must plot the real environment in the software by first adding an instrument to the system deck.



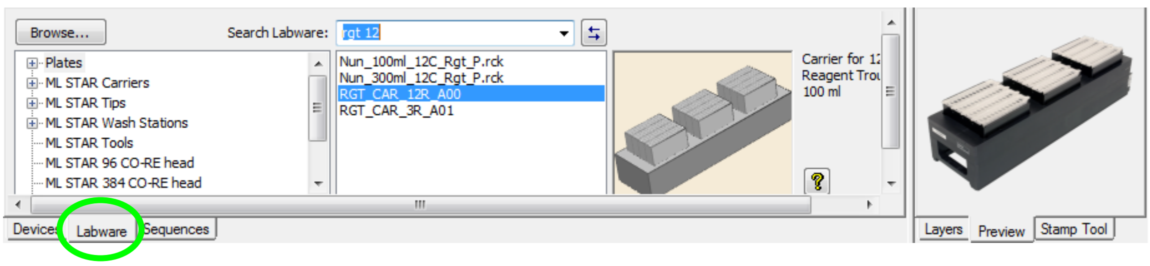
After successfully adding an instrument, choose between two different views in the deck editor. The first view is the **System** Tab which shows an external perspective of the instrument, including third-party components.



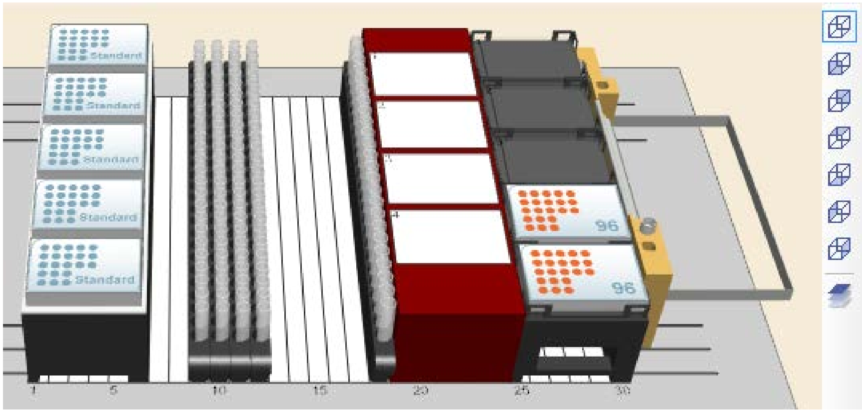
The **ML_STAR** Tab shows an internal perspective of the instrument which is the deck layout. This is a graphical illustration of the work surface of the Microlab instrument. It contains all information about the labware used and x/y/z coordinates of the positions.



The Microlab instrument can be used with various kinds of labware such as: Tubes, Microplates, Reagent Troughs, etc. The software comes with a set of standard labware definitions. Labware is available by clicking the Labware Tab found in the Deck Layout Editor.



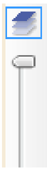
To allow easier access to any kind of labware, use the view selection found on the right side of the labware view window. The deck layout is displayed in 3D and can be rotated to any angle.



Choose among the different views:

- 3D View
- Front View
- Back View
- Top View
- Bottom View
- Left View
- Right View

The clipping function will allow cutting-off overlaying labware from the labware placed underneath. This makes the creation of, for example, sequences very simple.



Please make sure that the deck view has an appropriate size at different z-heights (CVS / BVS); otherwise, the clipping slider will not be shown.

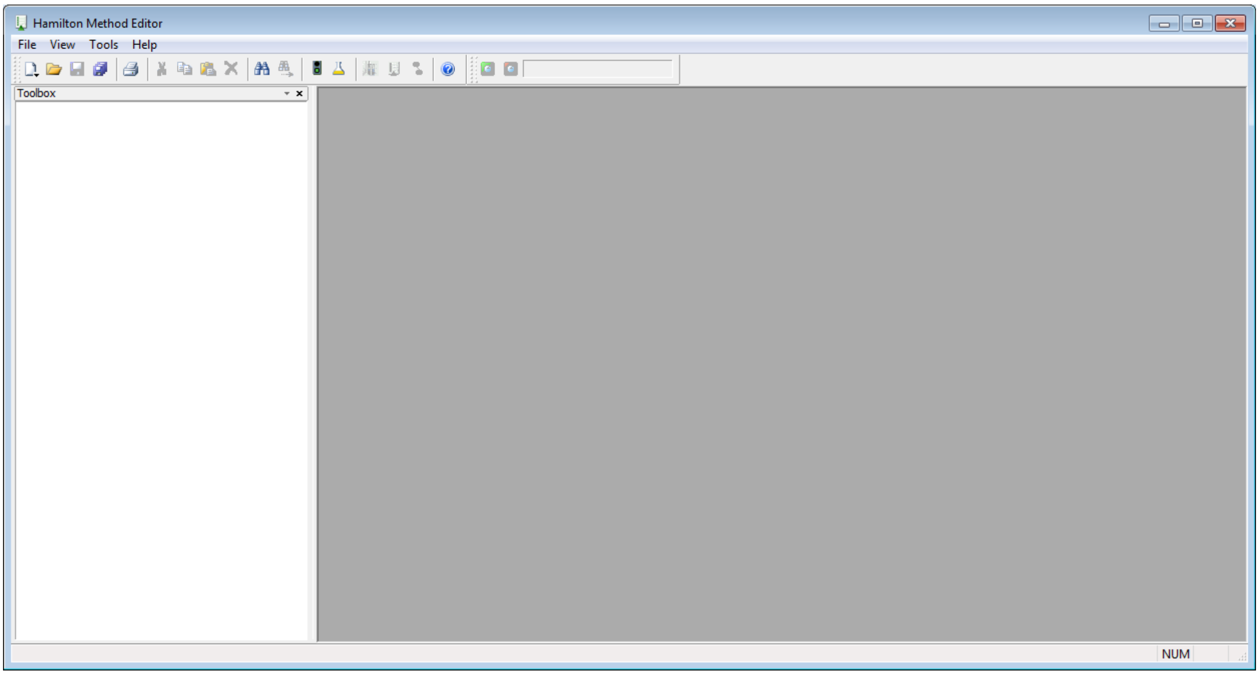
If a labware of a different or new kind must be defined, the "**Labware Editor**" helps in defining the new labware.

4.2.1 Introducing the Method Editor

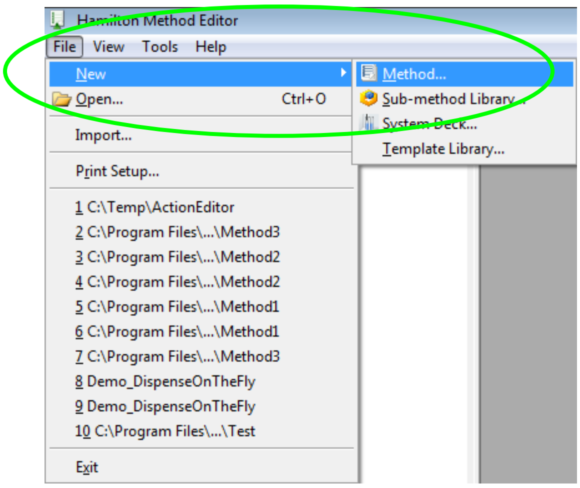
This section is a fast and easy introduction to the "**Method Editor**". This section discusses the primary simple steps up to the programming proper. Setting up of the application can be started without explaining the details of VENUS Software. More detailed and illustrated examples can be found in other sections of this manual.

Access the "**Method Editor**" from the System Tools section of the VENUS application.

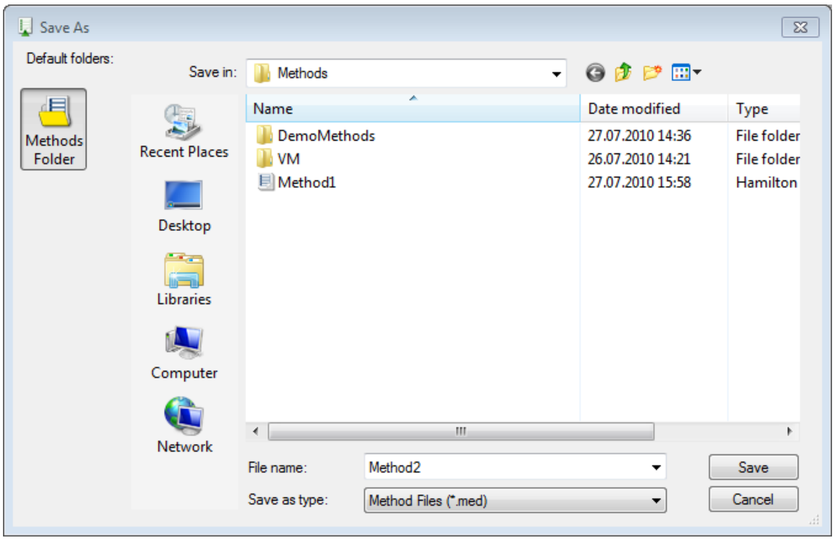
1. When started, the application appears as the image shown below:



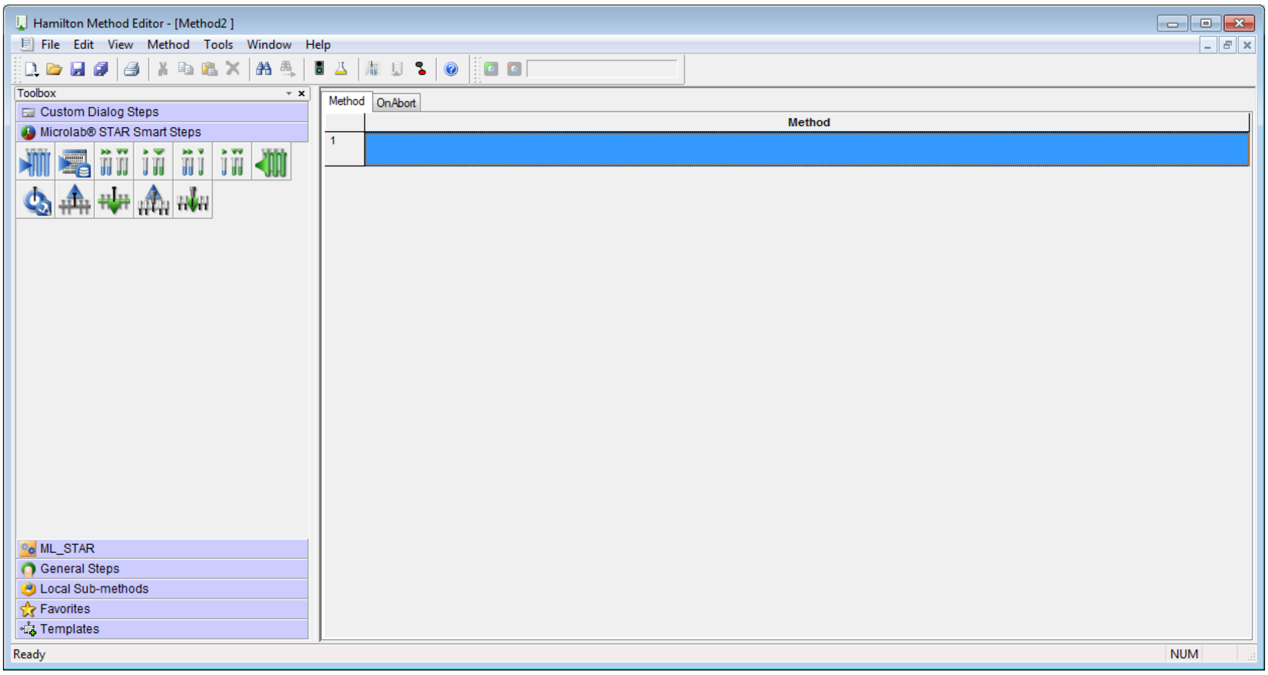
2. To create a method, click on the "  " Icon found on the toolbar or select "**File → New → Method**" as shown in the image below.



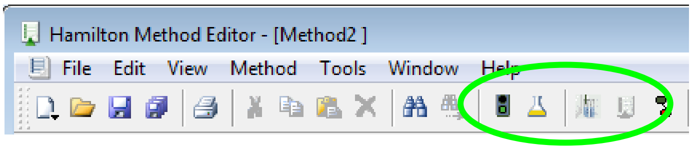
3. A selection dialog appears as shown in the image below.

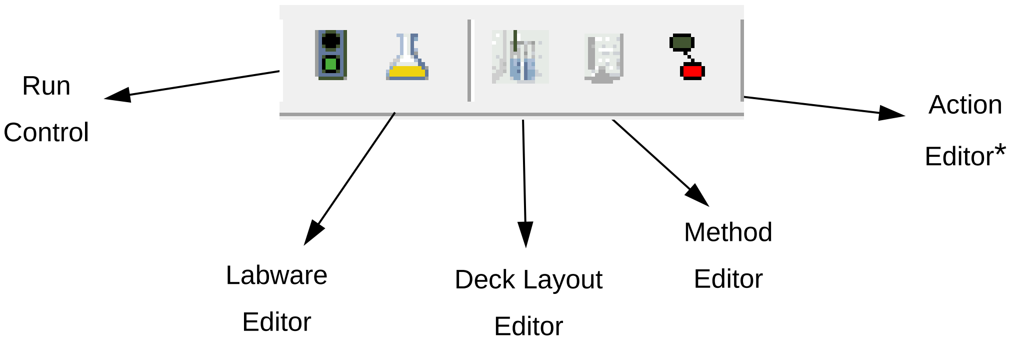


4. Select "**Methods**" and type a file name for the first method, then click the **[Save]** Button. The new method is saved, and the Step Editor window will be displayed:

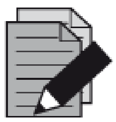


5. To switch to other views like the "**Deck Layout**" or the "**Action Editor**", use the quick link icons found on the toolbar.





*This icon leads to the action editor if the scheduling software is not installed. When the scheduling software is installed, this icon will lead to the activity editor.



NOTE

There are three predefined arrangements of views available.

*Select "**view → views →...**"*

or use

***["Alt" + 8]** for Easy view*

***["Alt" + 9]** for Standard view*

***["Alt" + 0]** for Full screen*

At this point, only an empty method and deck layout have been created. This is the base for the first method. Now, the newly created environment must be extended by the labware, sequences, pipetting steps, etc.

The empty deck must be filled with labware and carriers according to the real instrument deck. Adding labware and carriers onto the deck layout is easy (refer to Section 6.2 Adding Labware on the Deck). The deck layout contains all geometrical information the instrument requires for the pipetting and plate handling steps.

Once the real instrument deck has been mapped in the software deck layout, the method is then prepared for programming (refer to Section 7 How to Create a Method) for further details.

In the programming phase, the programmer will add steps to the step editor or add actions to the Action Editor view. Adding a step or adding an action will both prompt a user dialog to request the necessary values for the instrument.

5 Editor Details

5.1 Method Editor

Methods are programs that string together a number of specified commands (such as aspirate or dispense) to instruct the instrument to perform a function. They can be as complex as or as simple as the programmer desires. Access to a method editor is necessary to define or edit methods.

Action Editor

The action editor view is a very simplified way to program methods. In the toolbox of the action editor, all possible actions on an instrument can be found. “Drag-and-Drop” of an action will invoke a wizard to receive all necessary information from the user. At the end, a graphical view of a method is shown.

Graphical Method Editor

The graphical method editor (called method editor) is a syntax-free editor that presents a group-like graphical editing environment. Common constructions such as loops, conditionals and built-in functions are supported by this editor. It also provides a variable management system. This system simplifies variable usage by allowing implicit declarations and by supplying complete variable context to edit dialogs.

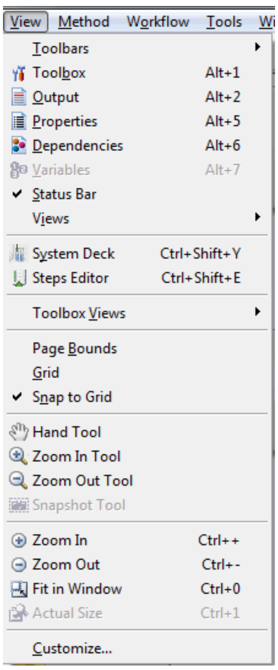
The output of the method editor is a linked set of files that may be executed by the *Run Control*.

In most cases (but not necessarily), a method refers to one single deck layout.

5.1.1 Views and Windows

The method editor is divided into multiple attachable windows (these are empty during start-up):

Changing the View



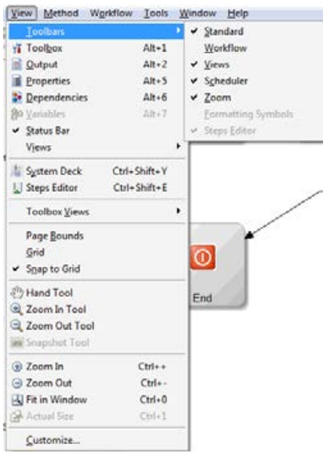
These windows can be activated by selecting the “**View**” Menu:

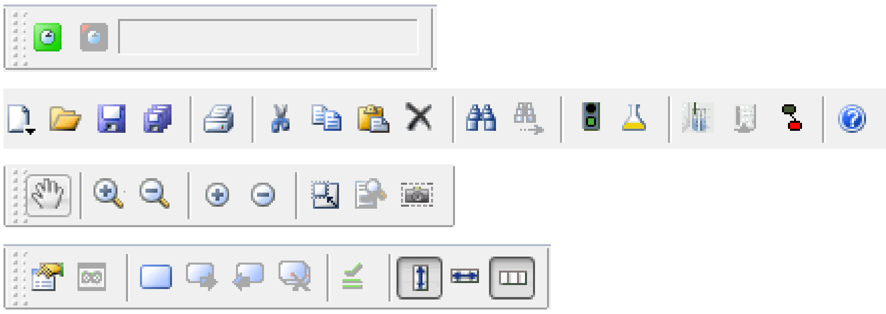
5.1.2 Toolbars

The toolbars contain some of the most frequently used commands to be executed directly by a mouse click. The toolbar can be docked or detached by a drag-and-drop.

The "**Toolbar**" Icons are shown on the left of the corresponding menu functions.

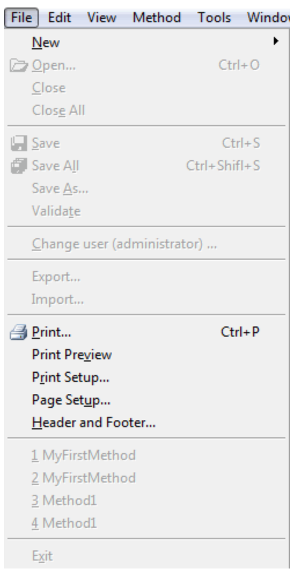
The toolbars shown in the images below can be activated by clicking the "**View → Toolbars**" Menu:





5.1.3 Menu Options

5.1.3.1 File Menu



The "**File**" Menu contains the sub-menu options for Windows™-controlled applications.

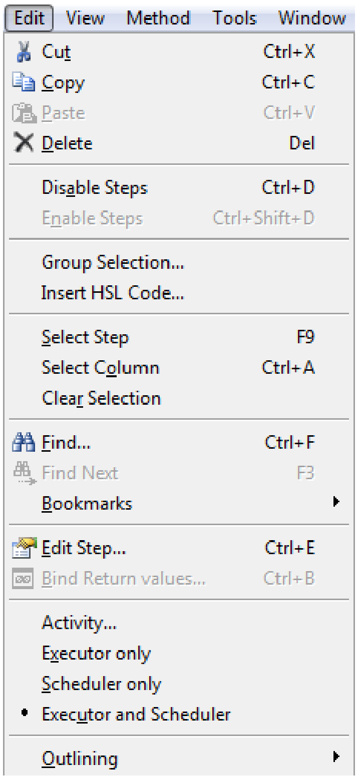
The "**File**" Sub-menu options "**Import**" and "**Export**" allow interchange of complete methods with all relevant files attached. For more information, refer to Section 5.1.4 Method Export and Import.

The "**File**" Menu options differ slightly, depending on whether the Method Window or the Deck Layout Window is activated.

Information about the "**Bind Return values…**" Sub-menu can be found in Section 11.1.1 Variables and Section 11.1.2 Return Values.

The "**Edit**" Menu is only available if the Method-Editing Window is activated. For the System Deck Layout Window, no "**Edit**" Menu exists.

5.1.3.2 Edit Menu



The "**Edit**" Menu also contains familiar features like "Copy" and "Paste".

The "**Select Step**" and "**Select Column**" and the corresponding shortcuts allow keyboard-driven selection of single steps or all steps.

To select more than one step, move to a step using the cursor up-down and/or page up-down keys and then click **[F2]**. This may be repeated as often as needed. After doing so, copy the selected steps to the clipboard.

A familiar way to copy and paste is pressing **[Ctrl-C]**, moving to the desired insert position and clicking **[Ctrl-V]**. The same mouse-driven operations can be performed by holding-down the **[Ctrl]** key and left-clicking on the desired steps to select them.

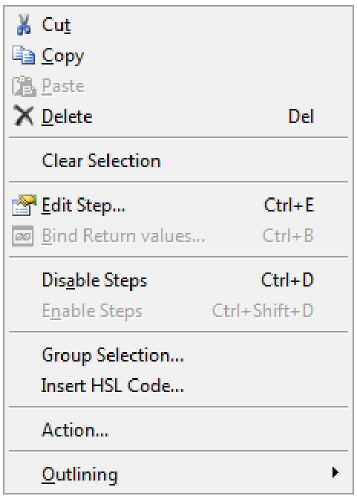
The "**Edit**" Menu differs depending on having the Scheduler installed or not: If not, the "**Action…**" Command can be selected. If yes, an "**Activity…**" Command will be added, with "**Executor only / Scheduler only / Executor and Scheduler**" Commands.

The "**Edit Step…**" Sub-menu opens the parameter dialog box of the selected step. The dialog will depend on which step is selected.

The "**Edit**" Menu is only available if the method-editing window is activated, for the System Deck Layout Window, no "**Edit**" Menu exists.

Information about the "**Bind Return values...**" Sub-menu can be found at the end of the Section 11.1.1 Variables and the Section 11.1.2 Return Values.

5.1.3.3 Context Menus



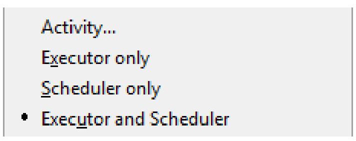
Method Editor Context Menu

Right-clicking on any selected step opens a pop-up menu where choices such as Cut, Copy or Delete can be made. Select the desired command.

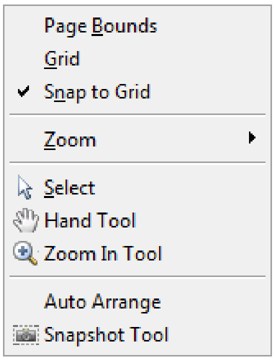
Note that with the Context Menu, steps can be deleted, edited, disabled, enabled and combined into groups.

It is also possible to insert HSL Code (Refer to Section 27.1 Glossary) into a method, for instance, to simplify a complex calculation. Be aware that new variables in an HSL Block are local and cannot be accessed from the method.

If the Scheduler is not installed, the "**Action…**" Command is visible. Refer to Section 12.2 Method for Sample Preparation Using the Action Editor to learn more.



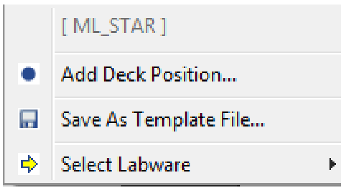
If the Scheduler is installed, the "**Activity...**" Menu, with "**Executor only / Scheduler only / Executor and Scheduler**" Commands is presented. For more information about the "**Activity…**" Command, please consult the Scheduler Manual.



Action Editor Context Menu

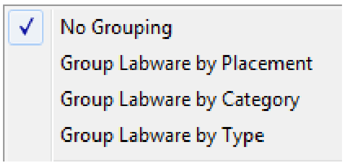
Right-clicking in the Action Editor opens the popup menu. The menu options "**Page Bounds**", "**Grid**" and "**Snap to Grid**" will simplify the positioning of the Action graphics within the window.

- The "**Zoom**" offers 5 different zoom stages (25% - 200%) and a "Fit in Window" option to zoom out for a full Action Editor view.
- The "**Select**", "**Hand Tool**" and "**Zoom In Tool**" controls the usage of the mouse pointer.
- "**Auto Arrange**" helps to align the action boxes.
- The "**Snapshot Tool**" is used to copy parts of the display to the clipboard.



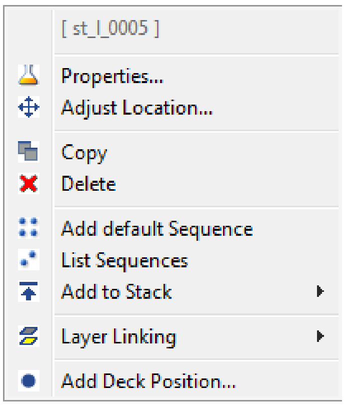
System Deck Editor Context Menu

Right-clicking on the Deck Layout opens the pop-up menu. It makes the most frequently used Deck Layout Commands available.



Select Labware

Labware display can be grouped by placement, category or type in order to make the selection simpler.



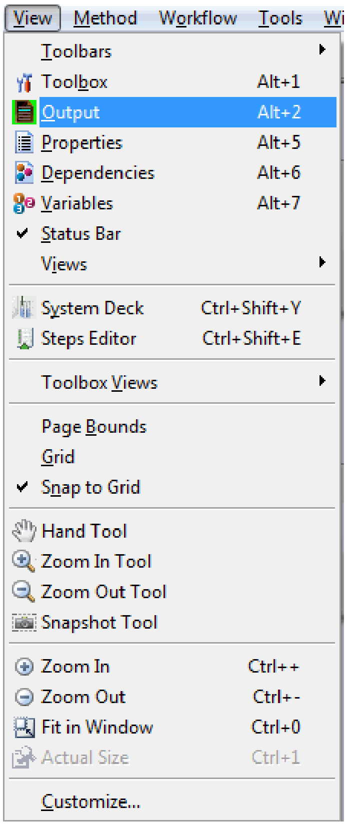
System Deck Editor Context Menu with Selected Labware

Right-clicking on the selected labware triggers the System Deck Editor Context Menu.

- "**Properties**" allows changes to the labware ID.
- "**Adjust Location**" is used to make free-form placements of labware.
- "**Copy**" will create another instance of the labware.
- Use "**Delete**" to remove the selected labware.
- Select "**Add default Sequence**" if the labware does not have a default sequence yet.
- "**List Sequences**" will display a list of all sequences related to the selected labware.
- Use "**Add to Stack**" to add multiple plates of the same type on top of the selected base plate.
- "**Layer Linking**" allows tagging a label on the labware. This label then controls the visibility by layer.
- "**Add Deck Position**" is used to create a position on the deck.

5.1.3.4 View Menus

The "**View**" Menu options appear differently in the Graphic Method, Deck Layout and Action Editors.

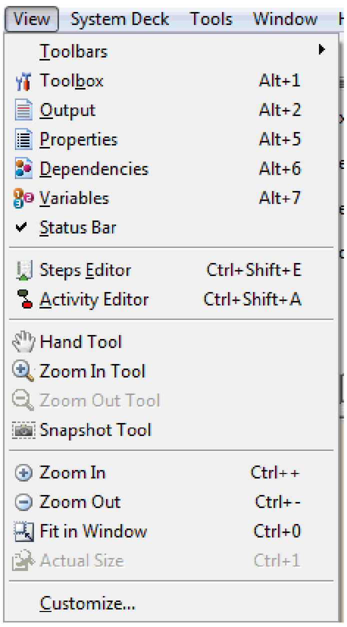


View Menu of the Graphic Method Editor

The "**View**" Menu can change the look of the Graphic Method Editor. The windows can be switched on or off. It can also determine the view of the steps in the method window as well as their appearance.

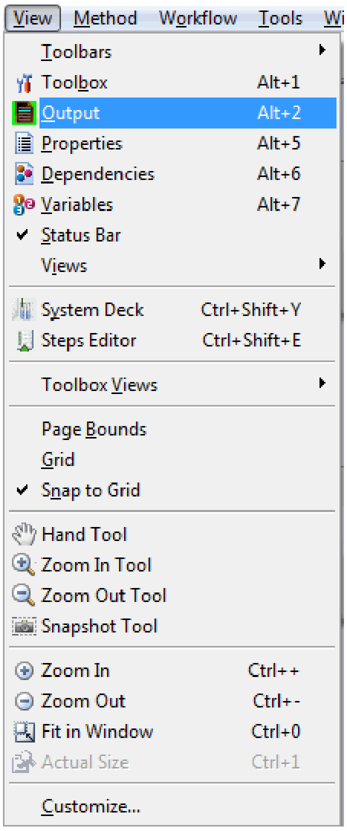
It is also possible to switch to the System Deck or to the Action Editor.

"**Customize…**" offers several setting options regarding step visibility, step colors, toolbox group orders, toolbars and the appearance of commands.



View Menu of the Deck Layout Editor

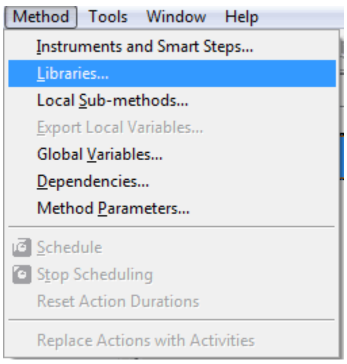
The "**View**" Menu of the Deck Layout Editor differs slightly from the one in the method editor. It contains zooming options for the deck view.



View Menu of the Action Editor

The "**View**" Menu of the Action Editor differs slightly from the View Menu in the Method Editor. It contains zooming options for the action view and also allows setting grids, paging bounds, etc.

5.1.3.5 Method Menu



Method Menu

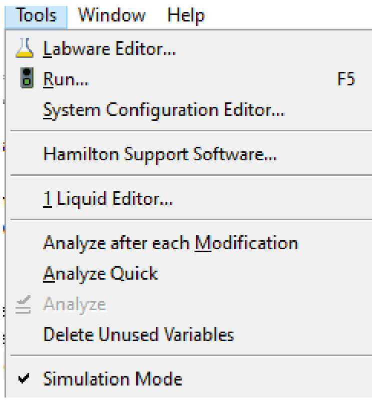
The "**Method**" Menu contains functions relevant to method editing, such as:

- Linking Instruments and Deck Layouts to a Method
- Including Libraries
- Handling Local Sub-Methods
- Exporting Local Variables
- Adding Files to a Method's Package for Exporting
- Setting Method Parameters

Several run options become available when the Scheduler is installed.

To upgrade methods written in the Action Editor to the scheduling environment, replace the actions with activities.

5.1.3.6 Tools Menu



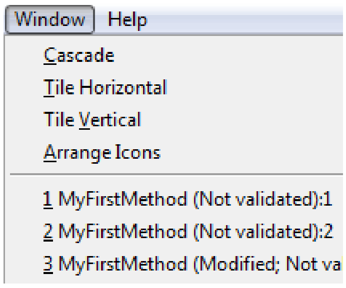
Tools Menu

The "**Tools**" Menu contains functions to call the different editors from within the Graphic Method Editor.

- "**Labware Editor**" is used to modify or create new labware
- "**Run**" starts up the run control editor
- "**System Configuration Editor**" is used to modify settings. For example, specifying simulation settings
- "**Hamilton Support Software**" installs additional labware and libraries
- "**Liquid Editor**" is used to create and modify liquid classes
- **"Simulation Mode"** toggles the simulation mode on/off

The lowest division of the menu controls the behavior of the method analyzer.

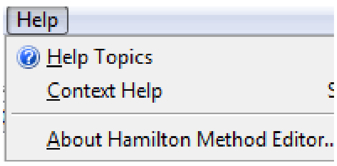
5.1.3.7 Window Menu



Window Menu

The "**Window**" Menu allows the option of arranging the windows in different ways.

5.1.3.8 Help Menu



Help Menu

The "**Help**" Menu offers additional information about the Hamilton VENUS Software through the help topics and the context Help.

"**About Hamilton Method Editor**" will display the currently installed version of the software.

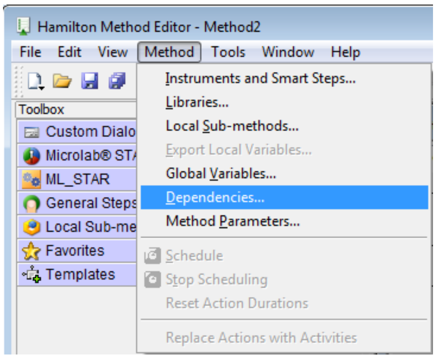
5.1.4 Method Export and Import

Exporting and importing methods allow interchanging of methods between different PC's. Once a method has been created and tested, it can then be packed in a single file with all the relevant information which includes the following:

- Method
- System Deck
- Required Labware
- Required Liquid Classes
- Included Libraries (standard and self-created)

It is also possible to add files to the package that are needed to execute the method but are not part of the Hamilton Software.

Example: an Excel sheet that is read to define the sample volumes.



To do so, select **Method → Dependencies** and add all files that will be part of the export package.

The advantage of such an export-import transfer is that not all labware, libraries, etc. have to be installed by default on another PC. Such transfer of files ensures that all information will be stored on the target PC when the package is imported.

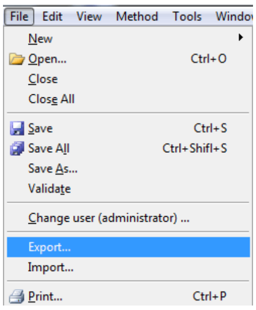
Exporting methods is also a useful procedure for backing up the methods.

NOTE

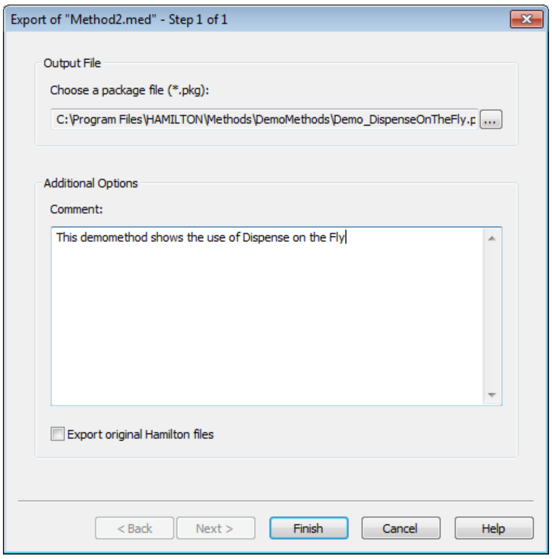
All data will be compressed, so a complete method can be transferred easily even through e-mail.

5.1.4.1 Exporting a Method

To collect all information used by a method, and to create one package file, follow the steps shown below.

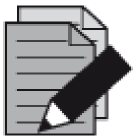


Choose “**File → Export…**” within the method to be transferred. A dialog box will open, requesting for a filename.



Click the **[…]** Button to select the package name including the path desired to export.

Click **[Finish]** to complete the operation.



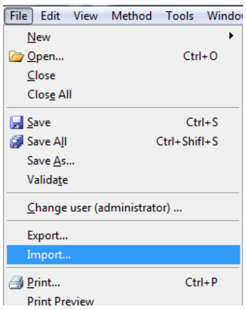
NOTE

If a method has never been run on an instrument – neither on the simulator nor on a real instrument – not all data may be collected. It is recommended to run a simulation before creating a package for exporting.

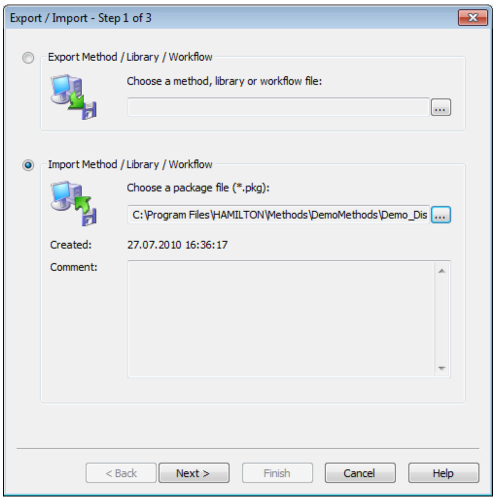
5.1.4.2 Importing a Method

The method in a package must be imported once received. To do this, perform the following steps:

1. Start the "Graphical Method Editor".

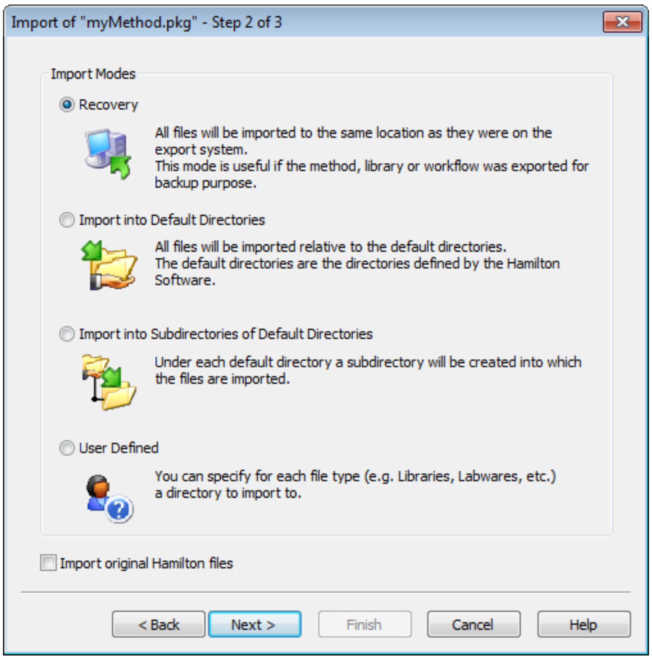


2. Select among the "**File → Import…**" Menu



3. Choose the desired package by clicking on **[…]**
4. Once a package is selected, the comment written during the export process will be seen along with the name or path of the package.
5. If the selected package is not the desired one, click **[…]** again to select another one.
6. Click **[Next >]** to determine the location in which the package will be unpacked.

There are four different possibilities:



Recovery: A useful way to restore a destroyed method. If the method was exported for backup purposes, re-importing it while selecting the recovery option should be used. The same directory structure will be used to extract and store all the necessary files that were collected during the export procedure.

Import into Default Directories: Useful when data should be stored in the default directories as defined in VENUS Software. An example is by using the labware directory to save labware files.

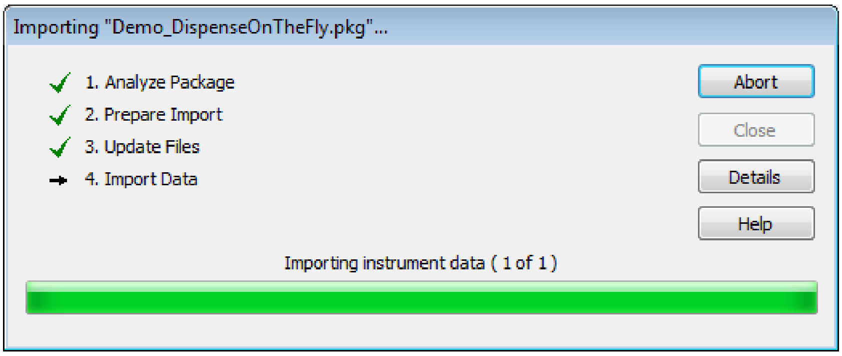
Import into Sub-Directories of Default Directories: Similar to the second possibility, however using this option will create sub-directories to the default directories with the names specified. This option guarantees that the existing files having the same names with the new files will not be overwritten.

User-Defined: Provides the most flexibility in selecting the target directories. The next step will ask for a definition of the target directory for all groups of files, method files, library files, labware files and other files.

Import Original Hamilton Files: If this option is enabled, original Hamilton files are imported into a target environment; otherwise, these files are skipped. This could however lead to non-working methods.

7. Click **[Next >]** again, to get to a summary of the current directories and, if necessary, further input fields and selection possibilities to change the desired directories. The summary will be updated on an ongoing basis with any changes that have been made.
8. Once the selections correspond to the requirements click **[Finish].** This will unpack and store all of the files.

9. The system imports all necessary files and prompts the dialog below upon completion. Click **[Close]** to exit.

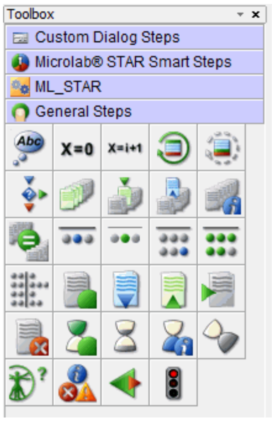


5.1.5 Toolbox Window

5.1.5.1 Different Views of the Toolbox Groups

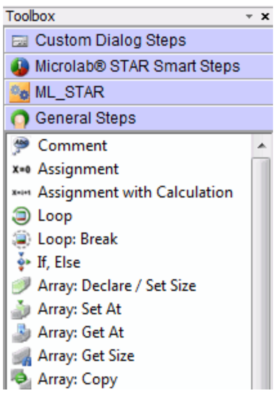
The Toolbox Window can be displayed in several views.

The different view styles are shown in this section, using the "**General Steps**" group which is always available -- even in an empty method, as an example. In the further discussion of groups, the large icons will be listed. The group view may be in any of the three possible styles below.



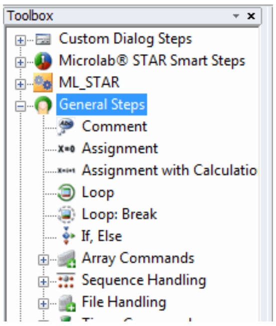
Icon View

This is the default style. A group (library) added will first be displayed in this style. If the mouse is hovered over an icon, a "**Tool Tip**" will appear with the function's name.



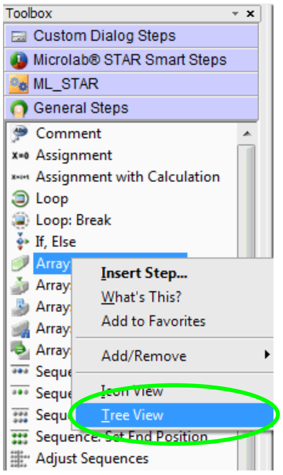
List View

This style displays all functions, each with short descriptions. The sub-group names are also listed; however, this requires more space to display the entire list.



Tree View

This style may be more comfortable to work with. The sub-groups are displayed, and can be opened, as shown, in the case of the "**Array Commands**" sub-group, and closed, as shown, in the case of the "**File Handling**" sub-group. Open or close sub-groups by clicking the “+” or the “-“ sign found on the left side of the sub-group name.

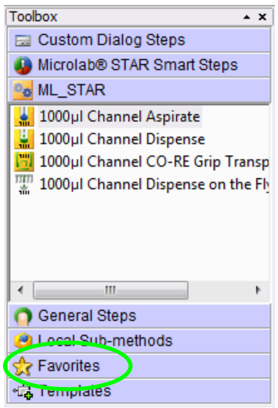


Select View

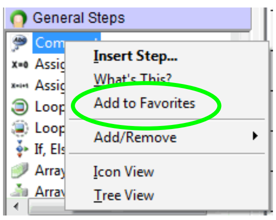
To change the style of the toolbox, right-click on the toolbox window and a menu will appear. Select the preferred view style from the menu.

5.1.5.2 Favorites

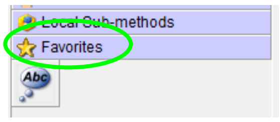
The Favorites Bar is available to have quick access to the most used commands.



Every desired step can be added to the Favorites group.



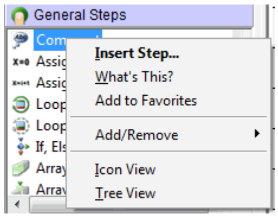
To do so, select the step to be added. Enter the Context Menu by right-clicking and selecting “**Add to Favorites**”.



This step is now part of the Favorites group.

If a step is a member of the Favorites group, the related library will automatically be installed as soon as the step is used in the method.

The favorites group is stored locally on the computer.



In this menu, steps can be deleted, moved up, moved down and can be sorted alphabetically.

5.2 Command Overview by Step Groups

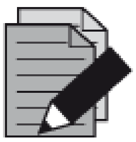
The VENUS Software offers various standard **Command Libraries**. For the Microlab instrument, the software offers three levels of commands which are:

Power Steps, which simplify programming of common tasks such as Transfer Samples, Add Reagent, Serial Dilution, Replicates and Hit Picking.

Smart Steps, which combine a complete pipetting task such as plate copy, serial dilution, pooling etc. Smart Steps incorporate customizable error handling which includes a guided programming wizard, predefined error recoveries and customized recovery strategies.

Easy Steps, which are combinations of e.g., tip pickup and aspiration. Easy steps allow for very flexible programming and include full error handling. However, the sequence support is limited (e.g., for pipetting into all wells of the entire plate, a loop must be added).

Single Step commands are used when the highest flexibility of the system is required. These commands allow even the most complex pipetting and plate handling tasks. However, tip handling and sequence support are not included.

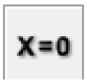
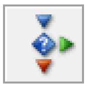
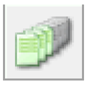
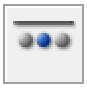
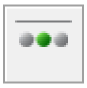
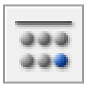
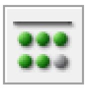
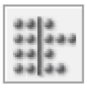


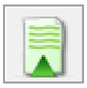
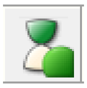
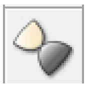
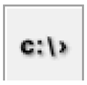
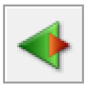
NOTE

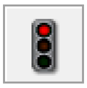
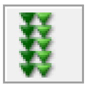
After a new installation a reduced selection of steps are available. To make hidden steps available, enable them via the Configuration Editor. Refer to *Section 3.4 Step Selection* *and / or* *Section 3.5 Microlab Instrument Settings*.

5.2.1 General Steps

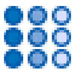
A detailed description of all commands belonging to the "**General Steps**" Group in the method editor toolbox window is directly accessible in the software. For more information, refer to the Help Menu.

General Steps		
Command	**Icon**	**Action Performed**
Comment		For any programmer-defined comments in a method
Assignment		Initialize a variable with a value
Assignment with Calculation		Perform basic arithmetical operation
Loop		Loop commands for start and end of loop will be inserted
Loop Break		Unconditional loop termination
If, Else		Conditional command
Array: Declare / Set Size		Define the type and the size of the array (1-based)
Array: Set at		An element within an array is set at a given index (1-based)
Array: Get at		Get the value of an array-element at a given index (1-based)
Array: Get Size		Get the number of elements (1-based)
Array: Copy		Copy all elements of an array to another (1-based)
Sequence: Get current position		Get current position (the next available position in a sequence)
Sequence: Set current position		Set current position
Sequence: Get end position		Get end position
Sequence: Set end position		Set end position
Sequence: Adjust sequences		Adjust sequences (Adjust all given sequences to the length of the shortest one)

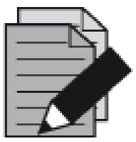
General Steps		
Command	**Icon**	**Action Performed**
File: Open		Open a file before file access
File: Read		Read data out of a file
File: Write		Write data into a file
File: Set position		Set the file-pointer to a specific position
File: Close		Close a file after file operation
Timer: Start timer		Start a timer
Timer: Wait for		Wait for timer (timeout)
Timer: Read elapsed time		Read elapsed time from a started timer
Timer: Restart		Restart a timer
Dialog: User input		Display an input box with one or several lines
Dialog: User output		Display an output box
Communication: Open		Open a COM port
Communication: Read		Read data from a COM port
Communication: Write		Write data to a COM port
Communication: Close		Close an open COM port
Shell		Call external program out of the Hamilton software
Set Event		Create an event to take place in a parallel process
Wait for Event		Wait until an event in a parallel process has taken place
Return		Assign a return value to a sub-method and return

General Steps		
Command	**Icon**	**Action Performed**
Abort		Abort method
Error Handling by the User		Identify a section with programmer-defined error handling
Begin parallel process		Start a bifurcation for programming parallel processes
End parallel process		End a bifurcation for programming parallel processes
[1] In contrast with HSL arrays, which are 0-based. For the HSL Method Editor, see Chapter 19.		

5.2.2 Power Steps

Power Steps		
Icon	**Name / Function**	**Section**
	**Transfer Samples** Sets up a single transfer of a given set of sources to a set of targets.	13.7.2
	**Add Reagent** Sets up the distribution of a reagent to a set of targets over one or multiple pieces of labware.	13.7.3
	**Serial Dilution** Sets up the distribution of a reagent, samples addition and serial dilution of a set of samples. Can be performed in tubes, plate rows, or plate columns.	13.7.4
	**Replicates** Simplifies the process of creating copies of a set of samples or a full pattern of a plate.	13.7.5
	**Hit Picking** Is very similar to the Transfer Samples step, with the option of getting the sources, destinations, and/or transfer volume information from a file (worklist).	13.7.6
	**Load and Match** Loads and scan selected sample tube carriers and finds on the deck barcode matches from a given worklist	13.7.7

5.2.3 Smart Steps



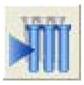
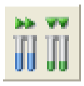
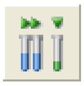
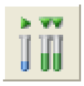
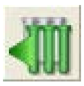
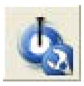
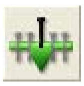
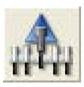
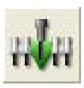
NOTE

Once a deck layout has been linked to a method, the instrument-specific steps will be visible in the toolbox.

Smart Steps are powerful commands for programming the Microlab instrument.

Smart Steps combine many single steps for specific tasks like filling a microplate starting with tubes, aliquoting a reagent to a complete plate, loading the deck, etc. The available Smart Steps are listed in the following table.

To use instrument-specific steps (load, pipette, etc.) the method must be first linked to an existing deck layout first (see Section 7.1 New Method) while keeping the “**Smart Steps**” Checkbox selected.

Smart Steps		
Icon	**Name / Function**	**Section**
	**Advanced Load setting** Define other settings than the default of the load steps	14
	**Load** Load carrier on deck	14
	**Load and Match** Load samples and match loading information with worklist information, to generate data for use in the following process steps. Data must be present in Hamilton database, e.g. by using Import Worklist from data handling steps	14
	**1000 µL Channel Pipette - Simple (1-1)** Simple pipetting for simple aspiration/dispense cycle	13
	**1000 µL Channel Pipette - Replica (1-n)** Pipetting for cycles of aspirations/dispenses where the liquid from one source is dispensed into multiple target containers (no aliquoting)	13
	**1000 µL Channel Pipette - Pooling (n-1)** Pipetting for cycles of multiple aspirations/dispenses where liquid from multiple source containers is dispensed into one target container	13
	**1000 µL Channel Pipette - Aliquot** Pipetting for cycles with one aspiration followed by multiple dispenses.	13
	**Unload** Remove carrier from deck	13
	**1000 µL Channel Needle Wash Settings** Specify parameters for needle wash	19
	**1000 µL Channel Needle Pick up** Pick up needles from wash station (or Racks)	13 19
	**1000 µL Channel Needle Eject** Release needles in racks or wash station (and start wash)	13 19
	**1000 µL Channel Tip Pick up** Pick up disposable tips from tip rack	13
	**1000 µL Channel Tip Eject** Eject disposable tips into tip waste	13

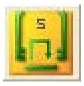
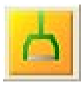
The Method Editor allows the free combination of Smart Steps, Easy Steps and Single Steps.

For all Smart Steps, the instrument name is to be selected. This is set by default to the Microlab STAR short name and cannot be changed within the Smart Steps.

A selection is necessary only in the case of two or more instruments linked together.

5.2.4 Easy Steps

The EASY Steps belong to the "**Microlab STAR**" Group found in the method editor toolbox window. These commands are indicated by orange-colored backgrounds of the icons.

Easy Steps		
Icon	**Name / Function**	**Section**
	**1000 µL Channel Aspirate** Aspirate liquid with the pipetting channels	12.4 13.5.1
	**1000 µL Channel Dispense** Dispense with the pipetting channels	12.4 13.5.1
	**5 mL Channel Aspirate** Aspirate liquid with the pipetting channels	13.5.1
	**5 mL Channel Dispense** Dispense with the pipetting channels	13.5.1
	**CO-RE 96 Head Aspirate** Aspirate liquid with the CO-RE 96 Probe Head	13.5.1
	**CO-RE 96 Head Dispense** Dispense liquid with the CO-RE 96 Probe Head	13.5.1
	**CO-RE 384 Head Aspirate** Aspirate liquid with the CO-RE 384 Probe Head	13.5.1
	**CO-RE 384 Head Dispense** Dispense liquid with the CO-RE 384 Probe Head	13.5.1
	**1000 µL Channel CO-RE Grip Transport** Transport a plate with CO-RE Grip	16.1.1
	**5 mL Channel CO-RE Grip Transport** Transport a plate with CO-RE Grip	16.1.1
	**iSWAP Transport** Transport a plate with iSWAP	15.1.10

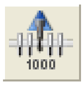
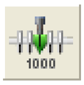
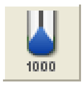
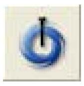
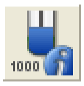
5.2.5 Single Steps

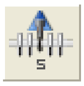
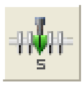
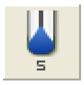
Single steps are restricted to single actions like picking up tips, aspirate, dispense, etc.

To make them available, enable them through the System Configuration Editor. Refer to Section 3.1 System Settings.

The Single Steps belong to the "**Microlab STAR**" Group found in the Method Editor Toolbox window with the suffix "**Single Step**".

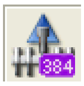
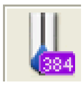
The tables found on the next pages supply brief overviews of the available instrument-specific commands. These commands are used in both the graphical and the HSL Method Editors. In the tree view of the "**Microlab STAR**" Group, the commands are structured in sub-groups (Liquid handling, Preparation, Transport, Miscellaneous), some in further sub-groups (Channels, CO-RE 96 Probe Head, CO-RE 384 Probe Head, Nano Pipettor, iSWAP, CO-RE Grip).

Liquid Handling: 1000 µL-Pipetting Channels		
Icon	**Name / Function**	**Section**
	**1000 µL Channel Tip Pick up (Single Step)** Pick up tips/needles from a Tip Carrier/Wash Station	13.2.1 13.5.3
	**1000 µL Channel Tip Eject (Single Step)** Eject tips into the waste or place the needles into the Wash Station	13.2.4 13.5.3
	**1000 µL Channel Aspirate (Single Step)** Used to aspirate liquid with the 1000 µL-pipeting channels	13.2.2 13.5.3
	**1000 µL Channel Aspirate 2nd Phase (Single Step)** This step is applicable for liquids separated in two phases (e.g. centrifuged samples), where the lower part of the liquid is well separated from the upper part of the liquid and has a recognizable higher viscosity.	
	**1000 µL Channel Dispense (Single Step)** Used to dispense liquid with the 1000 µL-pipetting channel	13.2.3 13.5.3
	**1000 µL Channel Dispense on the Fly (Single Step)** Used to dispense liquid in jet mode without stopping x-direction, using the 1000 µL-pipetting channel	13.5.3
	**1000 µL Channel Start Needle Wash (Single Step)** Starts a needle wash in a Wash Module	19.2.1
	**1000 µL Channel Wait for Needle Wash (Single Step)** Waits for the needle Wash Module to be ready	19.2.1
	**1000 µL Channel Get Last Liquid Level (Single Step)** Returns the liquid level height detected during previous aspirate or dispense step with enabled Liquid Level Detection	13.5.3

Liquid Handling, 5 mL-Pipetting Channels		
Icon	**Name / Function**	**Section**
	**5 mL Channel Tip Pick up (Single Step)** Pick up tips/needles from a Tip Carrier/Wash Station	13.2.1 13.5.3
	**5 mL Channel Tip Eject (Single Step)** Throw off tips into waste or into a specific sequence (e.g. empty Tip Rack) or ejects the needle into the Wash Station	13.2.4 13.5.3
	**5 mL Channel Aspirate (Single Step)** Used to aspirate liquid with the 5 mL-pipetting channel	13.2.2 13.5.3
	**5 mL Channel Dispense (Single Step)** Used to dispense liquid with the 5 mL-pipetting channel	13.2.3 13.5.3
	**5 mL Channel Dispense on the Fly (Single Step)** Used to dispense liquid in Jet Mode without stopping x-direction, using the 5 mL-pipetting channel	13.5.3

Liquid Handling, 5 mL-Pipetting Channels		
Icon	**Name / Function**	**Section**
	**5 mL Channel Get Last Liquid Level (Single Step)** Returns the liquid level height detected during previous aspirate or dispense step with enabled Liquid Level Detection	13.5.3

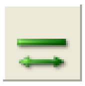
Liquid Handling, CO-RE 96 Probe Head		
Icon	**Name / Function**	**Section**
	**CO-RE 96 Head Tip Pick Up (Single Step)** Pick up CO-RE tips	13.2.1 13.5.3
	**CO-RE 96 Head Tip Eject (Single Step)** Discards the tips to the waste or into a defined sequence (e.g. an empty Tip Rack)	13.2.4 13.5.3
	**CO-RE 96 Head Aspirate (Single Step)** Aspirates liquid from a container using the CO-RE 96 Head	13.2.2 13.5.3
	**CO-RE 96 Head Dispense (Single Step)** Dispenses liquid into a container using the CO-RE 96 Head	13.2.3 13.5.3
	**CO-RE 96 Head Wash (Single Step)** Washes tips in an according wash position, moving the plunger up and down	21.1 22.1
	**CO-RE 96 Head Empty Washer (Single Step)** This step drains all chambers of the CO-RE 96 Head Washer	21.1 22.1

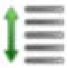
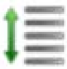
Liquid Handling, CO-RE 384 Probe Head		
Icon	**Name / Function**	**Section**
	**CO-RE 384 Head Tip Pick Up (Single Step)** Pick up CO RE tips in full / row-wise / column-wise or single mode	13.2.1 13.5.3
	**CO-RE 384 Head Tip Eject (Single Step)** Discards the tips	13.2.4 13.5.3
	**CO-RE 384 Head Aspirate (Single Step)** Draws liquid from containers	13.2.2 13.5.3
	**CO-RE 384 Head Dispense (Single Step)** Drops liquid into containers	13.2.3 13.5.3
	**CO-RE 384 Head Wash (Single Step)** Washes the tips	22.1
	**CO-RE 384 Head Empty Washer (Single Step)** This step drains all chambers of the CO-RE 384 Head Washer	22.1

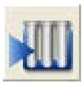
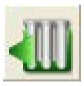
Liquid Handling, Nano Pipettor		
Icon	**Name / Function**	**Section**
	Nano Pipettor Aspirate (Single Step) Aspirates liquid	13.5.3
	Nano Pipettor Dispense (Single Step) Dispenses liquid	13.5.3
	Nano Pipettor Dispense On The Fly (Single Step) Dispenses liquid over an entire sequence while x-Arm is moving	13.5.3
	Nano Pipettor Wash (Single Step) Cleans tips by flushing system liquid through it and by ultrasonic treatment	13.5.3

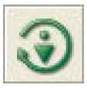
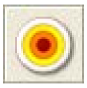
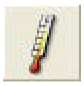
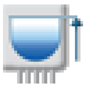
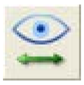
Liquid Handling: MagPip Channels (STAR^{V} and and VANTAGE only)		
Icon	**Name / Function**	**Section**
	MagPip Channel Tip Pick up (Single Step) Pick up tips/needles from a Tip Carrier/Wash Station	
	MagPip Channel Tip Eject (Single Step) Eject tips into the waste/Tip Carrier/Wash station.	
	MagPip Channel Aspirate (Single Step) Used to aspirate liquid with the MagPip -pipeting channels	
	MagPip Channel Dispense (Single Step) Used to dispense liquid with the MagPip -pipetting channel	
	MagPip Channel Dispense on the Fly (Single Step) Used to dispense liquid in jet mode without stopping x-direction, using the MagPip -pipetting channel	
	MagPip Channel Get Last Liquid Level (Single Step) Returns the liquid level height detected during previous aspirate or dispense step with enabled Liquid Level Detection	

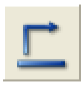
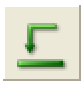
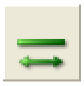
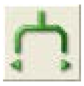
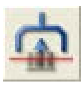
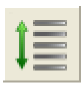
Track Gripper (VANTAGE only)		
Icon	**Name / Function**	**Section**
	Initialize (Single Step) Initializes the Track Gripper arm	
	Get Plate (Single Step) Eject tips into the waste/Tip Carrier/Wash station.	
	Place Plate (Single Step) Used to aspirate liquid with the MagPip -pipeting channels	

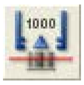
Track Gripper (VANTAGE only)		
Icon	**Name / Function**	**Section**
	**Move to Position (Single Step)** Used to dispense liquid with the MagPip -pipetting channel	
	**Lock/Unlock Doors (Single Step)** Controls the door locks of the logistics cabinet	

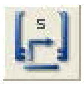
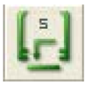
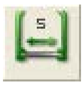
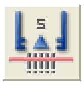
Entry Exit (VANTAGE only)		
Icon	**Name / Function**	**Section**
	**Initialize (Single Step)** Initializes the Entry Exit modules	
	**Move to Beam (Single Step)** Moves Entry Exit stacks to the beam	
	**Prepare Stack for Labware Removal (Single Step)** Raises the Entry Exit stacks to a gripper accessible height	
	**Prepare Stack for Labware Addition (Single Step)** Lowers the Entry Exit stacks to receive new labware	
	**Count Labware in Stack (Single Step)** Counts the pieces of labware loaded in a stack	
	**Inventory Stack (Single Step)** Scans the barcodes of a stack	
	**Random (Shelf) Access (Single Step)** Access a specific shelf of a random access magazine	
	**Move Random Access (Single Step)** Move a random access magazine to the top beam or the bottom sensor	
	**Lock/Unlock Doors (Single Step)** Controls the door locks of the logistics cabinet	

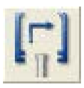
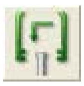
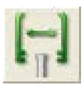
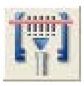
Preparation		
Icon	**Name / Function**	**Section**
	**Initialize (Single Step)** Initializes the instrument	14.3.1
	**Load Carrier (Single Step)** Loads a Carrier on the deck. If autoload is available, the Carrier will be moved inside automatically. Otherwise, a user dialog is shown.	14.3.1
	**Unload Carrier (Single Step)** Removes a Carrier from the deck. If autoload is available, the Carrier will be moved outside automatically. Otherwise, a user dialog is shown.	14.3.1

Preparation		
Icon	**Name / Function**	**Section**
	**Reload carrier (Single Step)** Load, unload or reload a Carrier	14.3.1
	**Calibrate (Single Step)** Measures precise position of a high-density Microplate (1536 wells) before aspiration/dispense.	14.3.1
	**Set Carrier Temperature (Single Step)** Sets the temperature of a TCC (Temperature-Controlled Carrier)	23.1
	**Get Carrier Temperature (Single Step)** Retrieves the temperature of a TCC	23.1
	**Lock/Unlock Front Cover (Single Step)** Locks the front cover	14.3.1
	**Nano Pipettor Prepare (Single Step)** Primes liquid system and / or pipettor and flushes pipettor	13.5.3
	**Camera Channel Move (Single Step)** Moves the camera channel to a desired position	
	**S-Tube Cap (Single Step)** This step allows capping of S-tubes.	
	**S-Tube Decap (Single Step)** This step allows decapping of S-tubes.	

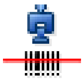
Transport: iSWAP		
Icon	**Name / Function**	**Section**
	**iSWAP Get Plate (Single Step)** Grip and lift a plate	15.1.11
	**iSWAP Place Plate (Single Step)** Set a plate down on a Carrier	15.1.11
	**iSWAP Move Plate (Single Step)** Move a plate from one Carrier across the deck to another Carrier	15.1.11
	**iSWAP Open Gripper (Single Step)** Spread the iSWAP Gripper fingers	15.1.11
	**iSWAP Close Gripper (Single Step)** Tighten the iSWAP Gripper fingers to grip a labware object	15.1.11
	**iSWAP Read Plate Barcode (Single Step)** Read the barcode affixed to a plate when Autoload is present	15.1.11
	**iSWAP Get First Plate Position (Single Step)** Stacker is checked for the first plate position	15.1.11
	**iSWAP Park (Single Step)** Move the iSWAP device to a parking position	15.1.11

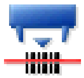
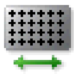
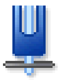
Transport: CO-RE Gripper with 1000µl-Pipetting Channels		
Icon	**Name / Function**	**Section**
	**1000 µLChannel CO-RE Grip Get Plate (Single Step)** Grip and lift a plate	16.1.2
	**1000 µL Channel CO-RE Grip Place Plate (Single Step)** Set a plate down on a Carrier	16.1.2
	**1000 µL Channel CO-RE Grip Move Plate (Single Step)** Move a plate from one Carrier across the deck to another Carrier	16.1.2
	**1000 µL Channel CO-RE Grip Read Plate Barcode (Single Step)** Read the barcode affixed to a plate when Autoload is present	16.1.2

Transport: CO-RE Gripper with 5 mL-Pipetting Channels		
Icon	**Name / Function**	**Section**
	**5 mL Channel CO-RE Grip Get Plate (Single Step)** Grip and lift a plate	16.1.2
	**5 mL Channel CO-RE Grip Place Plate (Single Step)** Set a plate down on a Carrier	16.1.2
	**5 mL Channel CO-RE Grip Move Plate (Single Step)** Move a plate from one Carrier across the deck to another Carrier	16.1.2
	**5 mL Channel CO-RE Grip Read Plate Barcode (Single Step)** Read the barcode affixed to a plate when Autoload is present	16.1.2

Transport: Tube-Gripper		
Icon	**Name / Function**	**Section**
	**Tube Grip Get (Single Step)** Grip and lift a tube	16.1.2
	**Tube Grip Place (Single Step)** Place a lifted tube	16.1.2
	**Tube Grip Move (Single Step)** Move a tube to a position without opening Grippers	16.1.2
	**Tube Grip Read Barcode (Single Step)** Move a tube in front of the Autoload Barcode Scanner and read	16.1.2

Transport: Gel Card Gripper		
Icon	**Name / Function**	**Section**
	**Gel Card Grip Get (Single Step)** Grip and lift a gel card	

Transport: Gel Card Gripper		
Icon	**Name / Function**	**Section**
	**Gel Card Grip Place (Single Step)** Place a lifted gel card to a sequence position and opens Gripper	
	**Gel Card Grip Move (Single Step)** Move a gel card to a position without opening Gripper	
	**Gel Card Grip Read Barcode (Single Step)** Move a gel card in front of the Autoload Barcode Scanner and read	
	**Gel Card Grip Punch (Single Step)** This step punches the foil of a labware item (gel card) from the selected sequence.	

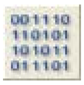
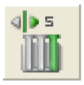
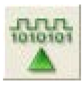
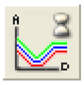
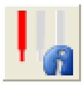
Transport: Punch Card Gripper		
Icon	**Name / Function**	**Section**
	**Punch Card Grip Get (Single Step)** Allows to grip and lift a card or a plate	
	**Punch Card Grip Place (Single Step)** Place a lifted card or plate to a sequence position	
	**Punch Card Grip Move (Single Step)** Move a card or plate to a position without opening Grippers	
	**Punch Card Grip Set Barcode (Single Step)** Allows to set a given barcode to the sample tracking database for the gripped labware item	
	**Puncher Move (Single Step)** Move the puncher head in Z to a sequence position	
	**Puncher Position Plate (Single Step)** Move the plate frame on the puncher to a position	
	**Puncher Punch (Single Step)** Punch a disc from a card into the well of the target plate	

Labware Handling Channels: AutoLys Tube		
Icon	**Name / Function**	**Section**
	**AutoLys Tube Cap (Single Step)** This step allows capping of AutoLys tubes.	
	**AutoLys Tube Decap (Single Step)** This step allows decapping of AutoLys tubes.	
	**AutoLys Tube Get (Single Step)** Grip and lift an AutoLys tube	
	**AutoLys Tube Lift and Lock (Single Step)** Allows to "lift and lock" the inner part of the AutoLys Tube	

Labware Handling Channels: AutoLys Tube		
Icon	**Name / Function**	**Section**
	AutoLys Tube Move (Single Step) Moves an AutoLys tube to a position without opening Grippers	
	AutoLys Tube Place (Single Step) Places an AutoLys tube to the specified sequence position	

Labware Handling Channels: Tube Twister		
Icon	**Name / Function**	**Section**
	Tube Twister Close Cap (Single Step) Used to tighten a cap on the tube within the selected Capper Station.	
	Tube Twister Get (Single Step) Grip and lift a tube	
	Tube Twister Move (Single Step) Move a tube to a position without opening Grippers	
	Tube Twister Open Cap (Single Step) Loosen a cap on a labware item (tube) on the selected Capper Station.	
	Tube Twister Place (Single Step) Place a lifted tube	
	Tube Twister Read Barcode (Single Step) Reads the barcode of a gripped tube with the integrated Barcode Reader	
	Tube Twister Mix This step spins the gripped labware item.	

Labware Handling Channels: Micronic Tube Handler		
Icon	**Name / Function**	**Section**
	Micronic Tube Cap (Single Step) Allows capping of Micronic Tubes.	
	Micronic Tube Decap (Single Step) Allows decapping of Micronic Tubes.	
	Micronic Tube Get (Single Step) Grip and lift a Micronic Tube	
	Micronic Tube Move (Single Step) Moves a Micronic Tube to a position without opening Grippers	
	Micronic Tube Place (Single Step) Places a lifted Micronic Tube	

Miscellaneous		
Icon	**Name / Function**	**Section**
	**Move Auto Load (Single Step)** Moves the autoload to a selected track number	
	**Firmware Command (Single Step)** Send the specified firmware command	
	**1000µl Channel Move to Position (Single Step)** Moves the pipetting channel to an absolute position, or to one relative to the current position, or to a sequence	
	**5 mL Channel Move to Position (Single Step)** Moves the pipetting channel to an absolute position, or to one relative to the current position, or to a sequence	
	**Read Port (Single Step)** Defines from which Microlab instrument port the status is requested (port 1 or 2)	
	**Write Port (Single Step)** Defines the status of a port from the Microlab instrument (port 1 or port 2)	
	**Tip Tracking Speed (Single Step) (obsolete in VENUS three)** This step sets a correction value to optimize the speed of liquid following during an aspiration or dispensation step	
	**Wait for TADM Upload (Single Step)** This step is waiting until all TADM data is uploaded.	
	**Get Channel Exclude State (Single Step)** This step returns the state of the pipetting channels	
	**CO-RE 96 Head Move (Single Step)** Used to move the CO-RE 96 Head to an absolute position, or relative to the current position, or to a sequence	
	**CO-RE 384 Head Move (Single Step)** Used to move the CO-RE 384 Head to an absolute position, or relative to the current position, or to a sequence	

5.3 System Deck

The System Deck Editor window can display two different views of the instrument.

The system view (available by pressing the System Tab in the lower-left of the window) shows the instrument from the outside including all surrounding instruments such as third-party components etc.

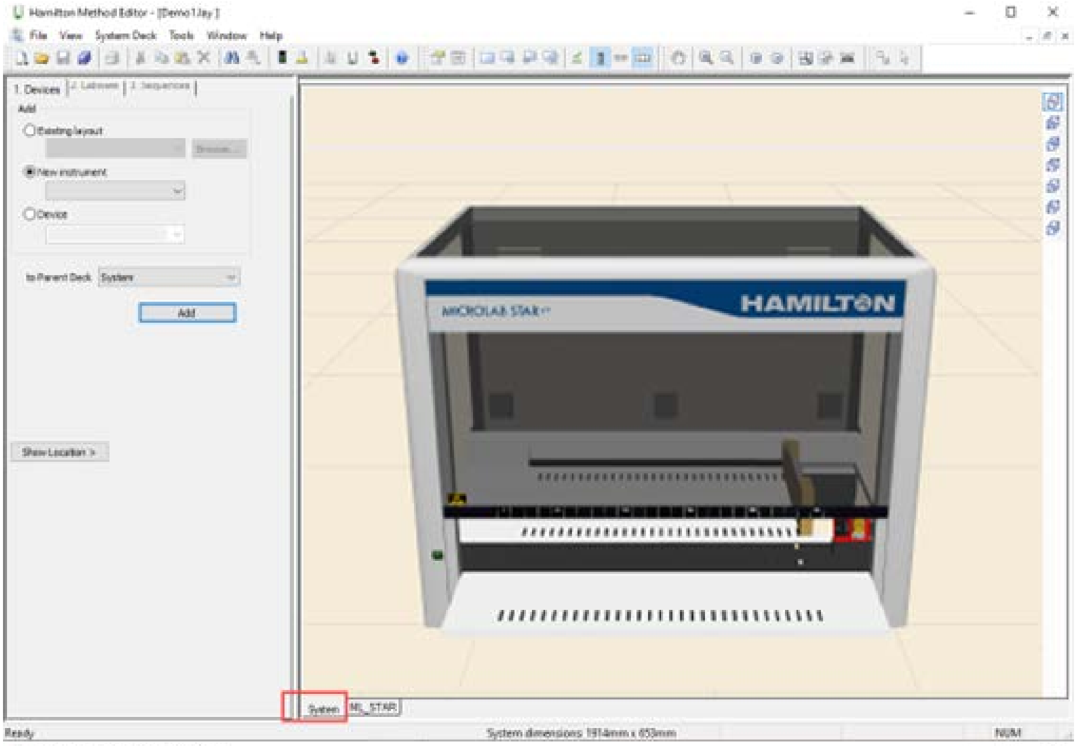
The programmer is able to design the real environment in the software by adding instruments (e.g., Microlab instrument pipetting workstation, Track Gripper plate handling arm, Entry Exit, etc.) and other devices (e.g. readers, washers, incubators, plate hotels, etc.) onto the system deck.

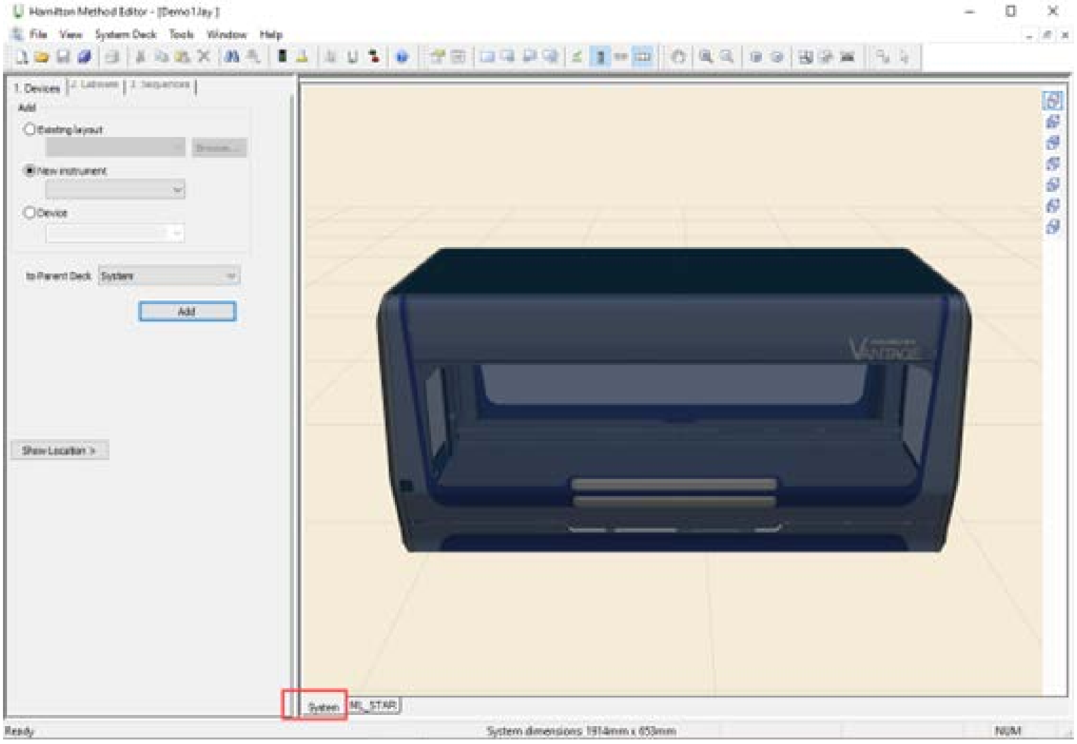
NOTE

As of this release, the Microlab STAR^{V} and VANTAGE will be named as ‘ML_STAR’ instruments on the deck layout and the method steps.

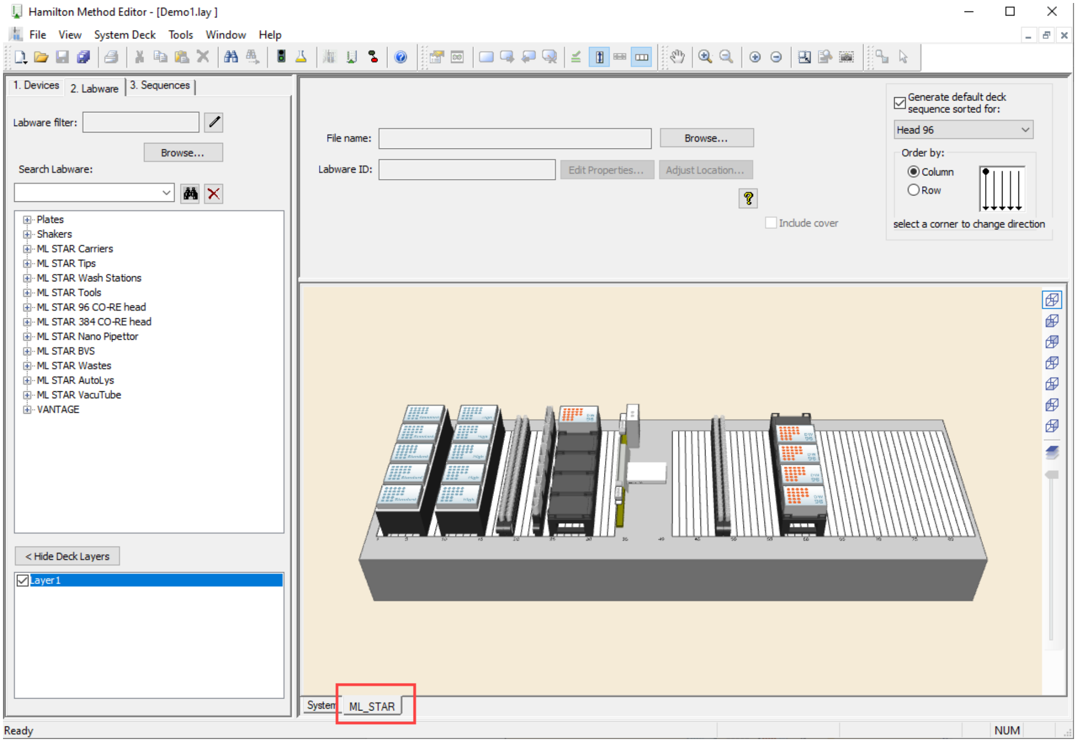
Instruments added on the system deck are ***graphical representations*** *of the instruments only. To control these instruments, driver libraries must be added to the method.*

The **System Tab** shows the Microlab instrument in 3D view.





The “**ML_STAR**” Tab shows the deck layout of the Microlab instrument in 3D view.



Each added instrument will be represented by its specific instrument tab.

The instrument is represented by a deck. A deck layout is a graphical illustration of the work surface of a pipetting robot. It contains all the information about the carriers and labware being used, as well as the x/y/z coordinates of the positions.

5.4 Sequences

Definition

VENUS Software uses sequences for pipetting, transport, tip handling etc.

A sequence is a list that specifies the order of execution. A sequence contains three columns:

Index - column that identifies each sequence position.

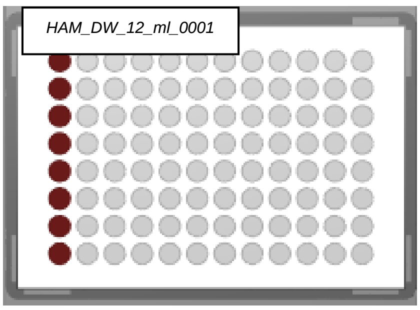
Labware ID - specifies the labware on which the position is defined (e.g. Plate)

Position ID - container on which the sequence position is defined (e.g. Well A1)

Refer to the example shown below.

Example:

	Labware	Position
1	HAM_DW_12_	A1
2	HAM_DW_12_	B1
3	HAM_DW_12_	C1
4	HAM_DW_12_	D1
5	HAM_DW_12_	E1
6	HAM_DW_12_	F1
7	HAM_DW_12_	G1
8	HAM_DW_12_	H1



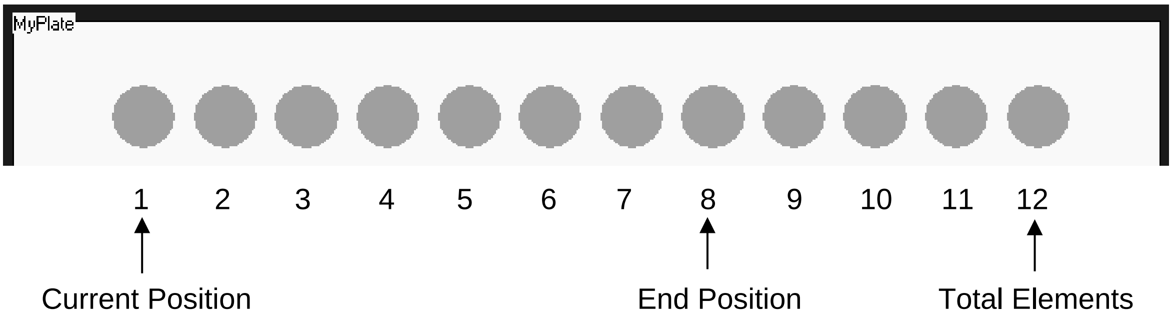
Almost every action of an instrument requires two sequences: a source and a target sequence. This is true for pipetting from an aspirate to a dispense sequence, when transporting labware from the getPlate sequence to the placePlate sequence, when coupling tips from the pickup sequence and ejecting it to the waste sequence etc.

Pointers:

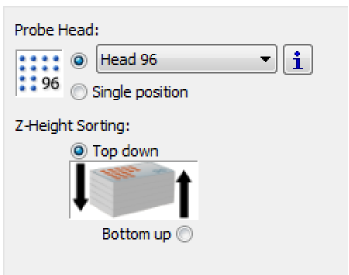
Every sequence (such as the ones shown below) has three pointers which may change during a run:

- Current position (which is the ***next available*** position in the sequence)
- End position (the ***last position*** to be used)
- Total number of elements (the overall length of the sequence)

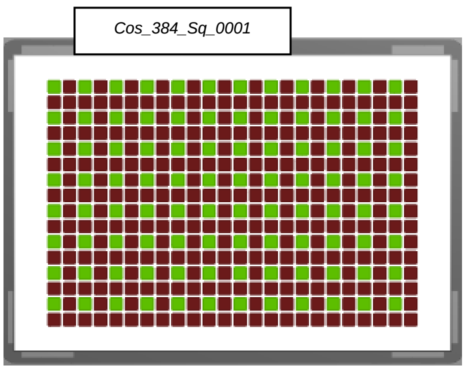
Consider this Example:



Within the method, all three pointers may be modified or requested through the appropriate functions.



A **default sequence** is created automatically when plates or tube racks are placed on the deck. The sorting of these sequences depends upon the selected stamp tool.



Having for instance, the stamp tool for the CO-RE 96 Probe Head activated, the default sequence will be optimized to be processed with this head.

Custom sequences can be defined graphically in two ways:

- By clicking on each appropriate well on the plate or tube rack
- By using the rubber band zoom

In both ways, the currently selected "**Stamp Tool**" is used to sort the added position.

Example:

On the instrument, an action to aspirate samples from tubes and dispense into microplates in a manner that will be free of contamination – in which case disposable tips will be preferred.

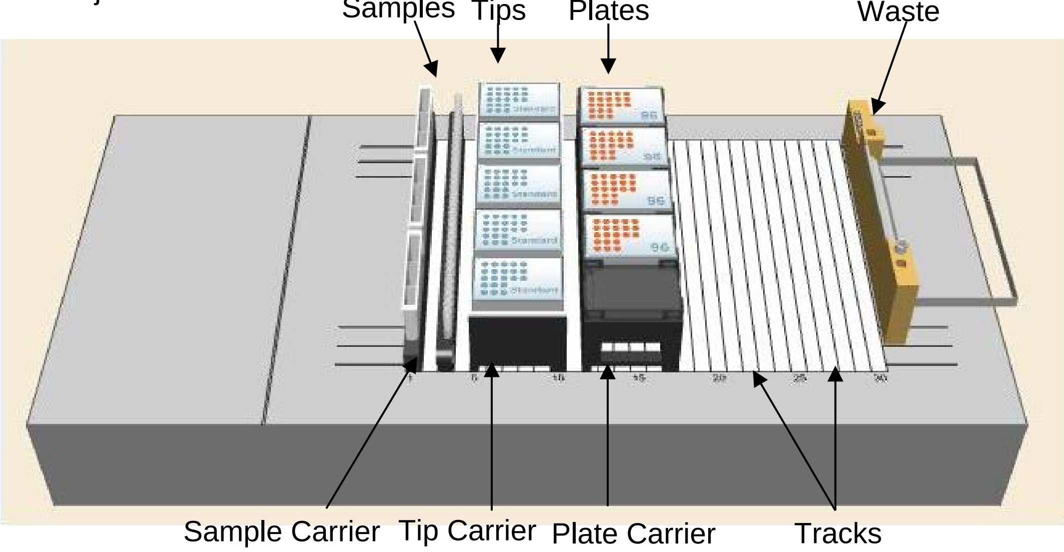
The example describes such a method:

Method: 1. Pick up Tips

2. Aspirate from Samples

3. Dispense to Plates

4. Eject to Waste



Definitions

- A **Method** describes operations on sequences.
- A **Deck Layout** describes the physical positions of labware on the instrument deck.
- A **Labware** item describes the geometry of parts which can be dealt with as a whole, such as the wells of a Microplate, or which can be combined on the deck, such as a carrier holding several plates.
- A **Sequence** describes the order in which labware positions are to be processed on the instrument's deck.

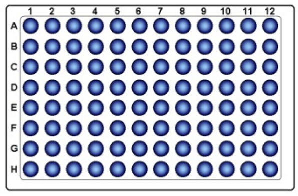
Advantage of Using Sequences

The advantage of working with sequences is the unrivalled flexibility.

As described in the Section 5.3 System Deck, VENUS Software obtains the coordinates of the pipetting spots out of sequences.

In an example of a 96-well Microplate and the 8-Pipetting Channel STAR, the default pipetting order is:

1st pipetting:	A1; B1; C1; D1; E1; F1; G1; H1	(= column 1)
2nd pipetting:	A2; B2; C2; D2; E2; F2; G2; H2	(= column 2)
3rd pipetting:	A3; B3; C3; D3; E3; F3; G3; H3	(= column 3)
4th pipetting:	A4; B4; C4; D4; E4; F4; G4; H4	(= column 4)
…	…	



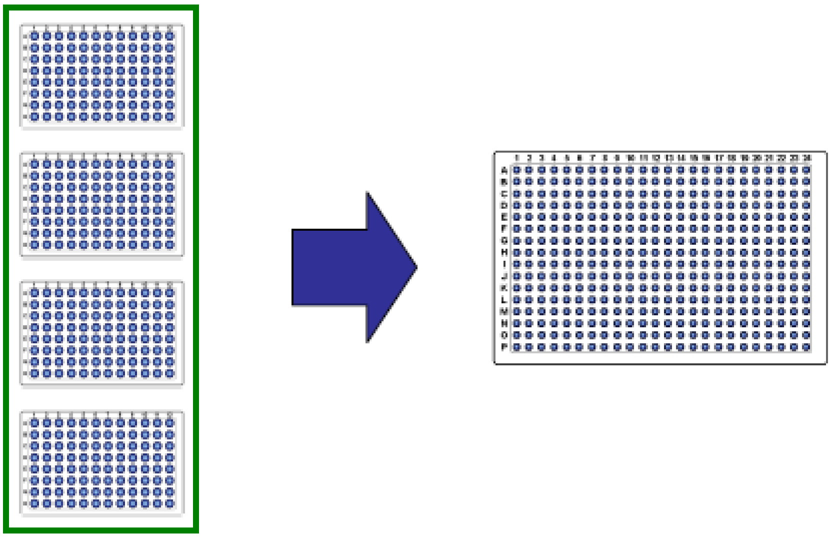
Because labware is represented by a sequence (refer to Section 5.4 Sequences), there is the ability to change the order of the pipetting steps, for example:

- Excluding of complete columns or single wells
- Sorting the pipetting spots by the characteristics of the: x-/y- coordinates, Position ID, etc.

Another advantage of the sequence philosophy is the option of merging several sequences like in the case of reformatting four 96-well Micro Plates to one 384-well Micro Plate.

Now all of the pipetting spots of the 96-well plates are merged into one sequence.

Only **one pipetting step** is now needed because of merging the 96-well Micro Plates to one source sequence.



The programming of the reformatting steps then becomes much easier. Within only one pipetting step, all wells of the four 96-well Micro Plates can be transferred to the 384-well Micro Plate.

5.5 Labware

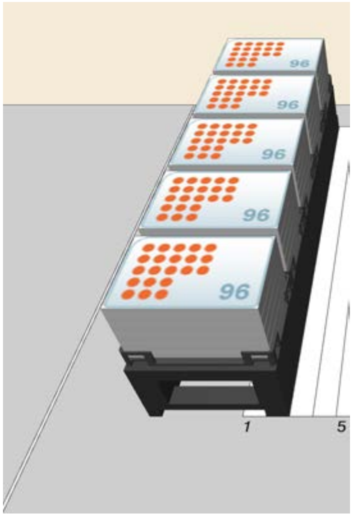
The software comes with a set of definitions for the standard labware items that are most commonly used in laboratories. The Carrier name and definition selected must always be identical to the name with which the physical carrier is labeled.

A Carrier feature is a varying number of locations for the placement of labware, such as tip racks, Micro Plates, etc.

For the standard Carriers and their names, refer directly to the software list which is shown through an image in the next page. The list box displays the available labware names. The selected piece of labware is defined briefly on the right-hand side of the window with its corresponding picture.

The benefits of a Plate Carrier to the left of Track 1 are:

- Higher deck density, lower reloads
- Work positions available for Multi-Probe Heads
- Labware storage positions available for iSWAP / CO-RE Grips



Definition

A **labware** item describes the geometry of objects which can be dealt with as a whole, such as the wells of a microplate, or which can be combined on the deck, such as a carrier holding several plates.

The Microlab instrument can be used with all kinds of labware such as tubes, micro plates, reagent troughs, etc. VENUS Software comes with a set of standard labware definitions.

Pre-loaded Tips and Tube Carriers

In addition to plate and Tip Carriers, Sample Carriers can also be placed on the deck. These carriers are pre-loaded with tubes of a specified size (diameter and length). In fact, these carriers are racks (*.rck) in contrast to the other carriers, which are carrier templates (*.tml). The Tube Carriers are examples of racks which fit directly into the track grids of the Microlab instrument.

There are also pre-loaded racks and carriers available; for example, a Tip Carrier equipped with standard tips (TIP_CAR_480_ST_A00) and more.

Naming Convention of Carriers

Example: PLT_CAR_L5_AC_A00

Type of Carrier	Construction	Orientation	Number of	Labware Info	Revision
PLT Plate Carrier	**CAR** Standard Carrier **APE** Application Engineering **DAT** Deck Adaptor Template	**L** Landscape **P** Portrait	Plate Positions: **3** **4** **5**	**AC** Deep Well Plate (archive) **MD** Medium Density (96/384-Well) **HD** High Density (1536-Well) **LI** Limbro **ST** Stack	**A00, A01**
SMP Sample Carrier	**CAR** Standard		Tubes: **12** **16** **24** **32**	**15x75** Tube Size	**A00, A01**
TIP Tip Carrier	**CAR** Standard Carrier	**L** Landscape **P** Portrait	1000µl-channel: **288** **384** **480** 5 mL tips: **72** **96** **120** 384 Head: **1920**	**LT** Low Volume 10ul **50µl** 50 µL Tip **ST** Standard Vol. 300 µL **HT** High Volume 1000 µL Filter **5mlT** 5 mL Tip Size **BC** Bar-Coded Tip Rack **NTR** Nestable Tip Rack	**A00, A01**
RGT Reagent	**CAR** Standard		Reagent troughs: **3, 4, 5**	**R** Reagent	**A00, A01**
CTR Control Carrier				**C** Controls	**A00, A01**
VER Verification					**A00, A01**

5.5.1 Types of Labware

Containers

Containers are vessels which hold liquids. They are usually placed in racks. They can also be placed directly onto the carriers, which is the case with reagent containers. An example for containers would be the tubes or the wells of a Microplate.

The filename has the extension “**.ctr**”.

NOTE

Pipetting is only done into containers!

Sequences can be created on containers only!

CO-RE tips and needles are also defined as containers.

Rectangular Racks and Plates

Rectangular racks are specialized grids for holding containers (or tips) in row and column order. A Micro Plate is a rack in this circumstance and the wells represent as containers. The rack is therefore a template describing a discrete number of positions for holding containers (or tips).

Examples of racks include a Tube Rack, a Micro Plate, a Microtiter Strip, a Deep-Well Plate, and a Tip Rack.

The filename has the extension “**.rck**”.

Circular Racks

Circular racks are specialized grids for holding containers in a segment of a circle.

The filename has the extension “**.crk**”.

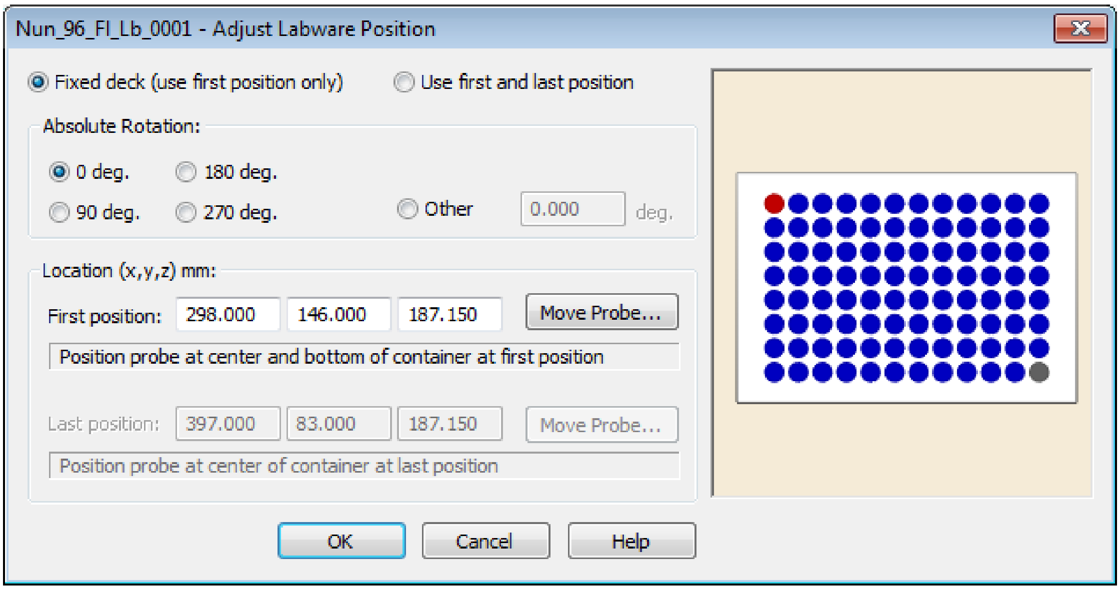
5.5.2 Reference Position

The command dialogs always use the container bottom as a reference position (fixed height, liquid level = 0).

The x, y, z values of the reference well (usually A1 or #1) stored in the instrument's system of coordinates are shown in the "**Adjust Labware Position**" Dialog.

To access this dialog:

1. Switch to the System Deck Editor.
2. Select the labware (Plate, TipRack etc.) to be used.
3. Right click on the labware item to enter the Context Menu.
4. Select "**Adjust Location**".
5. The dialog box shown below will appear.

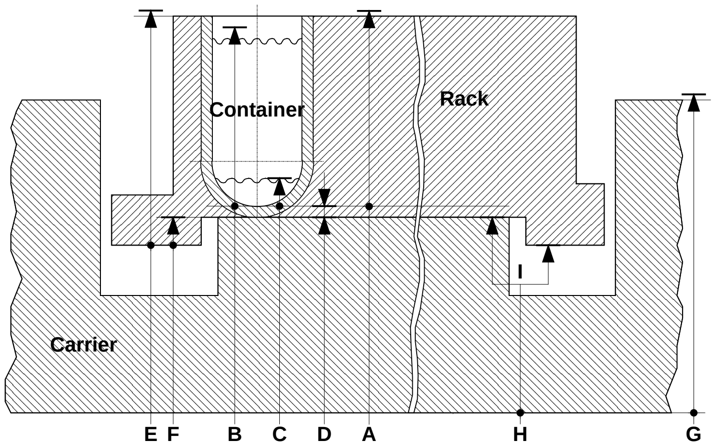


5.5.3 Height Parameters

Both the rack and the container have clearance heights. This is the minimal height at which the pipetting channels must pass over the labware so that their movement is not blocked by the labware. The software automatically takes the highest clearance height.

The maximum pipetting height is the deepest position the tip or needle can be placed inside a well. It is counted from the bottom of the container upwards and determines the dead volume of the container.

The LLD search height is the height at which the speed of the pipetting channel is reduced to look for the liquid surface.



Z positions of carriers, racks and containers

Container:

A: Clearance height

B: LLD search height

C: Maximum pipetting height

D: Container bottom thickness

Rectangular Rack:

E: Total calculated clearance height

F: Distance from rack base to container base

Carrier (Template):

G: Clearance height

H: Origin Z (also called "Base Offset")

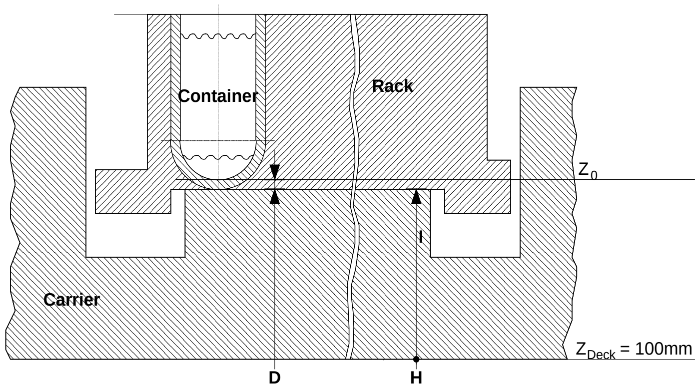
I: Snap to site of base of rack or to container

Regarding the z position, two different cases of rack placement on the deck can be distinguished in the instrument which is controlled by the switch "I" in the sketch above:

- Snap to base of container
- Snap to base of rack

These two cases are described in detail on the next page.

A “**Container-Based**” Rack is placed with the container bottom directly on the carrier (e.g. the Micro Plates on a Plate Carrier):



The reference position Z_0 is the lowest position in the well. Here, the reference height is calculated from

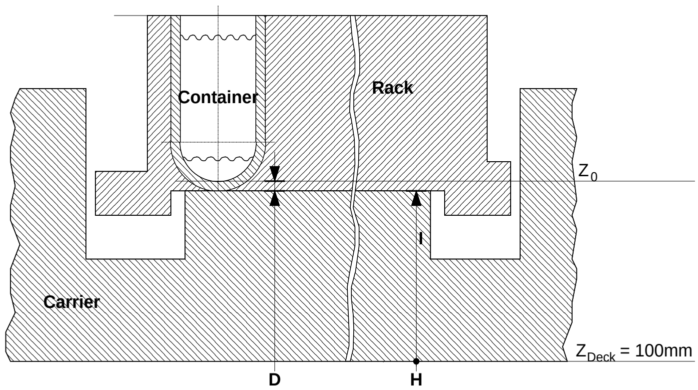
$Z_0 = Z_{deck}+Z_{carrier}+Z_{thick}$,

Z_{deck} is fixed at 100 mm,

Z_{thick} (D) is defined in the container labware,

$Z_{carrier}$ (H) is defined in the carrier template definition.

A “**Rack-Based**” Rack is placed with the frame on the instrument deck (e.g., a Tube Rack, where $Z_{carrier}$ (H) = 0, because the tube rack is used directly as a Carrier)



The reference position Z_0 is the lowest position in the well. The reference height is calculated using:

$$Z_0 = Z_{deck}+Z_{carrier}+Z_{basediff}+Z_{thickness}$$

Z_{deck} is a fixed quantity

$Z_{carrier}$ (H) is defined in the carrier definition

$Z_{basediff}$ (F) is defined in the rectangular rack labware

$Z_{thickness}$ (D) is defined in the container labware

PART III: Working with VENUS Software

6 How to Create a Deck Layout

In the deck layout section of the method editor, the programmer declares what labware (carriers, racks or containers) is used in the procedure and where these items are positioned on the deck. This action is called “**Defining a Deck Layout**”.

Creating a new method will also create a deck layout with the same name as the method.

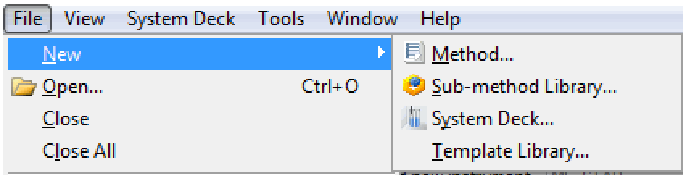
6.1 New Deck Layout

The easiest way to have a new deck layout is to create a new method. This will automatically create a new deck with the same name.

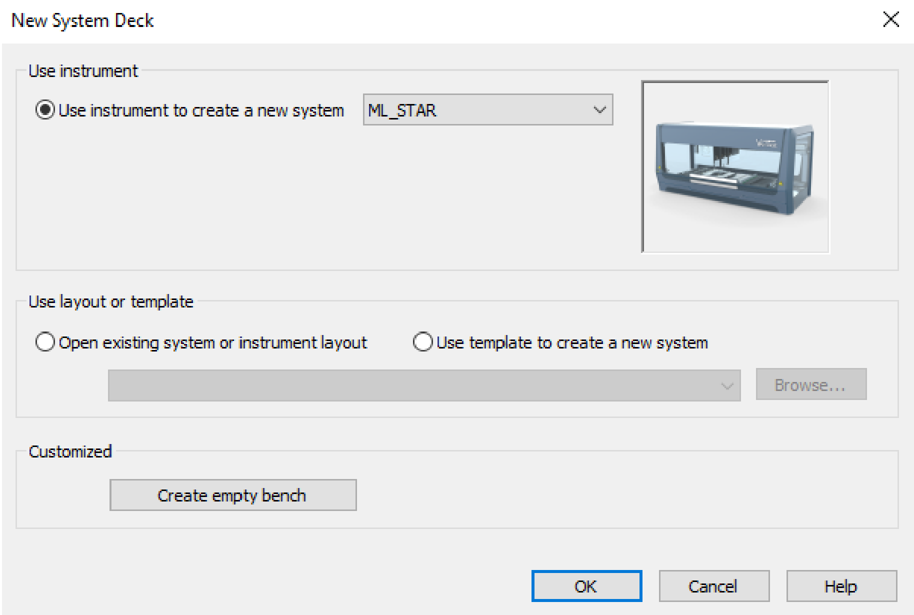
To access the newly created deck, simply click on the System Deck Icon:

System Deck

It is possible to create a deck layout without creating a new method. To do this, select “**File → New → System Deck**” in the Method Editor.

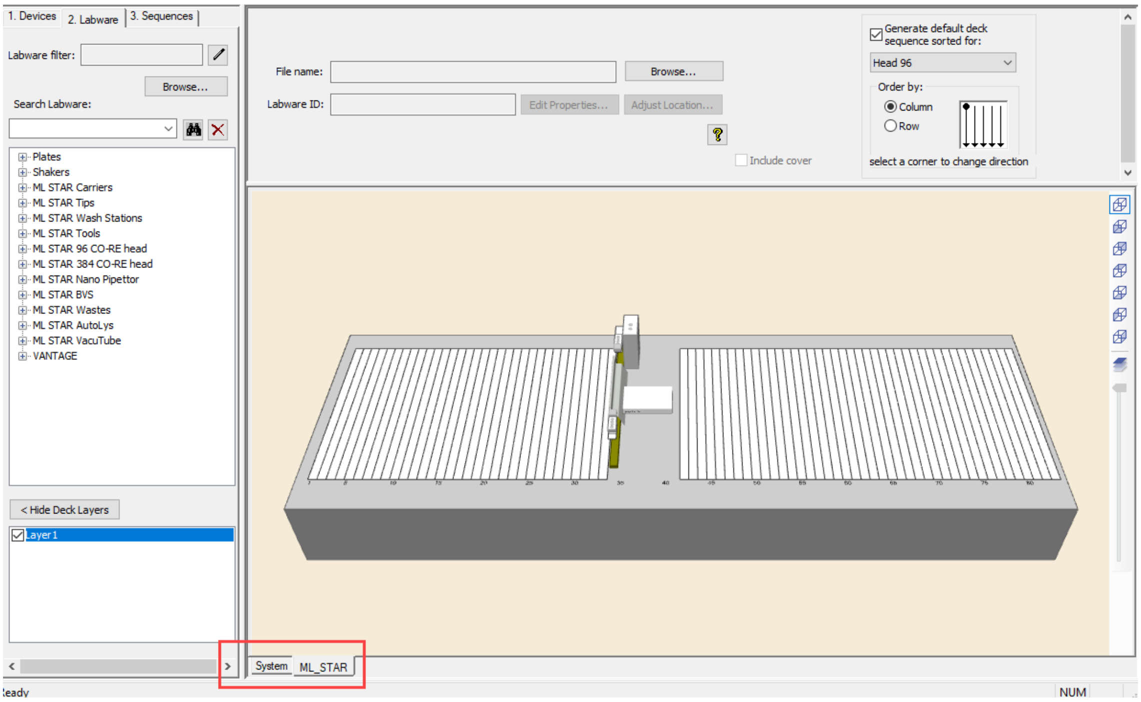


Fill out the dialog box being prompted as shown below.



- “**Open existing system or instrument layout**”, to base the deck layout on an existing layout or on a template.
- “**Use template to create a new system**” with an instrument-specific definition of dimensions, main grid and hidden grids, with pre-loaded standard labware such as a waste container. Deck templates can be customized.
- “**Use instrument to create a new system**”, to base a layout on the layout of the connected instrument. This is the most common way to start.
- “**Create an empty bench**” to start creating a layout with an empty deck.

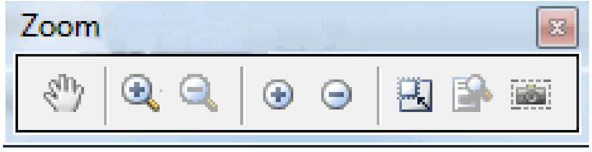
Once a choice has been made and possibly browsed for an existing template or layout, the menu bar will include the “**System Deck**” Menu and the “**System Deck**” Window will be displayed. The Deck Layout shows the tracks of the predefined grids from the appropriate deck template.



Use the tabs in the lower-left corner of the window, highlighted above, to switch between the system view (full instrument and third-party components) and the instrument tab (here: ML_STAR) to see the instrument deck.

Use the tabs on left vertical panel to add new devices/ instruments to the system, add labware to selected instrument (e.g., ML_STAR) or edit the sequences of the deck layout.

The functions in the Zoom toolbar Help in finding the right view.



 Drag and Move the View

 Zoom in with Rubber Band

 Zoom out with Rubber Band

 Zoom in by 25%

 Zoom out by 25%

 Zoom to Make All Components Visible

 Actual Size View (zoom to 100%)

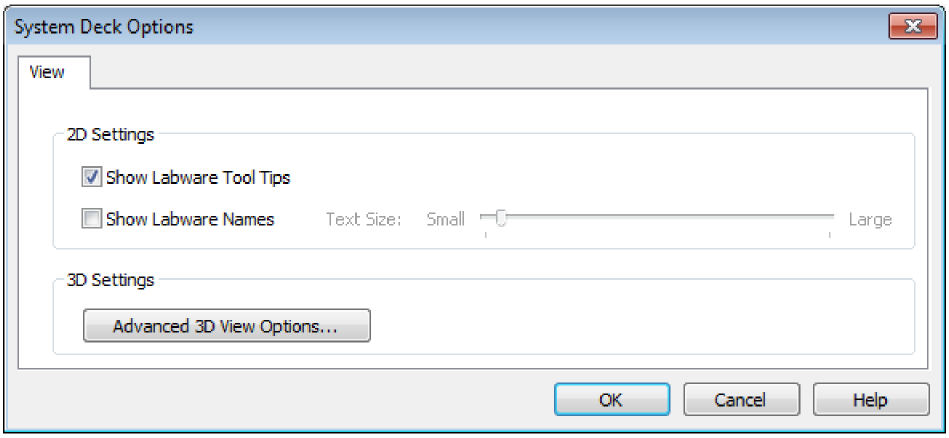
 Make a Screenshot with Rubber Band

6.1.1 System Deck Options

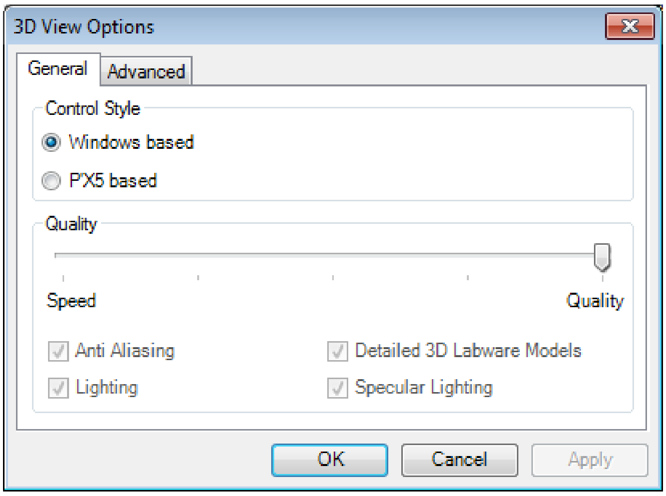
1. The "**System Deck → Options**" Menu includes settings for the appearance of the System Deck.

   It is possible to:

   - Show (or hide) labware tooltips
   - Show (or hide) labware names and specify the text size



2. Clicking the **[Advanced 3D view Options…]** prompts the settings for the:
   - Windows™ based or PX5-based control style
   - Render quality
   - Selection of graphic device for the computer



6.1.2 Save System Deck

The complete loading of the deck may be stored under a programmer-defined file name with an extension "**.lay**", for example, "**MyDemo.lay**". This is done by using the "**Save**" or the "**Save Copy as...**" command found in the "**File**" Menu.

6.1.3 Open Existing Deck Layouts

It is possible to load previously defined deck layouts (extension "**.lay**") by selecting "**File → Open**" in the "**File**" Menu or by activating the [] Button in the toolbar.

6.2 Adding Labware on the Deck

6.2.1 Introduction

Practically, a deck can be configured freely. Here, the term "**Labware**" refers to the Carriers (available from Hamilton), Reagent Troughs, Micro Plates, Sample Tubes (from various manufacturers) and Tip Racks.

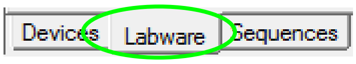
The general steps are as follows:

1. Add a **Carrier** to the deck
2. Add the **Plate** to the Carrier

It is possible to mix plates on a Carrier; like adding Micro Plates to a Deep-Well Plate Carrier. However, please keep in mind that the laser line of the Autoload reads at a specific height and will not be able to read barcodes on the Micro Plates if they are located on a Deep-Well Plate Carrier.

6.2.2 Adding a Carrier by the Search Labware Function

To add labware to the deck layout, make sure the “**Labware**” Tab is selected, highlighted on the picture below.

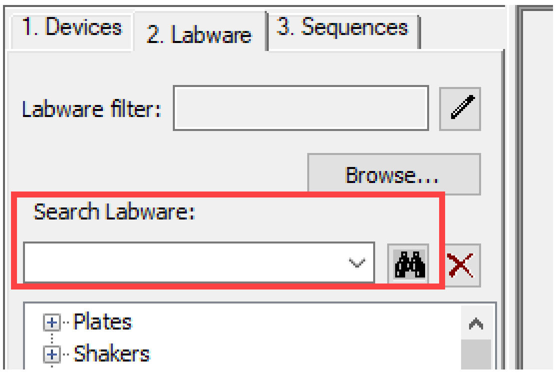


ATTENTION

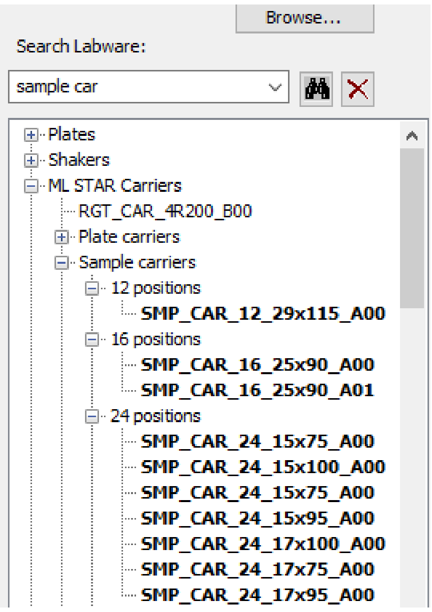
If a tube carrier is placed directly adjacent to a plate or a tip carrier, using low volume tips may lead to a collision when the pipetting *channel moves down to aspirate from the bottom of the tubes. Pipetting channels fitted with low volume tips cannot reach the deck surface. (Δz = 14.5 mm, z_{min} = 114.5 mm from origin) In this case, use larger tip sizes.*

The “**Search Labware**” Function is a powerful tool that simplifies the deck layout creation. Instead of browsing through the tree structure, a simple query in the “**Search Labware**” Field helps with finding the desired labware quickly.

To add a plate carrier to the deck layout, use the “**Search Labware**” Field found on top of the labware window, highlighted on the image below.



To find a labware, enter search criteria in the text field. The box right below the text field will display all the matching results, based on the specified search string. Please note that not only the labware name, but also its description in plain text is scanned to generate the matching results.



Example:

Entering **sample car** will list all matching pre-loaded sample carriers.

Select the desired carrier and “Drag-and-Drop” the item onto the deck layout.

NOTE

*Although it is very simple to find any kind of labware with the “**Search Labware**” Assistant, make sure that the deck layout contains the same Carriers as loaded on the actual deck.*

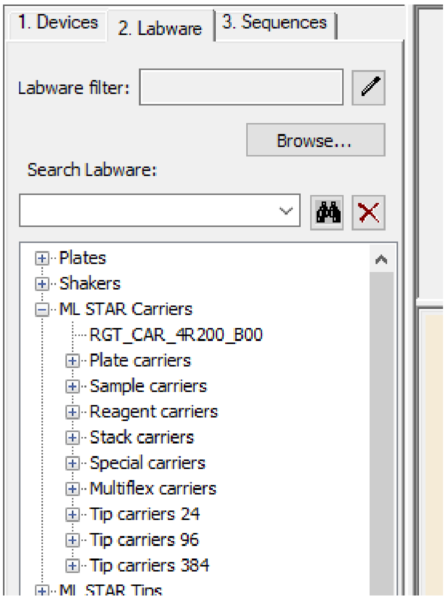
6.2.3 Adding a Carrier through Tree Selection

Another way to find the labware for the deck is to use the tree structure of the labware.

Click the “**Labware**” Tab in the left vertical window.

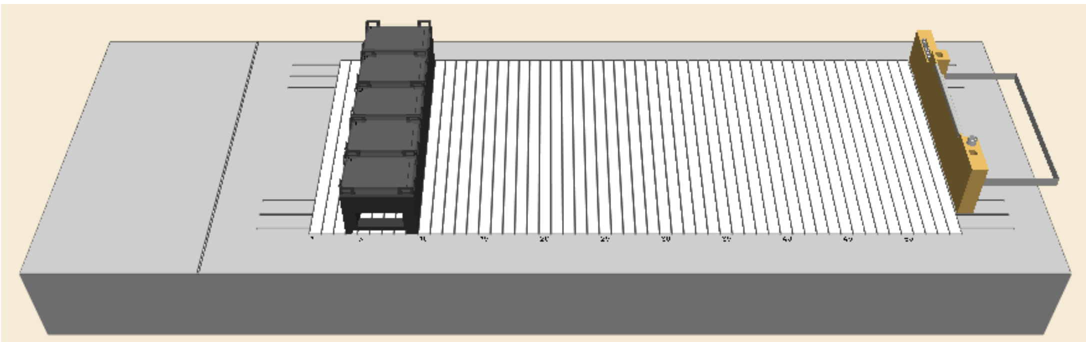
In the tree view located on the left side of the window, click **[Plate carriers]** (Microlab instrument Carriers). The list view in the center shows the available plate carriers.

In the list view located on the center of the window, click the carrier that should be placed on the deck. An image of the Carrier with its corresponding description will be shown on the right-most side of the window. Click on the image of the Carrier and “Drag-and-Drop” it to the appropriate location on the deck. All of these are shown in the picture below.

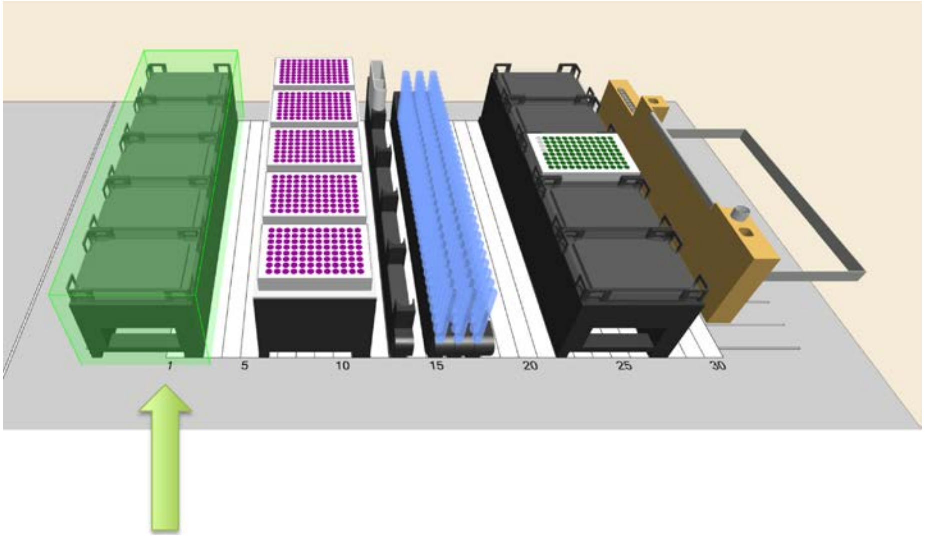


The place where the Carrier will be dropped is indicated by a frame during this operation.

The Carrier, once dropped, is added onto the deck layout and is projected on the left side of the deck as shown below.

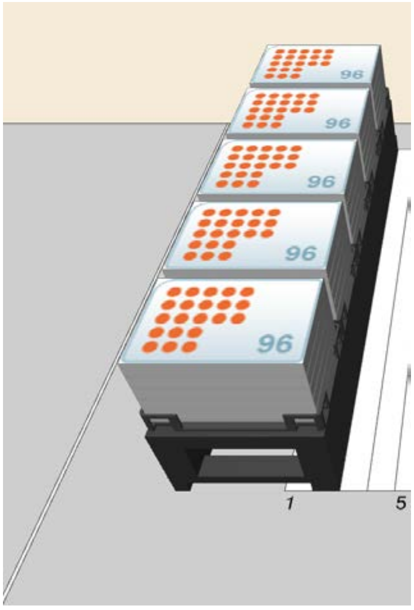


VENUS allows for the optimal use of deck space, allowing to place Carriers in the minus (left of Track 1 (-3 to +3)) area.



The benefits of a plate Carrier to the left of Track 1 are:

- Higher deck density, lower reloads
- Work positions available for Multi-Probe Heads
- Labware storage positions available for iSWAP / CO-RE Grips



6.2.4 Adding a Plate to the Carrier

The next step is to add a plate to the Carrier. The corresponding labware (Plates, Tip Racks, etc.) is sorted by manufacturer and stored within the labware directory. Standard plates from almost every major manufacturer (Nunc, Falcon, Greiner, etc.) are available. These names (or brands) can also be entered directly in the “**Search Labware**” Field.

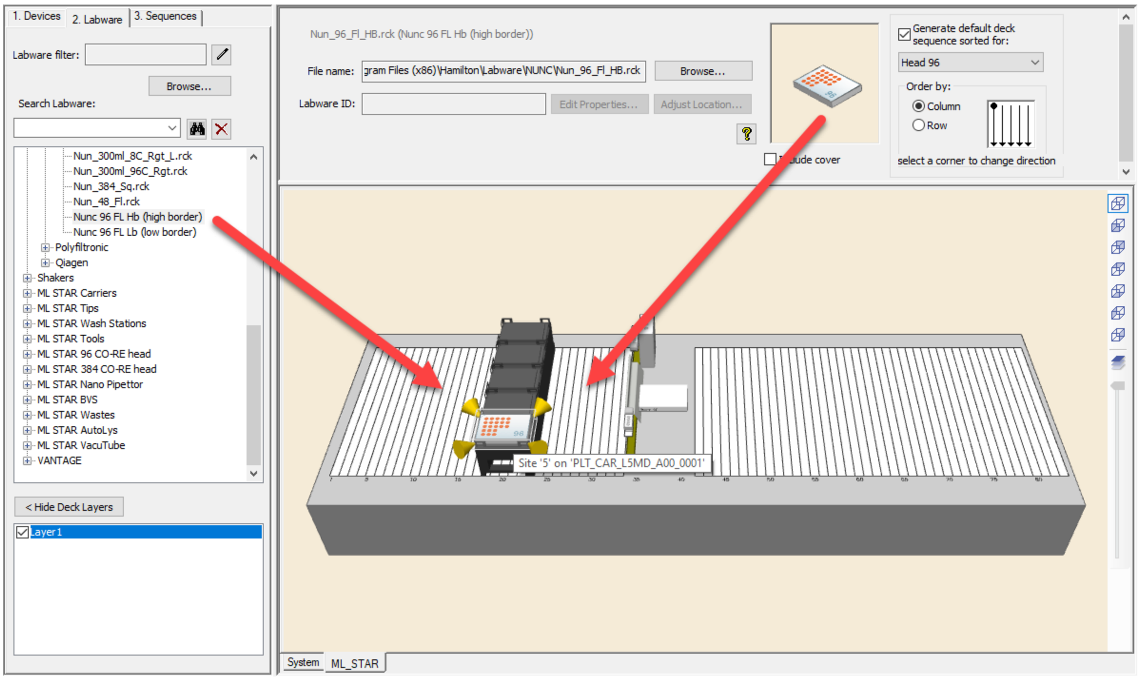
Micro Plates, Tip Racks, etc. are labware items that are generally called “**Racks**” and have “**.rck**” file name extensions.

To add a plate to the carrier, open the “**Plates**” Group from the tree list on the upper-left of the deck layout window. The different groups of plates are displayed in the list.

In the “**Plate**” List, click to select the appropriate plate group. In the list view, found at the center of the window, the plates belonging to the group are shown.

Click the desired plate type from the list. An image of the plate with its corresponding description will be presented on the right-most side of the window. To add the plate to the deck, either drag it from the labware list or the plate image on the right, and "Drag-and-Drop" it onto the location of the appropriate Plate Carrier. These are shown in the image below.

Four yellow arrows will indicate that the plate has been correctly snapped on the carrier position. The plate, once dropped, is added to the deck layout.



When hovered, the tool tip info displays the name of the carrier and its position.

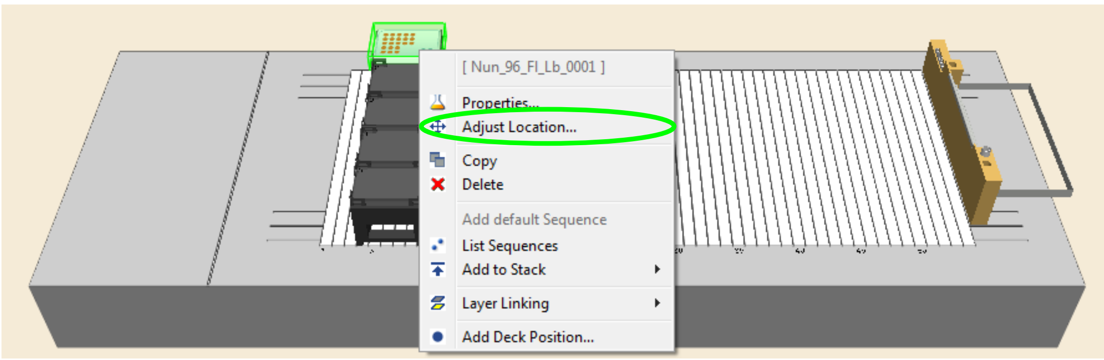
Plates (or labware) on a carrier may be moved between sites by simply "Dragging-and-Dropping".

6.3 Adding a Plate directly onto the Deck

Labware, as seen in the example below, can also be placed directly onto the instrument deck without using a carrier.

Drag-and-drop the item from the image view onto the deck (see the image below). The item is placed wherever the left mouse button is released.

To define the position of the item according to its coordinates, select the item and right-click to enter the Context Menu. Select **[Adjust Location…]** to trigger the settings dialog.

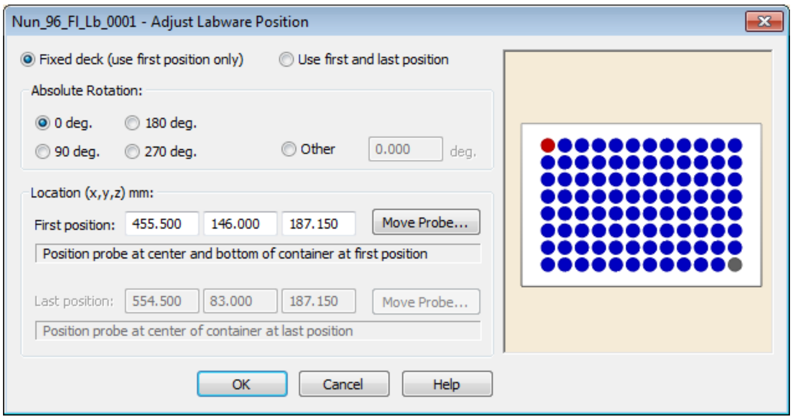


A dialog box which allows settings specifications will appear, as presented below.

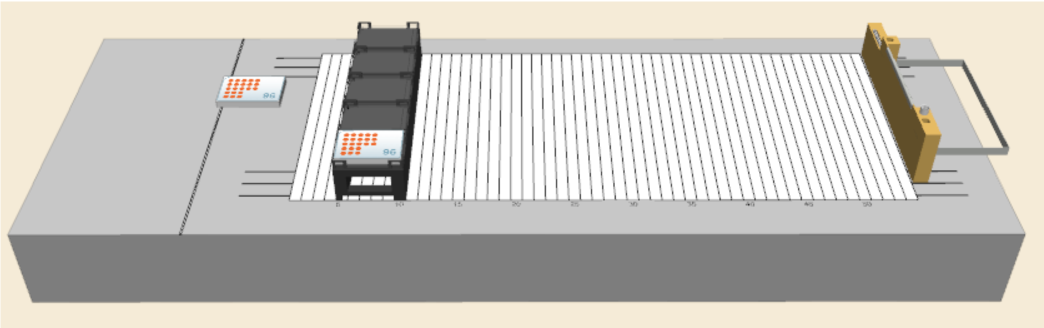
Use the "**Absolute Rotation**" Settings to rotate the labware item freely, "**Other**" or in increments of 90° clockwise.

Enter the desired position coordinates in the entry fields of "**Location**" for the reference point of the selected labware. The reference point is marked red (usually A1 for plates). Note that there is a 100 mm offset from the origin to the deck. This means that a plate placed directly on the deck's surface will have a Z-coordinate of 100 mm.

Click **[OK]** to close the dialog.



The item as seen below is moved to the specified position.



6.4 Removing Labware

To remove labware items from the deck:

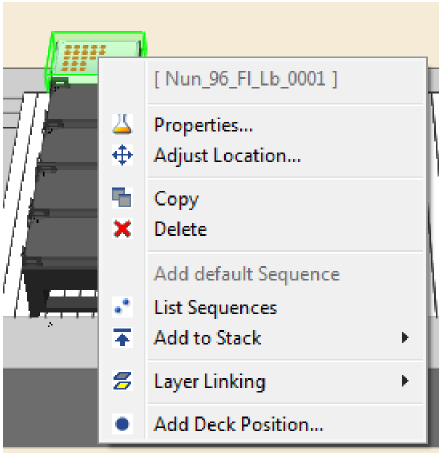
1. Right-click the labware item to be deleted.
2. Select "**Delete**" from the Context Menu.

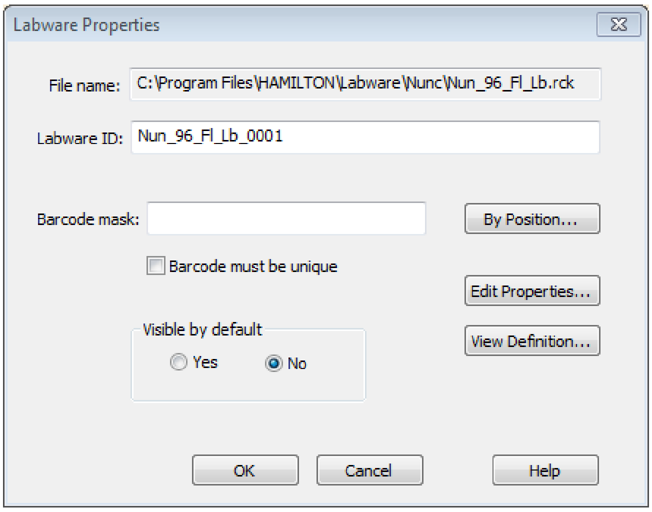
The labware item is deleted from the deck.

6.5 View Properties of Labware / Rename Labware

To rename a given labware item:

1. Right-click the labware item to be renamed.
2. Select "**Properties**" from the Context Menu. The labware properties dialog box will appear, showing the properties of the selected labware.





3. In the labware properties dialog box, the "**Labware ID**", the "**Barcode mask**", the visibility, and the labware's properties or definition (if this data is not write-protected) can be changed.

ATTENTION

Be careful when using 50 µL und 300 µL CO-RE tips on the same deck layout. Do not confuse one tip type with the other during the process of loading the deck. Check the label of the tip rack to see where the tip volume, in plain text, can be seen. Using an Autoload, activate the checkbox for the barcode mask of the tip racks. This will read the barcode label of the tip racks during loading time and prompts a message if the tips loaded are not the expected tips. The same applies when using 300 µL Slim and 1000 µL CO-RE tips.

Using the inncorect tip size may damage the pipetting channels!

6.6 Teaching Labware with the 1000 µL-Pipetting Channels

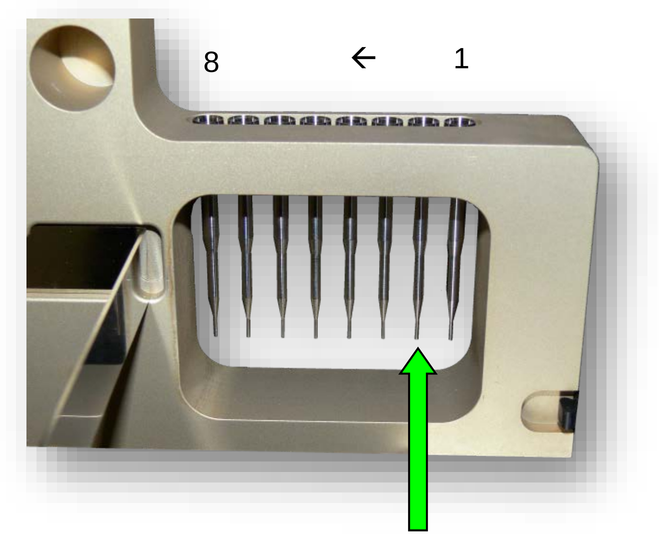
Teaching means:

- Manually guiding a pipetting channel to a particular location on the deck with tools in the VENUS Software.
- Assigning the name to the location.
- Instructing the Microlab instrument to "**remember**" it.

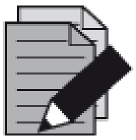
The precise position (the x-y-z coordinates) of labware items can be "**taught**" in this way, using the needle from the Teaching Station.

Later, the only need is to specify the Lab ID and the instrument instantly "**remembers**" exactly where it is positioned on the deck.

The image below shows a teaching station on the waste block with all needles loaded.



To teach the labware, only the 2nd needle is picked up by the pipetting channel. Make sure that at least this needle is loaded before activating the "**Move Probe**" Function.



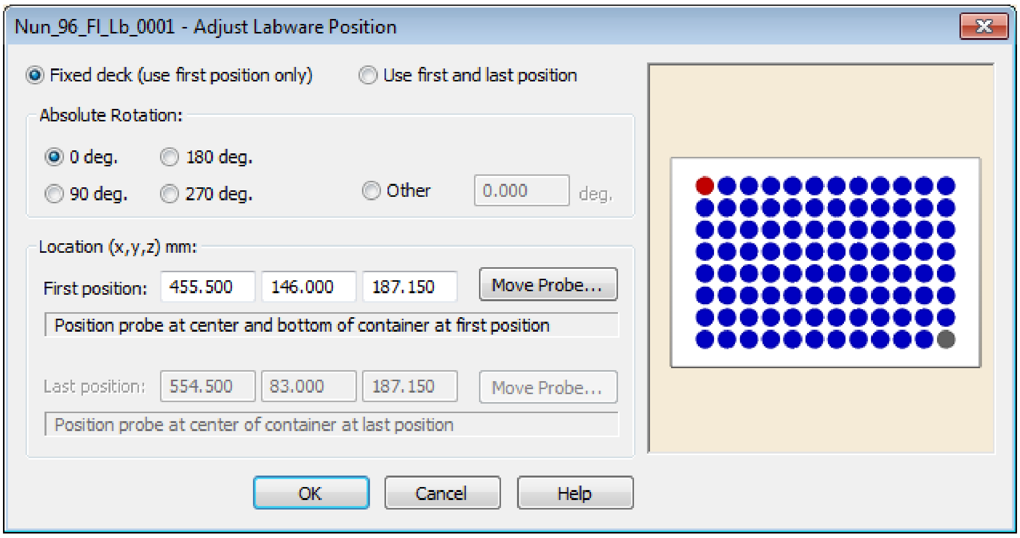
NOTE

The teaching needles are available as an option and are not part of the standard delivery package. They are also used in the maintenance methods.

If the labware to teach is not on the deck, add it.

To teach a labware position:

1. By drag-and-drop, pre-position the plate on the deck.
2. The “**Adjust Labware Position**” Dialog will be displayed, as seen on the picture below.



- The **red-colored well** (in the top row) indicates the “**first position**”.
- The **gray-colored well** (in the bottom row) indicates the “**last position**”.
- **Location:** The default x/y/z- coordinate is determined by the mouse-cursor when dropping the labware onto the deck. Keep in mind that the Z-Axis has an offset of 100mm
- **Rotation:** The labware can be rotated and aligned as required. The rotation functions and the orientation of the labware can also be adjusted.

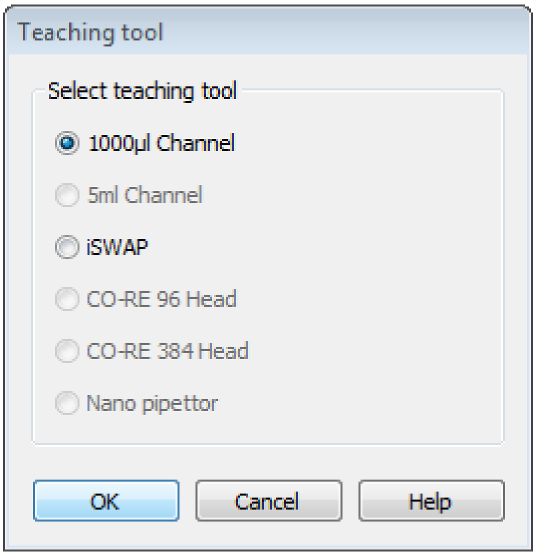
ATTENTION

Before the pipetting channel with the teaching needle is moving from the waste block (picking up position) to the rough position of the labware, make sure there is no danger of collision with objects higher than 140 mm (from the deck surface).

Normally only the first position has to be taught. If the rows and columns of a chosen labware do not correspond to the x/y-movements of the pipetting channels, there is the option to teach the first and the last position. This will allow for compensating for the slightest inaccuracy between the labware and the pipetting channel.

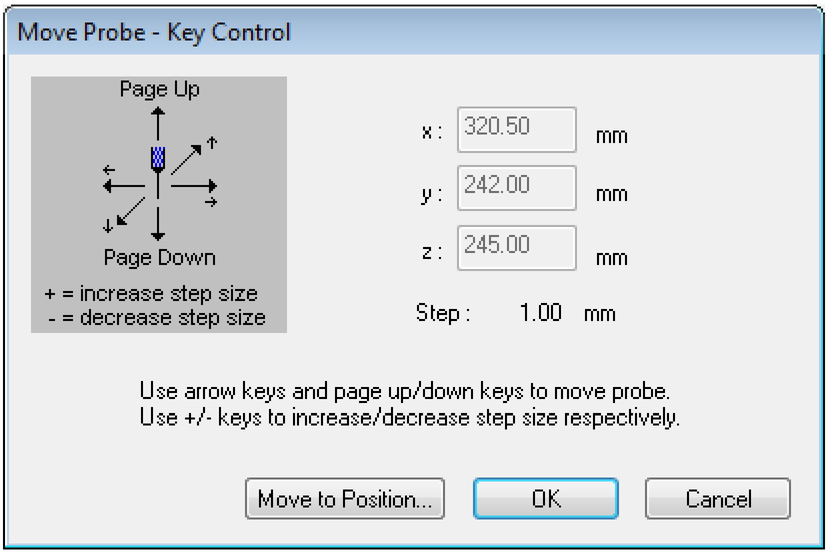
The Move Probe Function

1. Click on the **[Move Probe]** and the following screen will be displayed.



2. Select "**1000µl Channel**"

3. The pipetting channel will be moved to the waste block to pick up the teaching needle. With the teaching needle, the instrument moves to the x/y coordinate of the position to be taught.
4. The "**Move Probe – Key Control**" Dialog will appear.



5. As an alternative to step 2, select **[Move to Position...]** from the "**Tools**" Menu.
6. When **[Cancel]** is clicked, the pipetting channel is lifted in Z-direction first (to the traverse-height) before moving to the waste block. At the waste block, the pipetting channel ejects the teaching needle. No new positions are stored.

   The keyboard (arrow keys, page up, page down) can guide the pipetting channel to the labware position. The step size is 1 mm per default. Use +/- keys to increase / decrease the step size.

ATTENTION

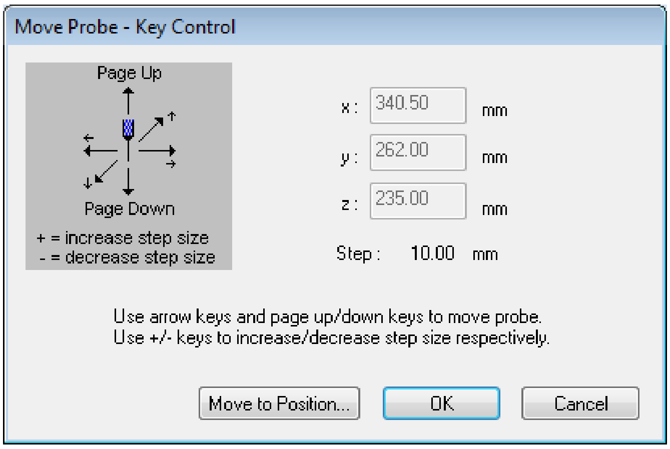
Make sure that the chosen step size will not trigger a crash. The system can be seriously damaged when colliding with the pipetting *channels. All collision control must be made by the user*.

7. The correct position is reached when the pipetting channel is at the bottom of the first or last well (corresponding with the red / gray position of the labware as seen in the picture of step 2) of the labware. Prevent the pipetting channel from crashing into the labware.

NOTE

*The reference point for "**teaching**" is the lower-end of the teaching needle and the reference well of the rack which is usually marked in red (upper- and left-most well = A1).*

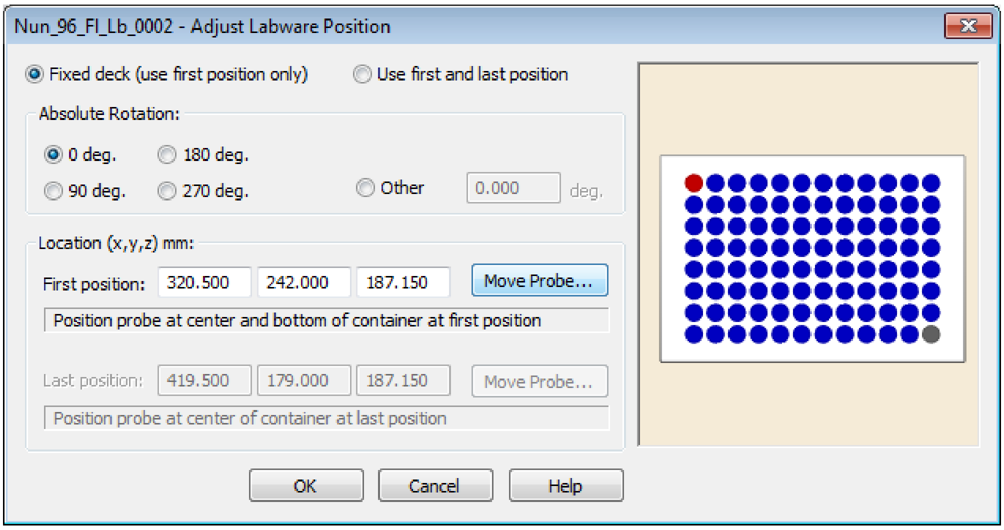
The current position and increment step size is displayed.



ATTENTION

Make sure that the step size is appropriate. When moving the x-arm with a step size of 100 mm, make sure to decrease the step size first. Otherwise, the pipetting channel will move 100 mm downwards and will possibly crash.

8. Once the position is reached, click **[OK]**. The coordinates are then stored. The stored x, y, z coordinates are displayed in the "**Adjust Labware Position**" Window as shown below.



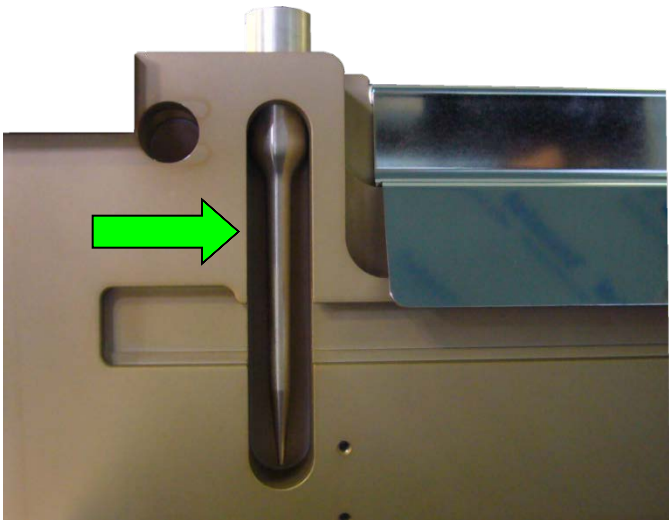
9. Click **[OK]**, to transfer the data to the deck layout.

ATTENTION

Do not use **[Move Probe]** *in the simulation mode. The computer will try to establish a connection to an instrument several times; otherwise, will display a communication error after approximately 10 seconds.*

6.7 Teaching Labware with 5 mL-Pipetting Channels

Teaching Labware with the 5 mL-pipetting channels is very similar to teaching with 1000 µL-pipetting channels. The teaching needle is located on the waste block, in front of the waste bag for tips. Follow the same rules and instructions as provided in Section 6.6 Teaching Labware with the 1000 µL- Channels.



NOTE

The teaching needle is available as an option and is not part of the standard delivery package. It is also used in the maintenance methods.

ATTENTION

Before the pipetting channel with the teaching needle is moving from the Waste Block (picking up position) to the approximate position of the labware, make sure there is no danger of collision with objects higher than 140 mm (above the deck surface):

6.8 Teaching Labware with iSWAP

If the Microlab instrument is configured with an iSWAP, positions of the plates can be "**taught**" for it. In order to teach the position of the iSWAP, the labware which will handle is required.

In the following example, a Micro Plate Carrier with at least one labware position is used.

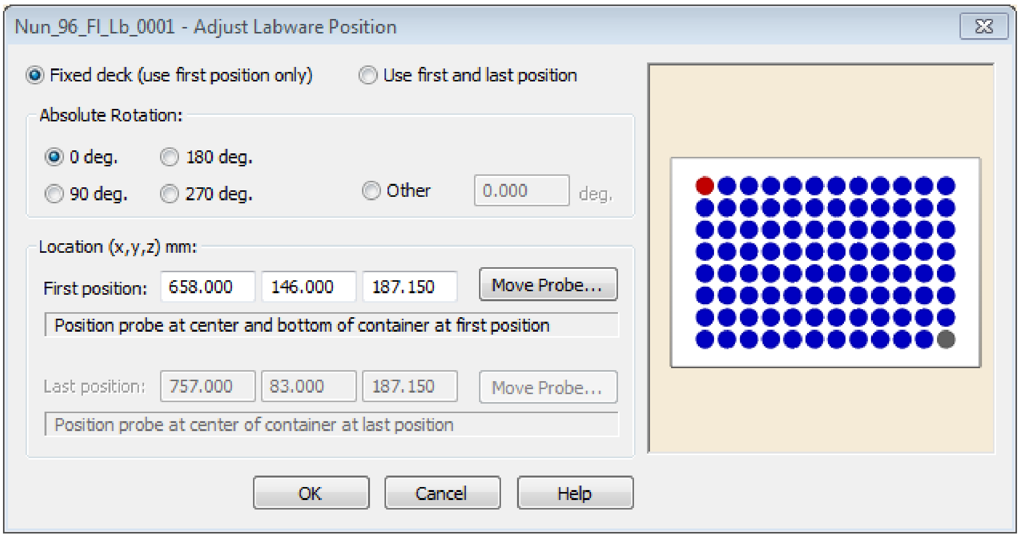
NOTE

For the teaching process, the software requires a defined labware position. For this reason, at least one dummy position for the labware is required.

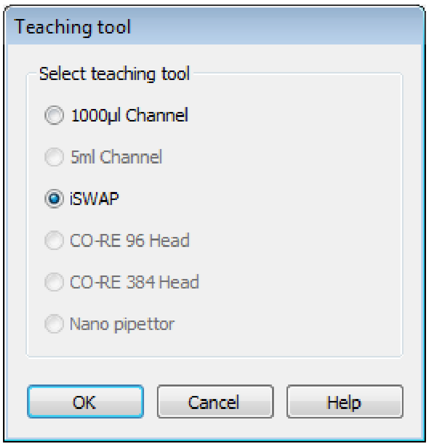
Place the carrier with the labware to the instrument. In the software, add the corresponding labware to the deck layout.

The following steps describe how to teach a position with the iSWAP is performed:

1. If the labware to teach is not on the deck, add it now.
2. Using the "**Drag-and-Drop**" technique, pre-position the labware on the deck.
3. After dropping the labware onto the deck the window, as shown, below will appear.



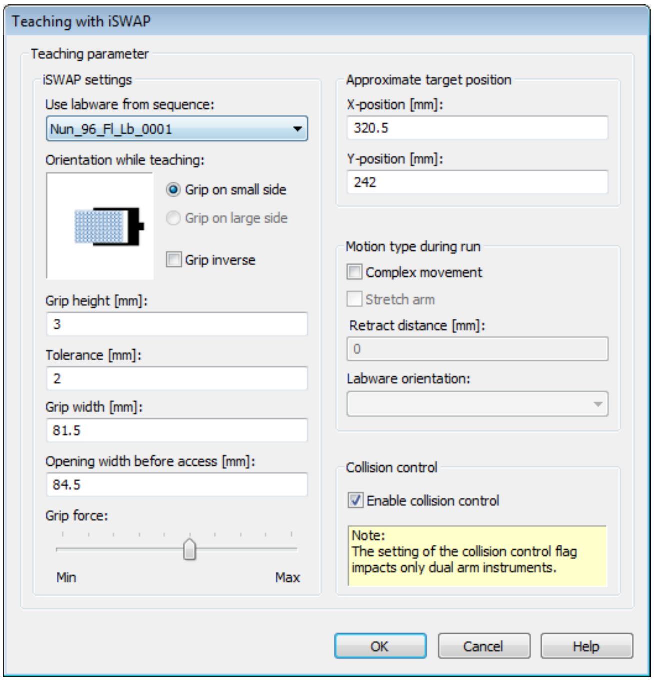
4. Clicking **[Move Probe…]** will prompt the dialog shown below.
5. Select "**iSWAP**" and click **[OK]**



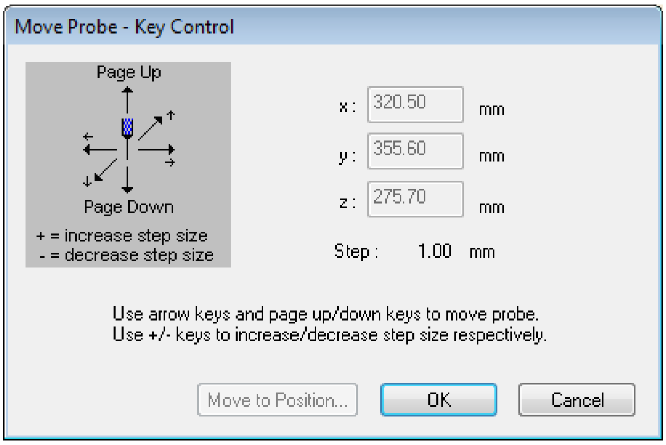
ATTENTION

If the iSWAP is moving with the plate to the approximate position, make sure that there is no danger of collision with other objects higher than 140 mm (above the deck surface).

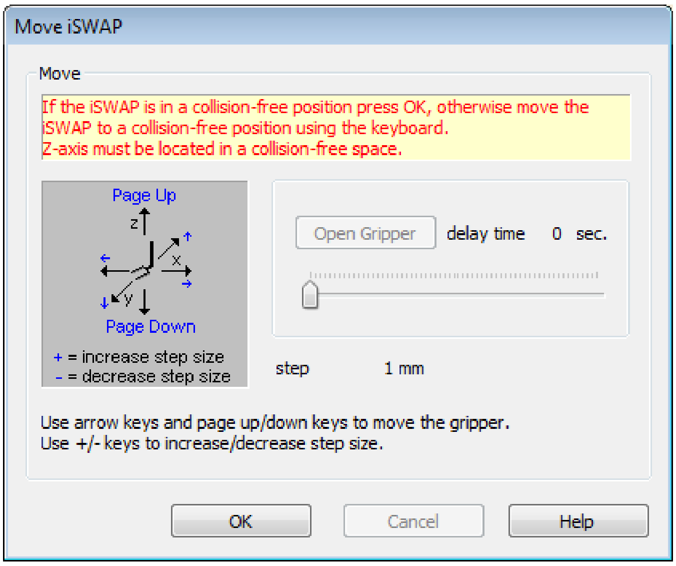
6. Click **[OK]** to proceed. The "**Teaching with iSWAP**" Screen appears.
7. In the "**Use labware from sequence**" drop-down list, enter a sequence of a plate that is already in a valid position. Select the labware that the iSWAP has to pick-up for the next steps of teaching. The default x/y coordinate "**Approximate target position**" is determined by the mouse-cursor when dropping the labware to the deck.
8. The iSWAP picks up the selected labware and moves to the x/y coordinate of the position to be taught.



9. The keyboard (arrow keys, page up, page down) can guide the iSWAP to the labware. The correct position is reached if the labware is touching the bottom of the transfer position. Prevent the iSWAP from crashing.
10. Click **[OK]** to proceed.

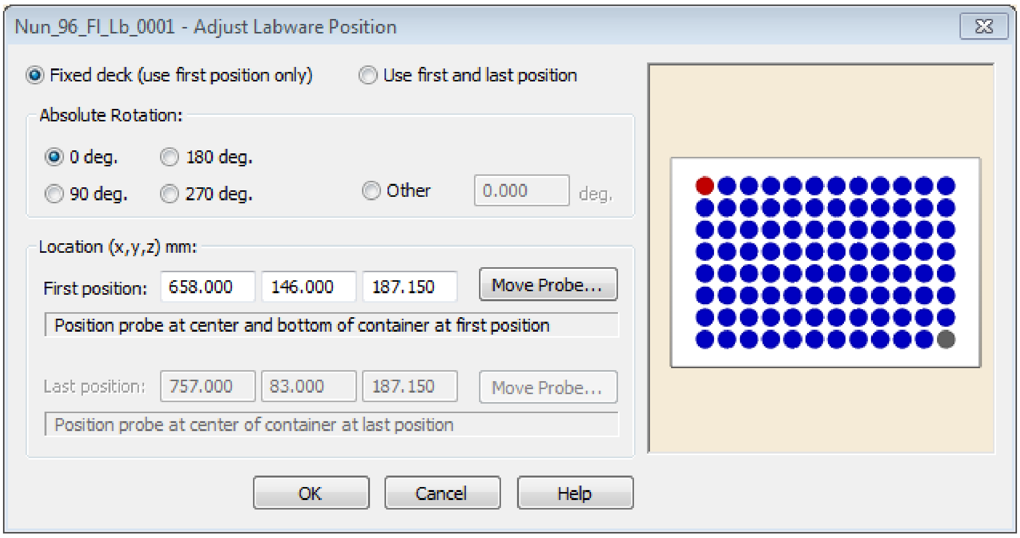


11. When **[Cancel]** is clicked, the iSWAP puts the labware back to the picking-up position and moves to its park position.



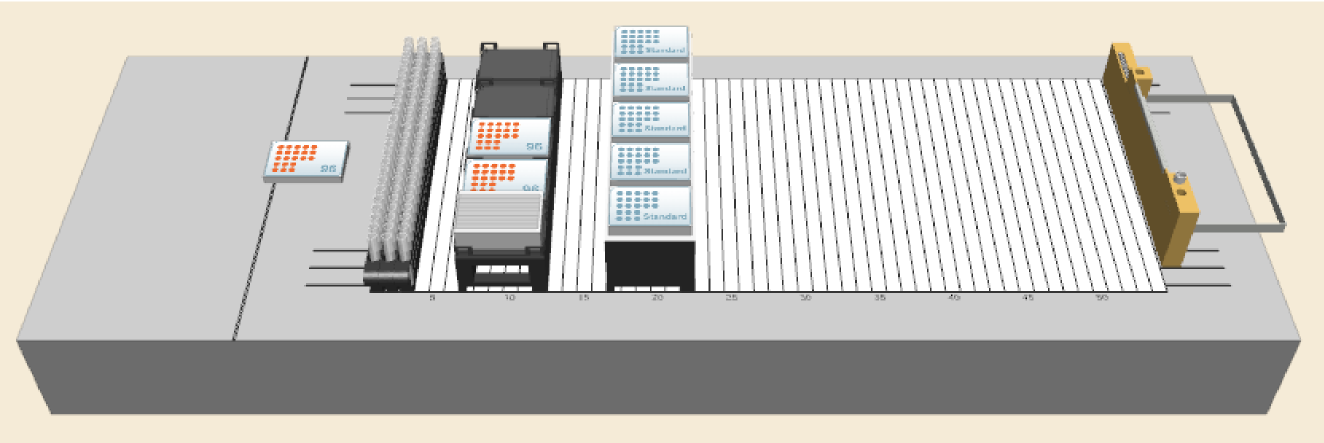
12. The taught coordinates are now stored. If the iSWAP cannot move up to the traverse height (e.g. shelving position where the iSWAP would collide with the shelf above it).

13. Click **[OK]** if the iSWAP is in a collision-free position.



14. The coordinates have been stored. The taught x/y/z coordinates are displayed in the "**Adjust Labware Position**" Window.

15. Press **[OK]** to transfer the data to the deck layout.



16. The iSWAP places the labware back to the pick-up position and returns to its park position.

7 How to Create a Method

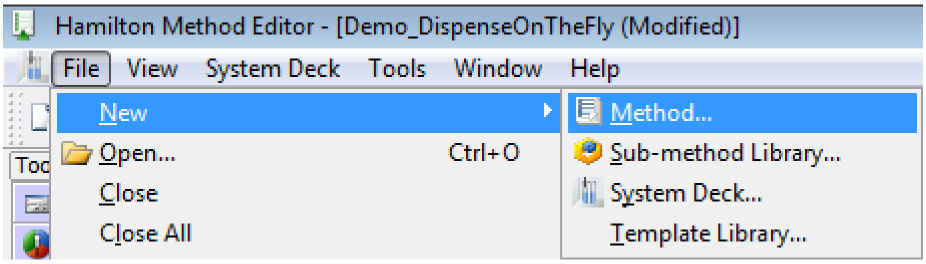
7.1 New Method

Double-click the shortcut icon of the "**Microlab STAR Method Editor**" displayed on the desktop of the PC to start the Method Editor or select "**Start → Programs → HAMILTON → Microlab STAR → Microlab STAR Method Editor** ".

1. To create a new method, click on the toolbar icon " ". A selection dialog will appear.

   or

2. Click on the "**File**" Menu, select "**New → Method**".



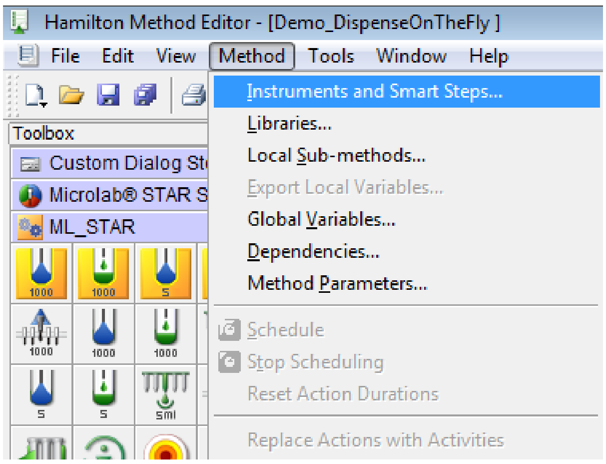
3. Select "**Method**" and click **[OK]**.
4. Enter a file name for the first method and click **[Save]**. The new method is saved.

7.2 Linking a Deck to a Method

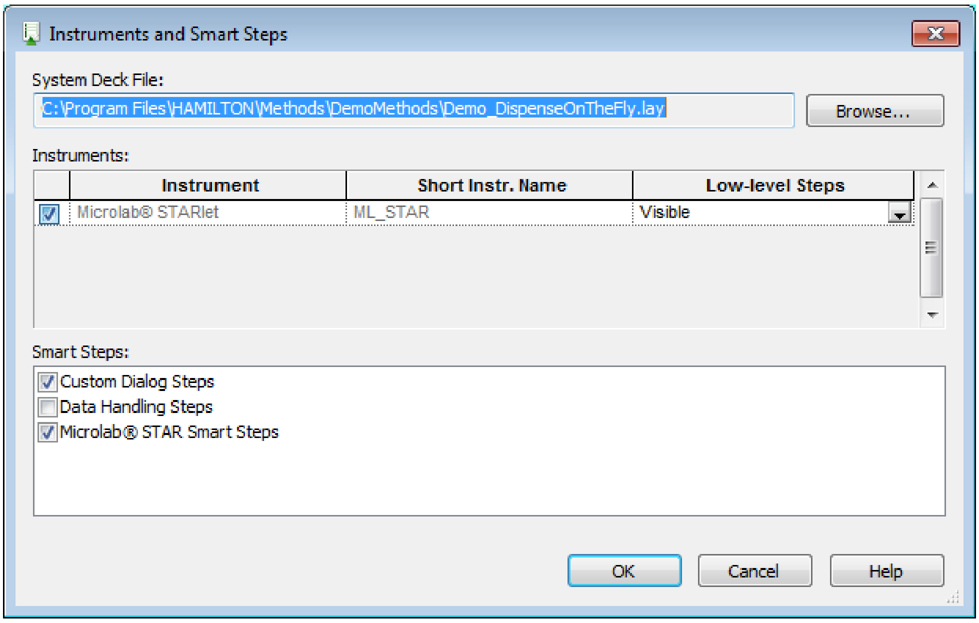
If an existing system deck will be used for a method that will be newly programmed, it has to be linked to the new method. How to create a System Deck is described in Section 5.3 System Deck.

To link the deck layout to the new method, simply follow these steps:

1. Click the "**Method → Instruments and Smart Steps...**" Menu. The "**Instruments**" Dialog will appear.



2. Click **[Browse]**, to browse for the System Deck. A file selection dialog appears.
3. Click in the file dialog to select a deck layout and click **[OK]**. The “**System Deck File:**” and “**Instruments:**” fields in the dialog show the selected file and instrument.
4. Click the checkboxes under “**Smart Steps:**”, to select “**Microlab STAR Smart Steps**”, “**Data Handling Steps**”, or “**Custom Dialog Steps**” (they can also be enabled and disabled later by activating the dialog again). Depending on the selection, more toolbars are added to the toolbox which can be found on the left side of the Editor’s Window.



5. As soon as the method uses instrument-specific steps, the “**Microlab STAR**” Option in the “**Instrument**” Section of the “**Instruments**” Dialog (see image shown above), is locked and can no longer be disabled. The System Deck can no longer be separated from the method until all instrument-specific steps are deleted again. The same applies to the options in the Smart Steps Section.

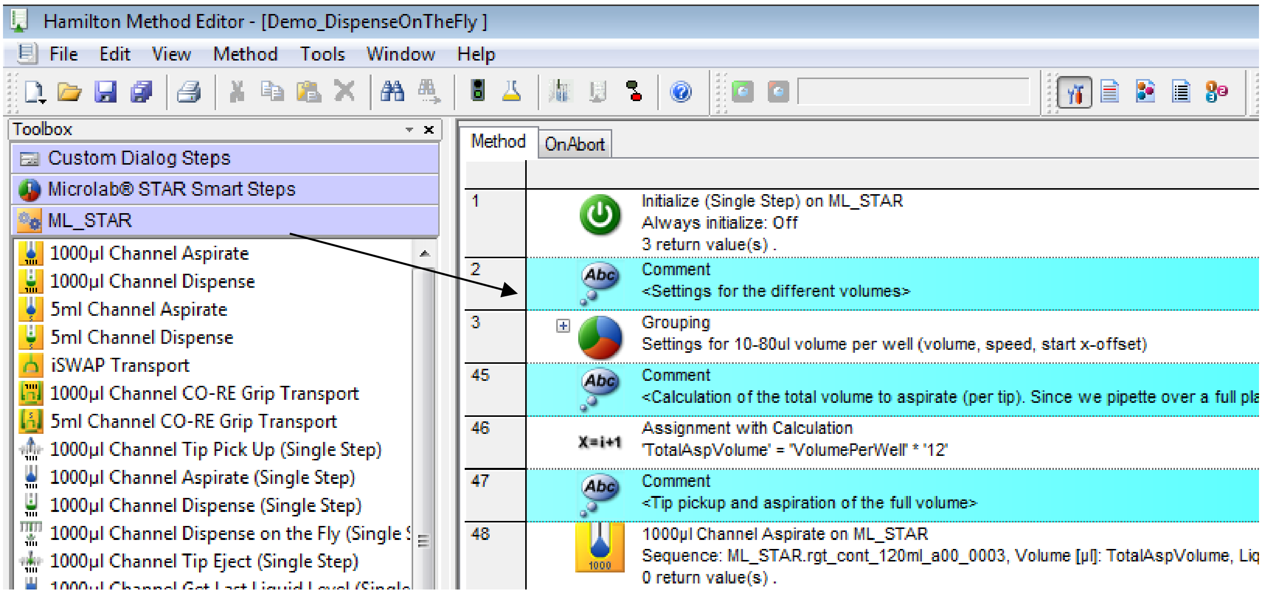
NOTE

Instrument-specific commands are only visible if an instrument is linked to a deck layout.

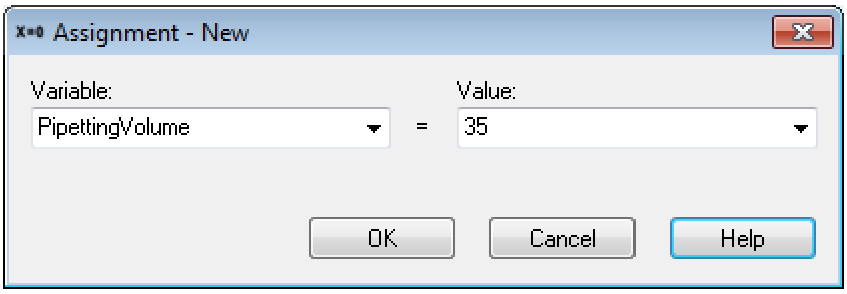
Linking the deck of another instrument to an Microlab instrument will result in an error message.

7.3 Programming

Dragging icons from the toolbox on the left and dropping them into the method window on the right will easily write a method.



Inserting a new command or performing a double-click on an existing command opens the corresponding dialog to edit the command-specific parameters. For example, using the “**Assignment**” Command, which is used to define and assign variables. A dialog box which allows values specification will prompt, as shown in the image below.



Copy - Paste Step

The familiar copy-paste technique can also be used. To apply, simply highlight the step or the block of adjacent steps.

On the “**Edit**” Menu, click “**Copy**” or click **[Ctrl]**-key and perform a left-click “Drag-and-Drop”.

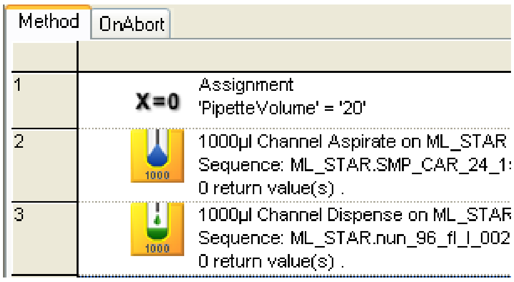
To insert a step from the buffer, click on the line above where the step should be inserted and paste.

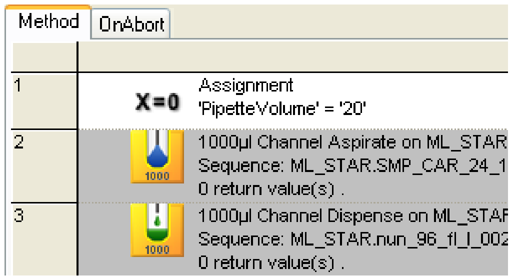
It is also possible to load different methods into the editor and copy steps between methods.

Enable/Disable Step

All steps can be disabled/enabled upon request. Perform a right-click on a step or a selected block of steps and selecting “**disable / enable**” from the Context Menu.

Disabled Steps change their color:



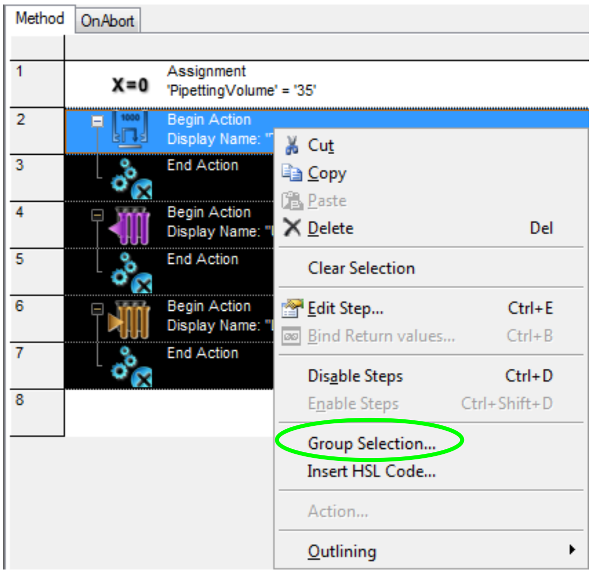


Enabled: white background shows that the step will be executed at runtime.

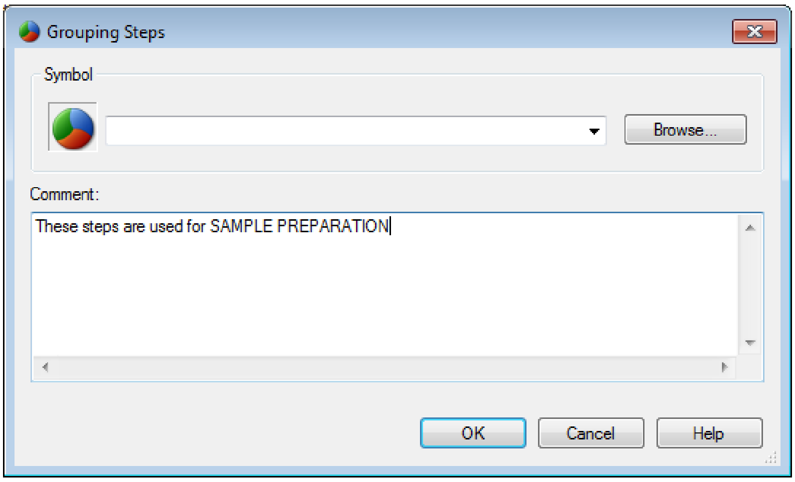
Disabled: dark background indicates that the step is not executed during run.

Group Steps

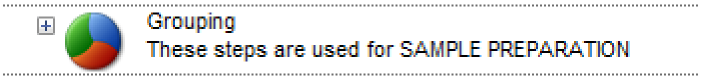
1. To have a structure in the method, it is possible to combine several steps into one group. To do so, select the desired steps and right-click to enter the Context Menu.
2. Select "**Group Selection**".



3. Fill out the “**Grouping Steps**” Dialog. An icon can be selected by using **[Browse…]**. A comment can also be added. Click **[OK]** to confirm.

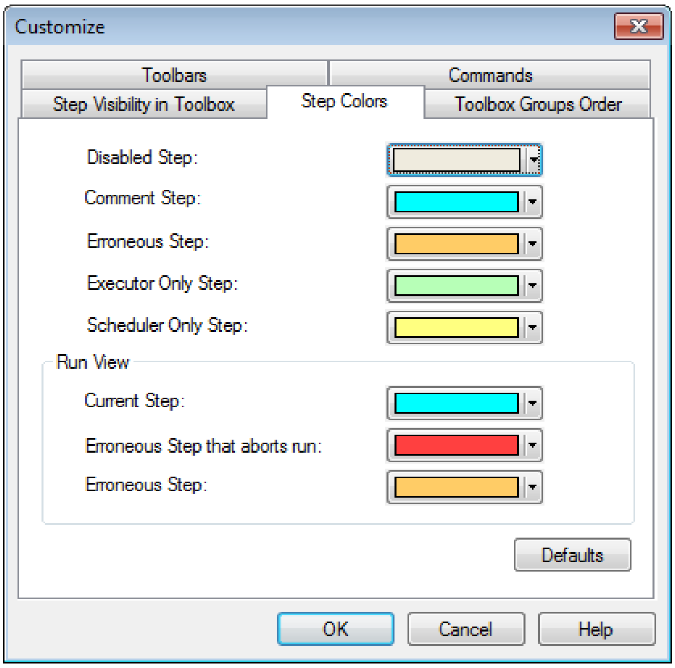


4. The group of steps will then be displayed in a single line. To expand, click on the “**+**” sign found on the upper-left of the line.



Changing the Step Color

The step color can be changed in the “**View → Customize → Step Color**” Menu:



Saving Methods

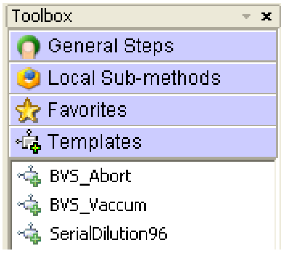
Once at least one change in a method has been made and an exit from the Method Editor is desired, a prompt will ask whether to save the changes or not.

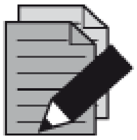
At any time, the work can be saved by selecting "**File – Save**" or simply through the toolbar icon "" for Save and ""for Save All.

7.4 How to use Templates

“**Templates**” offer a framework for commonly used method parts. This shortens the programming time immensely since the programmer does not have to start from scratch. Templates and sub-method libraries within a method are treated similarly.

A “**Templates**” or “**Toolbox Templates**” Tab is available. This is where the existing templates can be found.



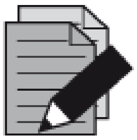


NOTE

Templates are not part of the basic installation of VENUS Software. They have to be added after the installation of VENUS Software.

7.4.1 Including Templates

Templates have to be stored in the folder **C:\Program files**\HAMILTON\Library\Templates (the non-bold sections of the path may differ if a different language is in use).



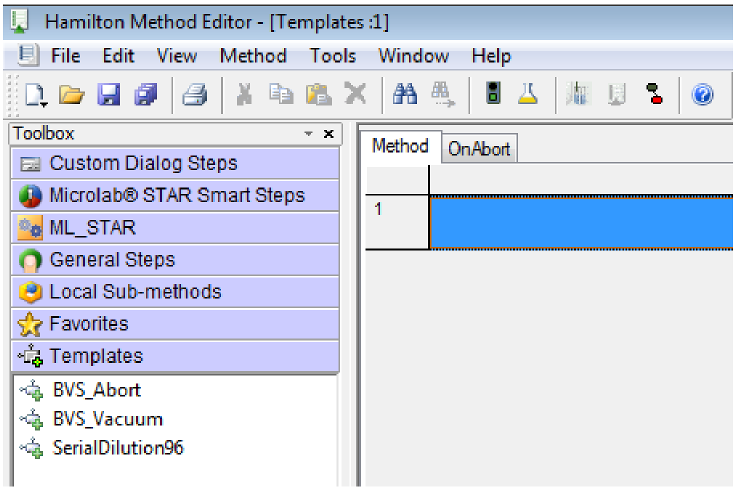
NOTE

For the availability of templates, please consult a local Hamilton Representative.

7.4.2 Using Templates

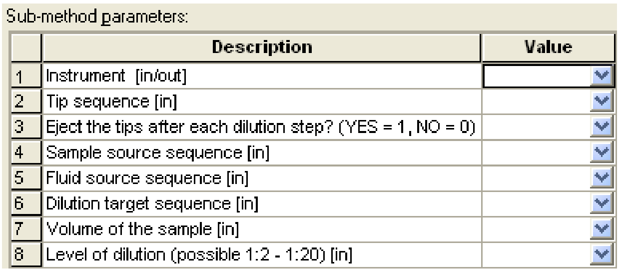
After the templates have been copied to the …/HAMILTON/Library/Templates folder, they will now be visible in the “**Templates**” or “**Toolbox Templates**” Tab found in the Toolbox on the left-side of the window when opening/creating a method.

To use a template, drag it from the toolbox onto the method.



If a template is using parameters, the parameter window opens and can be filled (this is very similar to the use of sub-methods / sub-method libraries). Refer to Section 9.1 Sub-Methods and Sub-Method Libraries.

The example below shows the parameter window of the serial dilution template, where the instrument, sequences, volumes and the concentration have to be defined.


Sub-method parameters:

	Description	Value
1	Instrument [in/out]	
2	Tip sequence [in]	
3	Eject the tips after each dilution step? (YES = 1, NO = 0)	
4	Sample source sequence [in]	
5	Fluid source sequence [in]	
6	Dilution target sequence [in]	
7	Volume of the sample [in]	
8	Level of dilution (possible 1:2 - 1:20) [in]	

After entering all necessary parameters and closing the window, the system will generate a “**Sub-method**” Tab in the method editor (see below) and include the sub-method steps to the new sub-method.

If the template fulfils all requirements, no further changes are necessary. Finish the method programming.

7.4.3 Changing Templates

There are two ways to change a template:

- Change an added template (in a method locally)
- Change the template's source file

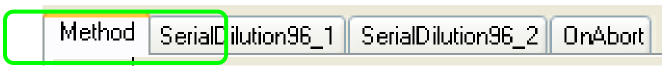
Change an added Template (in a method locally)

If the template's steps need modification, just click on the Template's Tab in the method. This switches to the e.g. SerialDilution96_1 steps. Here, all necessary changes can be made to completely fulfill all the requirements. For example, an aspiration of a sample can be modified with a mix step.

NOTE

Changes made in the method under the Template's Tab are only valid for the particular template.

It is also possible to add the same template several times into the same method. This could be the case when e.g. two serial dilutions with different steps are needed in the same method. The templates include an index as a suffix (see below) as an indicator, so that each template can be adapted according to the customer's needs.



Change a Template's source file

If a template should be changed to reduce the adaption work in every method, open the original Template file in the C:\Program Files\HAMILTON\Library\Templates folder. This folder is where the original Template files are stored. The Templates can be opened and modified from here. After applying the necessary changes, save the new version and open up a method to include the new version.

NOTE

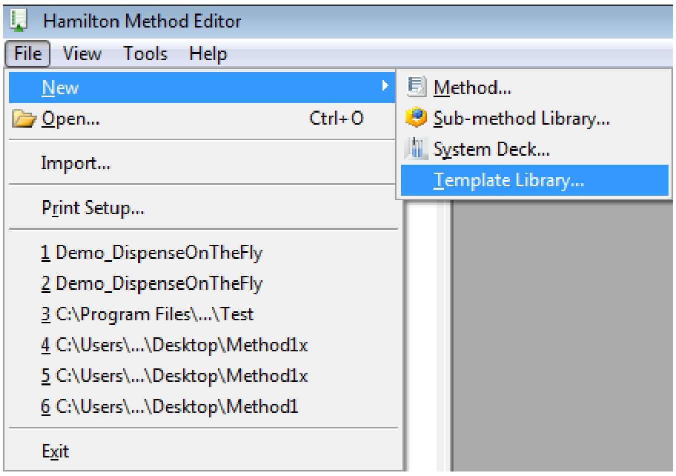
To change a template globally, please open the file located in the C:\Program Files\HAMILTON\Library\Templates folder and change the original template. All following includes will then contain the changes made on the template.

Changing an original template file will have no effect on the previously included templates. Only the newly added templates will contain the changes made in the original file.

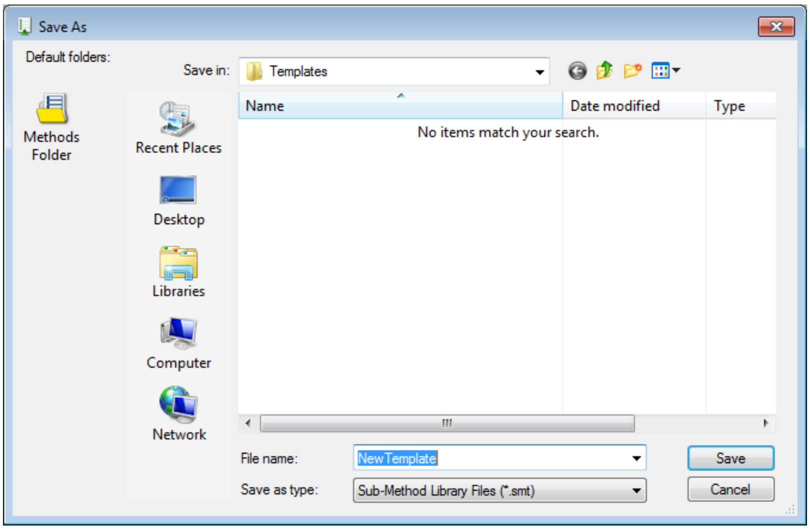
7.5 Programming Templates

There are different ways to create a template.

1. The easiest one is through the “**File → New → Template Library**” Menu.



2. Save the new template with a relevant name. Confirm with **[Save]**.



7.6 How to Edit an Existing Method

To edit an existing method:

1. Use the Toolbar Icon to open an existing method.

   or

2. From the “**File**” Menu, click Open.
3. The “**Open**” Dialog appears where all method files of the current directory will be selectable.
4. Once a selected method is opened, the Toolbox Window contains the "**General Steps**" and "**Local Sub-Methods**", all other instrument-dependent groups that may have been included (e.g. ML_STAR, Microlab STAR Smart Steps) and application-specific library groups (e.g. “**HSLTipCountingLib**” or “**HSLStrLib**”).

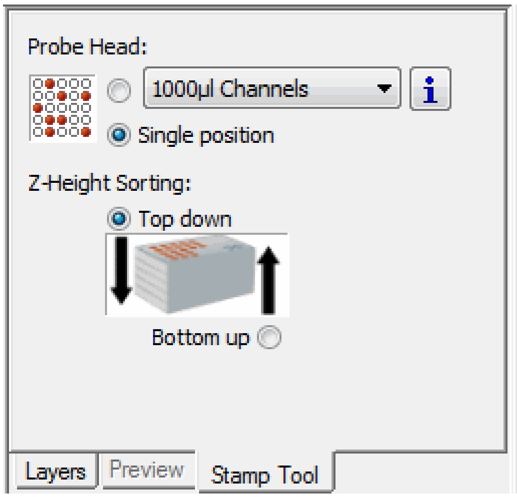
8 How to Create a Sequence

8.1 Sequence Editor

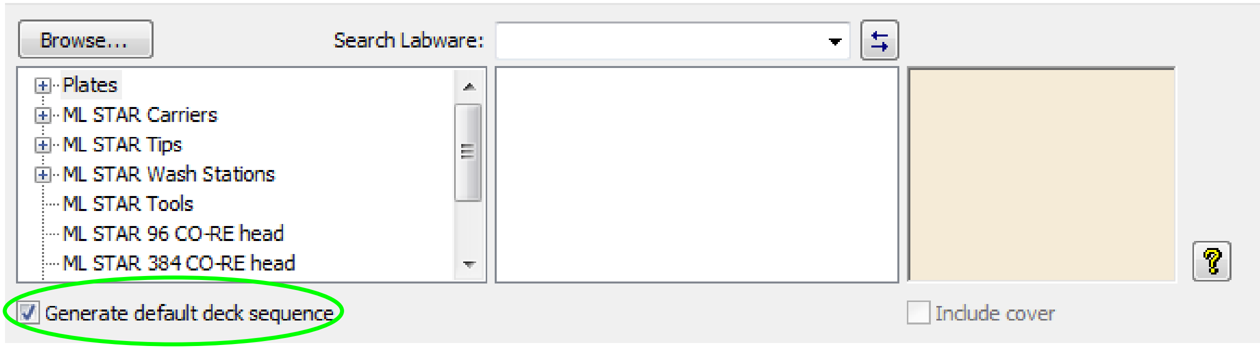
The Sequence Editor is part of the System Deck Editor. It is activated by clicking the "**Sequences**" Tab. With the sequence editor, the programmer declares series of cavities (containers) in an order in which they are processed by the instrument.

Automatic Creation of a Sequence

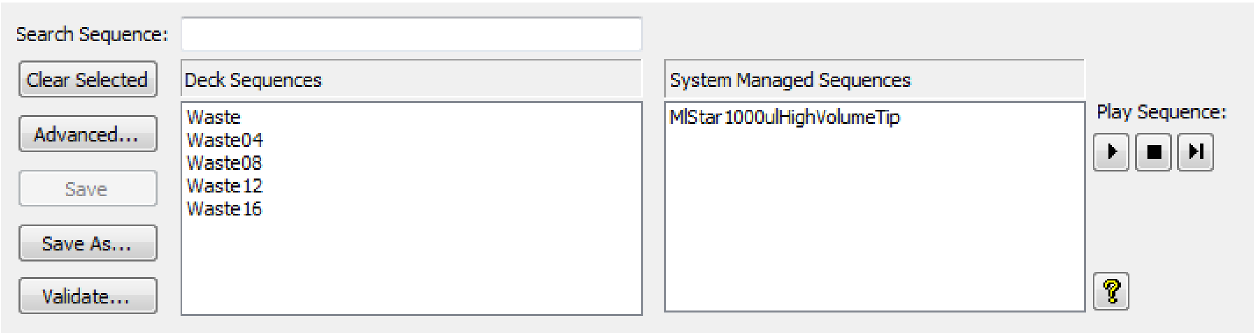
1. If a labware is added to the deck, a default sequence is generated automatically (as long as the 'Generate default deck sequence' checkbox is activated).
2. The system checks which stamp tool is activated. The Active Stamp Tool will be used for the sequence creation in the "**Stamp Tool**" Tab. To change the stamp tool, click the appropriate Probe Head Type.



3. To enable the automatic sequence generation, click on the "**Labware**" Tab, used when adding labware. Tick the "**Generate default deck sequence**" Box as highlighted on the image below.



4. Depending upon the Active Stamp Tool, the automatically-generated deck sequence on the new labware will be created.

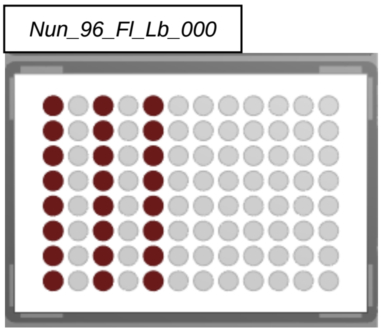


Creating a Sequence Manually

In addition to automatically-created created sequences, the Hamilton VENUS Software also allows the creation of sequences manually. This provides maximum flexibility and allows total control of the pipetting process. As a simple example, pipetting into every odd column of a plate can be done. Instead of jumping around in an automatic sequence by only setting a current and end position, create a sequence that only contains all desired wells for the aspiration.

To create a new sequence, perform the following steps:

1. Open the system deck and select the "**Sequences**" Tab.
2. Click the **[Clear Select]** Button to make sure no sequence is selected.
3. In the system deck window, select by clicking on the wells to add these labware positions to a sequence. The selected positions are highlighted in dark brown as shown below.



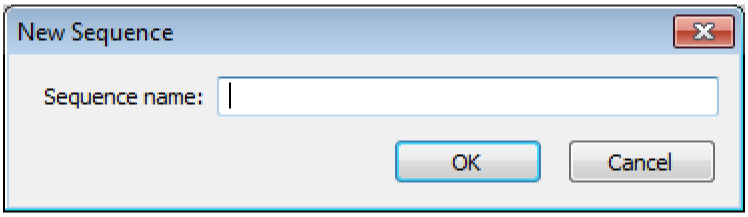
4. Left-click and do not release. Continue by moving over the desired positions, when the positions are covered by the rectangular parameter and release the button.

   It is possible to use the rubber band action to add positions to a sequence. Please notice that the selected stamp tool will be used to sort the sequence when the rubber band action is being used.

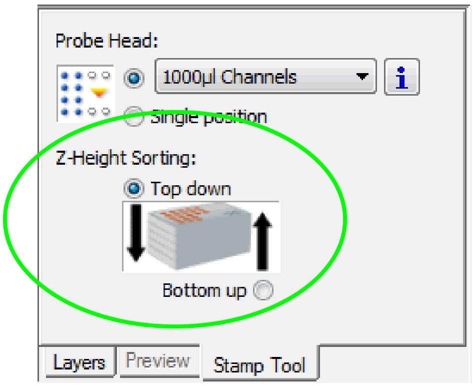
NOTE

To select a labware position, click on it once. If a labware position is clicked the second time, it will be added to the sequence again.

5. Once all the positions belonging to a sequence have been selected, click the **[Save as]** Button found on the upper-left side of the window. A save dialog where the name for the sequence is specified will appear.



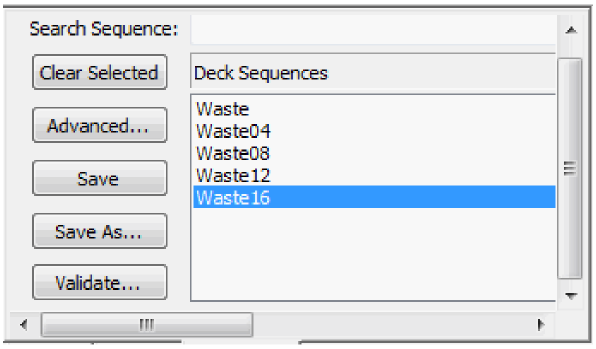
6. Make sure to use relevant names such as "**TargetPlateOddColumns**" instead of "**MySequence1**". Enter a name for the sequence and click **[OK]**. The sequence is shown with the name in the "**Deck sequences**" List at the top-center of the deck layout section, and the color changes from dark brown (draft sequence) to light brown (saved sequence).
7. In creating sequences on stacked labware (e.g. a stack of plates), select the Stack Z-Height Order in the Stamp Tool box first. The sequence will then be sorted top-down (descending) or bottom-up (ascending), depending on the chosen sorting method.



Checks

To envision the created sequence, select the sequence name from the "**Deck sequences**" List. The name of the sequence appears in blue, and the positions belonging to this sequence are highlighted in light brown.

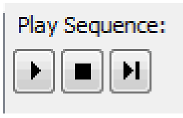
1. To have the sequences checked by the system, click **[Validate]** on the lower-left side. A dialog will appear listing the invalid positions; otherwise, it will state the following message: "**All sequences for this instrument are valid**".



or

2. Use the "**Play Sequence**" Function to see the sequence being processed. The "**Play**" Function is controlled by three buttons above the tabs in the upper-half of the method editor screen. The **[Play]** Button first deselects all positions in the sequence and then plays the sequence. As each position is selected, it is at the same time coated in the deck-layout view and highlighted in the grid and tree views. Play can be paused at any time and can be resumed by clicking on the **[Play]** Button again.

3. Select the sequence to be played in the Deck Sequence list by making use of the buttons below.



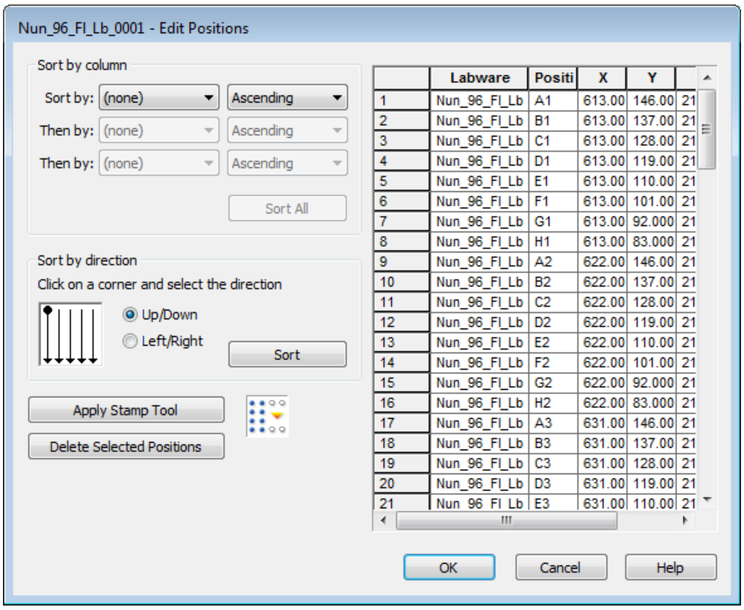
 Play continuously

 Stop

 Play the sequence step by step

Advanced Sequence editing

To see all positions of a sequence in a list, double-click the sequence or select the sequence and click **[Advanced]** found on the left side of the "**Deck sequences**" List window. The position of the selected sequence will be displayed, as shown in the image below.



The grid contains one row for each labware position in the sequence.

Editing positions

The “**Advanced**” Window offers several sorting options.

Sort by column

This allows sorting the sequence by LabwareID, PositionID, x, y and z. It is possible to specify up to three sorting options. Sorting will be executed as soon as the **[Sort All]** Button is clicked.

Sort by direction

This allows sorting the whole sequence from top-down or left-right. Sorting will be executed as soon as the **[Sort]** Button is clicked.

Apply Stamp Tool

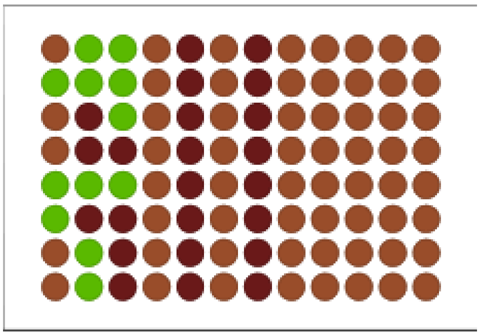
This sorting option refers to the selected stamp tool in the stamp tool panel. When, for example, the “**Head 96**” stamp tool is activated, the sequence will be sorted to process the positions with the Head 96.

Delete Selected Positions is used to delete one or more positions from the sequence. Click on the index (the number on the left side) of the position(s) to delete and click the **[Delete Selected Positions]** Button.

	Labware	Positi	X	Y	Z
1	Nun_96_Fl_Lb	A1	298.00	530.00	187.15
2	Nun_96_Fl_Lb	B1	298.00	521.00	187.15
3	Nun_96_Fl_Lb	C1	298.00	512.00	187.15
4	Nun_96_Fl_Lb	D1	298.00	503.00	187.15
5	Nun_96_Fl_Lb	E1	298.00	494.00	187.15
6	Nun_96_Fl_Lb	F1	298.00	485.00	187.15
7	Nun_96_Fl_Lb	G1	298.00	476.00	187.15
8	Nun_96_Fl_Lb	H1	298.00	467.00	187.15
9	Nun_96_Fl_Lb	A2	307.00	530.00	187.15
10	Nun_96_Fl_Lb	B2	307.00	521.00	187.15
11	Nun_96_Fl_Lb	C2	307.00	512.00	187.15
12	Nun_96_Fl_Lb	D2	307.00	503.00	187.15
13	Nun_96_Fl_Lb	E2	307.00	494.00	187.15
14	Nun_96_Fl_Lb	F2	307.00	485.00	187.15
15	Nun_96_Fl_Lb	G2	307.00	476.00	187.15
16	Nun_96_Fl_Lb	H2	307.00	467.00	187.15
17	Nun_96_Fl_Lb	A3	316.00	530.00	187.15
18	Nun_96_Fl_Lb	B3	316.00	521.00	187.15

The **[CTRL]** Button can be used to select several specific positions in the sequence. All selected sequence positions are highlighted in the deck layout.

	Labware	Positi	X	Y	Z
1	Nun_96_Fl_Lb	A1	118.00	530.00	187.
2	Nun_96_Fl_Lb	B1	118.00	521.00	187.
3	Nun_96_Fl_Lb	C1	118.00	512.00	187.
4	Nun_96_Fl_Lb	D1	118.00	503.00	187.
5	Nun_96_Fl_Lb	E1	118.00	494.00	187.
6	Nun_96_Fl_Lb	F1	118.00	485.00	187.
7	Nun_96_Fl_Lb	G1	118.00	476.00	187.
8	Nun_96_Fl_Lb	H1	118.00	467.00	187.
9	Nun_96_Fl_Lb	A2	127.00	530.00	187.
10	Nun_96_Fl_Lb	B2	127.00	521.00	187.
11	Nun_96_Fl_Lb	C2	127.00	512.00	187.
12	Nun_96_Fl_Lb	D2	127.00	503.00	187.
13	Nun_96_Fl_Lb	E2	127.00	494.00	187.
14	Nun_96_Fl_Lb	F2	127.00	485.00	187.
15	Nun_96_Fl_Lb	G2	127.00	476.00	187.
16	Nun_96_Fl_Lb	H2	127.00	467.00	187.
17	Nun_96_Fl_Lb	A3	136.00	530.00	187.
18	Nun_96_Fl_Lb	B3	136.00	521.00	187.
19	Nun_96_Fl_Lb	C3	136.00	512.00	187.
20	Nun_96_Fl_Lb	D3	136.00	503.00	187.
21	Nun_96_Fl_Lb	E3	136.00	494.00	187.



Saving the Sequence

Once the labware positions in the sequence are in the desired order, save the sequence by clicking the **[Save as]** Button.

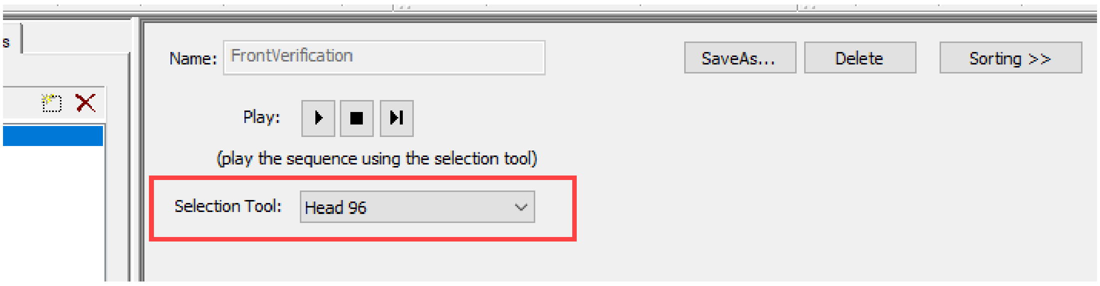
Saving affixes the order of each row in the grid as the order of the labware positions in the sequence. Upon saving, a prompt will request a name for the sequence.

Any existing sequence in the Deck Layout can be activated for viewing and/or editing by selecting the sequence name in the drop-down menu.

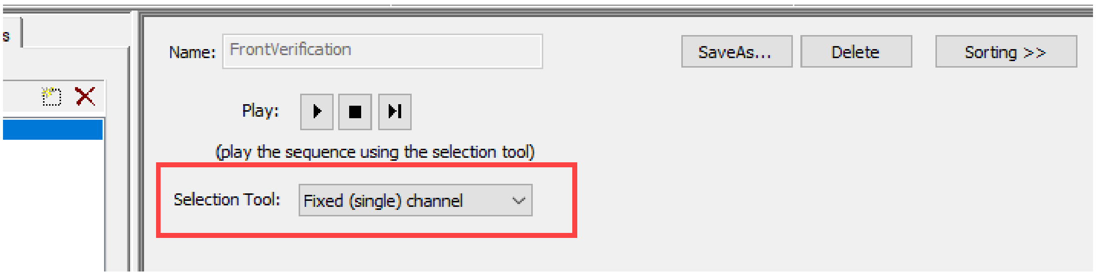
Additional labware positions can be inserted in the active sequence by selecting them in the Deck Layout View. Existing positions can be removed. Positions can be reordered as described above. The sequence can be resaved to fix its positions, according to the new order.

8.2 Stamp Tool

The "**Stamp Tool**" Function allows for applying the sequence positions to a labware depending on the pipetting tool being used: 1 to 16-Pipetting Channels, CO-RE 96 Probe Head, CO-RE 384 Probe Head, CO-RE 384 Probe Head STP (see Section 12.8 Method using the CO-RE 384 Probe Head) and Nano Pipettor. The order is applied during the adding of labware onto the deck or when using the rubber band function to create new sequences.



The "**Fixed (single) channel**" Selection adds 1 position to the new sequence wherever the cursor is positioned and ticked (on a container).



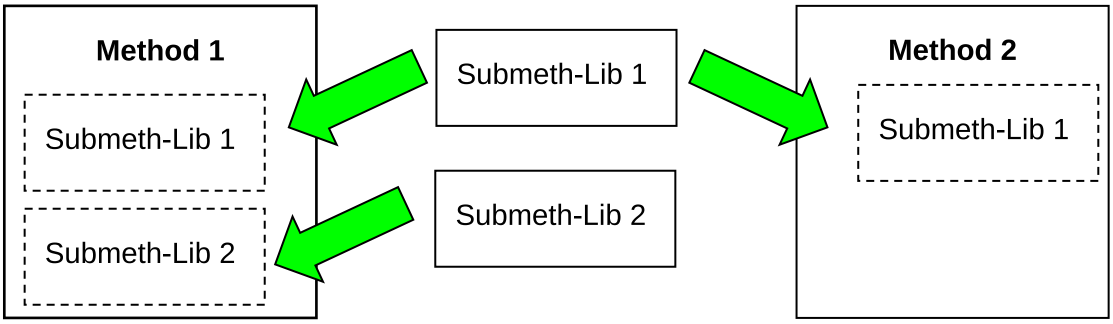
9 How to Create Sub-Methods

9.1 Sub-Methods and Sub-Method Libraries

Parts of the existing methods can usually be re-used at a later time in the same method or in other methods. The VENUS Software enables grouping of useful command routines that are called sub-methods.

A local sub-method is part of a particular method and remains linked to it. It is displayed along with the method it belongs to.

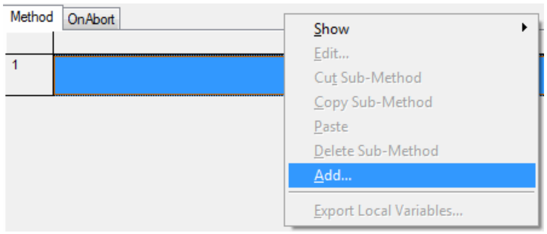
A **sub-method library** contains an independent sub-method that may be used in several independent methods. To make use of the advantages of sub-methods, the functions of the "**Sub-method**" Library must be included in a method.



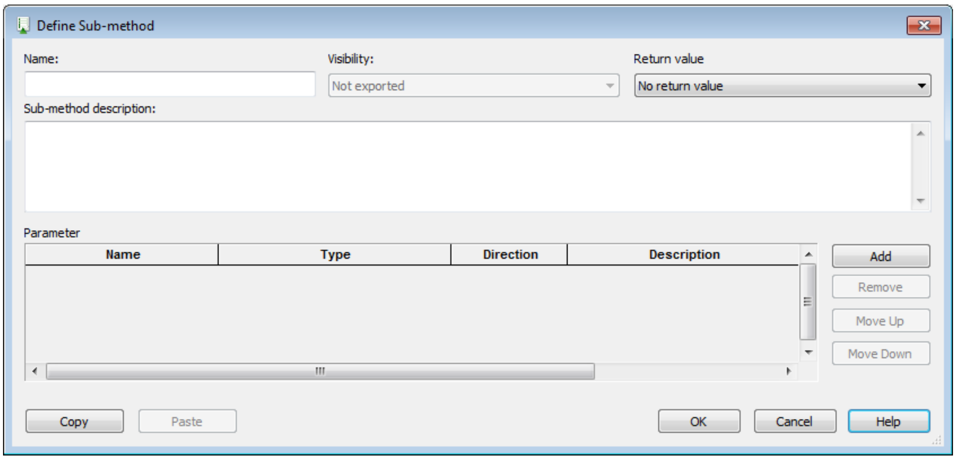
Although sub-methods and sub-method libraries are different in terms of scope, the programming is very similar.

9.2 How to Create Sub-Methods

1. In a method, right-click in the top area beside the "**Main Method**" Tab / "**OnAbort**" Tab to open the Context Menu then click the **[Add...]** Button to open a new dialog box where the interface of a sub-method can be defined.



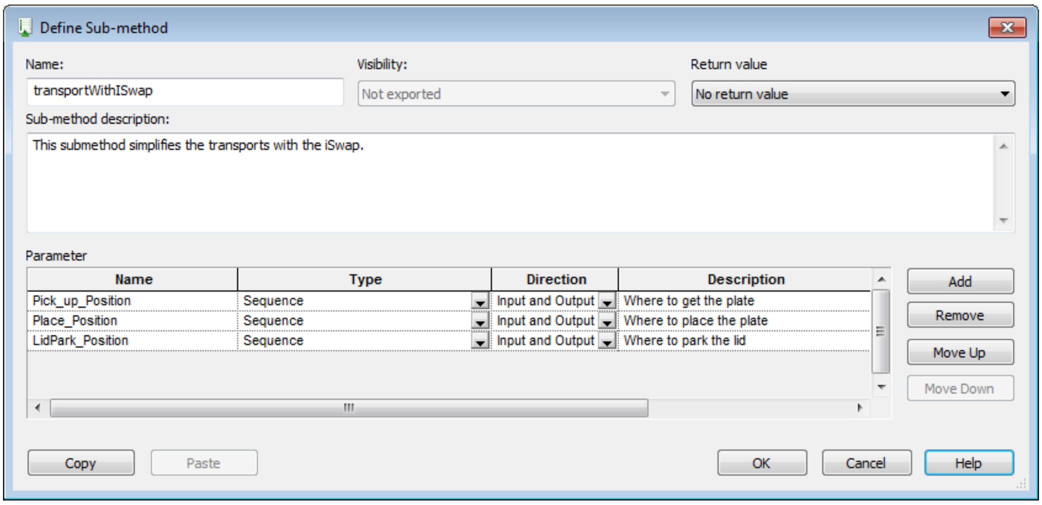
2. The "**Define Sub-Method**" Dialog, as shown below, will appear.



3. Specify a sub-method name.

   The "**Visibility**" Option can only be used in sub-method libraries. Such collection of steps may itself contain functions/procedures which will not be exported. In this case, they are of local nature and hence are not visible by the user of the library. Local sub-methods will always be visible in the method within which they are defined.

4. Enter a "**Sub-method description**" to briefly describe its use. This entry is optional but helpful to other programmers who may access the method. Note that quotation marks are not required and should not be used in this field.



One or more parameters can be assigned to a sub-method; to do this, perform the following steps:

5. Click **[Add]** for each parameter to be assigned.
6. For each parameter, specify:
    - A name (which will be referenced throughout the sub-method),
    - A type (variable, sequence, etc.) selected from the list in the drop-down list available,
    - A direction (input, output or both)
        - ⇨ "**Direction**" refers to whether the variable used to call the sub-method in the main method may be altered by the sub-method or not.
        - ⇨ The "**Input**" Option means that the original variable is not to be altered.
        - ⇨ The "**Output**" Option means that the original variable will be overwritten by the sub-method, and the value of the original variable cannot be used inside the sub-method.
        - ⇨ The "**Both**" Option (both input and output) means that the original variable will be altered by the sub-method.
7. A short description of the parameter (without quotation marks).

Example:

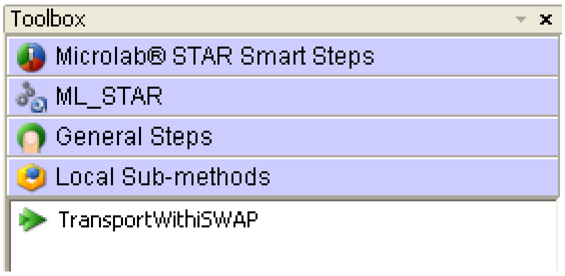
If the main method contains a variable par = 10 and the sub-method contains the operation par = par + 5:

- The "**Input**" yields the result 10 (the value of the parameter is copied from the main method and overwrites the value in the sub-method).
- The "**Output**" yields 5 (no value is read from the main method).
- Then "**Both**" yield 15 (the value of the parameter is copied from the main method and is added to the operation in the sub-method).

To remove a parameter, click one of its description fields, and then click **[Remove]**.

To reorder parameters, select one and click **[Move Up]** or **[Move Down]**.

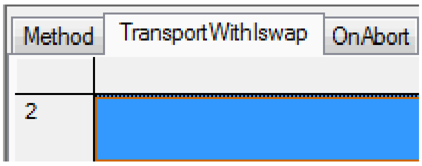
8. Click **[OK]** to store the sub-method interface. The system returns to the previous window, and the new sub-method is included in the "**Local Sub-methods**" Tab found in the toolbox window.



9. Additional sub-methods will be added to the "**Local Sub-methods**" Tab. The main method itself is always on the first tab.

Modifying a Sub-Method

To modify an existing sub-method (e.g., by adding or deleting steps), click on its tab in the method window and it will open for editing.



Variables in Sub-Methods

Just like in a method, variables can be created in a sub-method. However, unlike those in a method, sub-method variables remain local (see Section 11.1 Variables and Return Values).

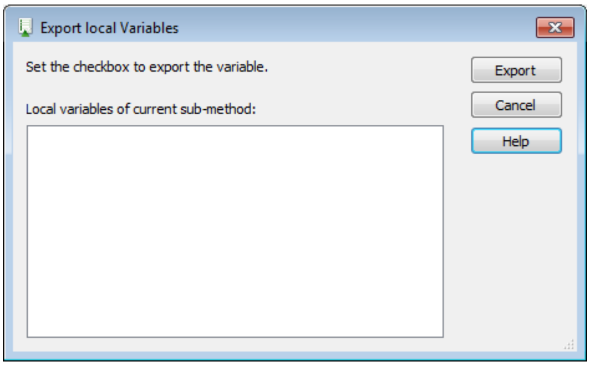
Promoting Variables to Global Status

A local variable defined inside a sub-method can be promoted to global status if it has to be used elsewhere.

From the Menu, select "**Methods**" and choose "**Export Local Variables…**". A dialog will appear, containing a list box that displays the local variables in the current sub-method.

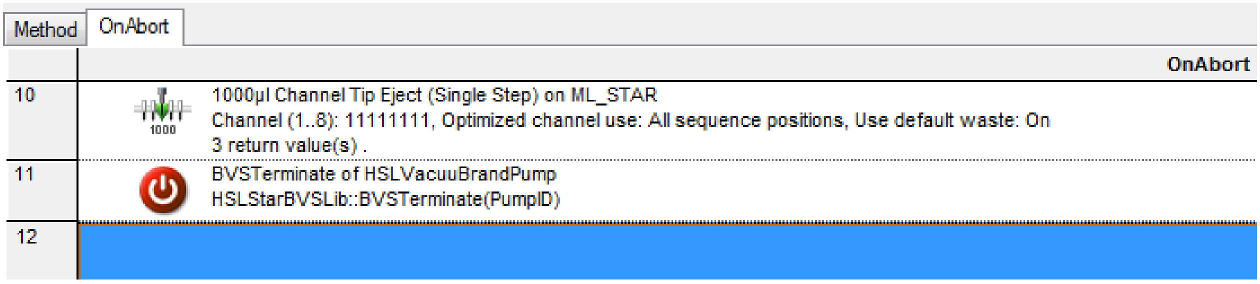
1. Tick the box of all variables that will be promoted to global status.

2. Click **[Export]** to promote the variables globally.



9.3 Sub-Method OnAbort

The "**OnAbort**" Sub-method appears in every method. This sub-method is used to execute steps in case of an (instrument- or user-caused) abort. It can contain indicated to do instructions when the method is aborted. If a method finishes successfully, the steps in the OnAbort sub-method are not executed.

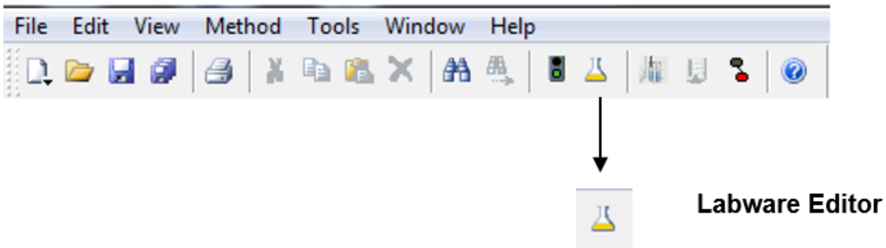


10 How to Create Labware Files

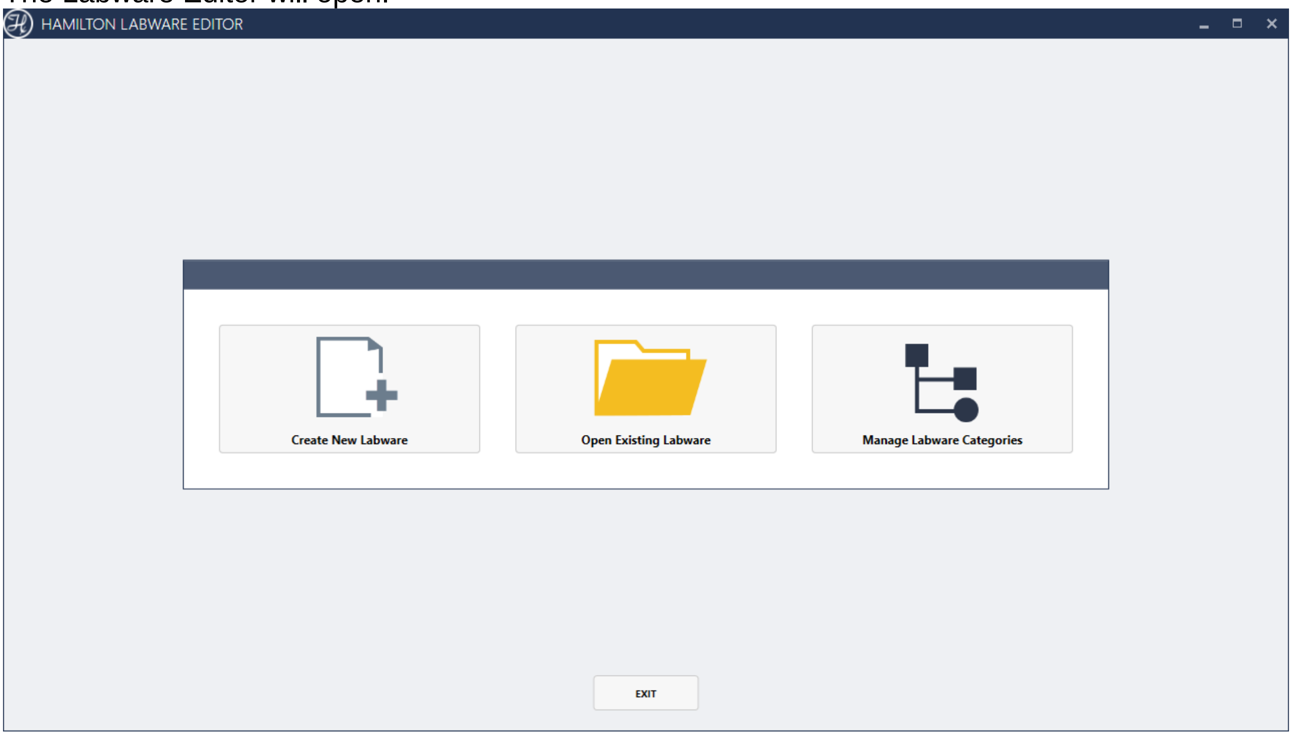
If the labware is not pre-defined, custom racks and containers can be defined using the Labware Editor. The custom labware can be used like any other pre-defined labware object from the library. A labware object is a representation of the real physical labware.

To define the new labware:

1. Select "**Tools → Labware → Labware Editor …**" in the Method Editor or the "**Define Labware**" Icon found on the toolbar:



2. The Labware Editor will open.



3. To create a new labware, select "**Create New Labware**" from the Labware Editor main screen. This will display different choices for different types of labware. This will be explained in the following sub-sections.


Racks & Tubes
Container (Plate Well / Tube)
Microtiter Plate
Regular Rectangular Rack
Irregular Rectangular Rack
Circular Rack
Carriers, Pedestals, & Templates
Custom Template
Nimbus
STAR

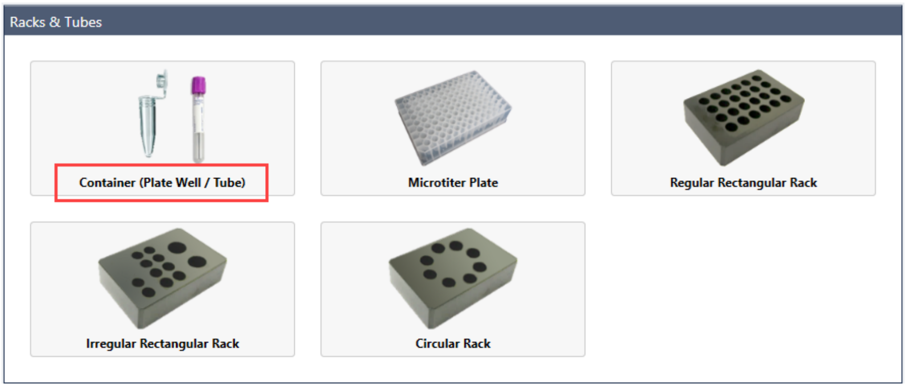
10.1 Defining a Labware

The following section illustrates the procedures for defining a labware, using the example of a rectangular rack. The first step of each labware definition is to describe the geometry of a single container. The second step is by putting several containers together which creates an overall labware (e.g. tube).

10.1.1 Defining a Container

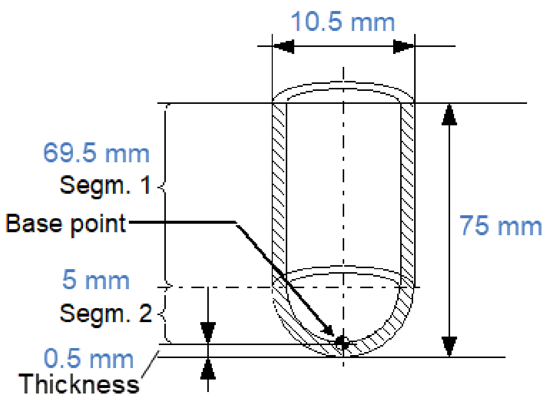
A completed container can then be used in a rack definition.

1. Select "**Create New Labware"→ Container**".

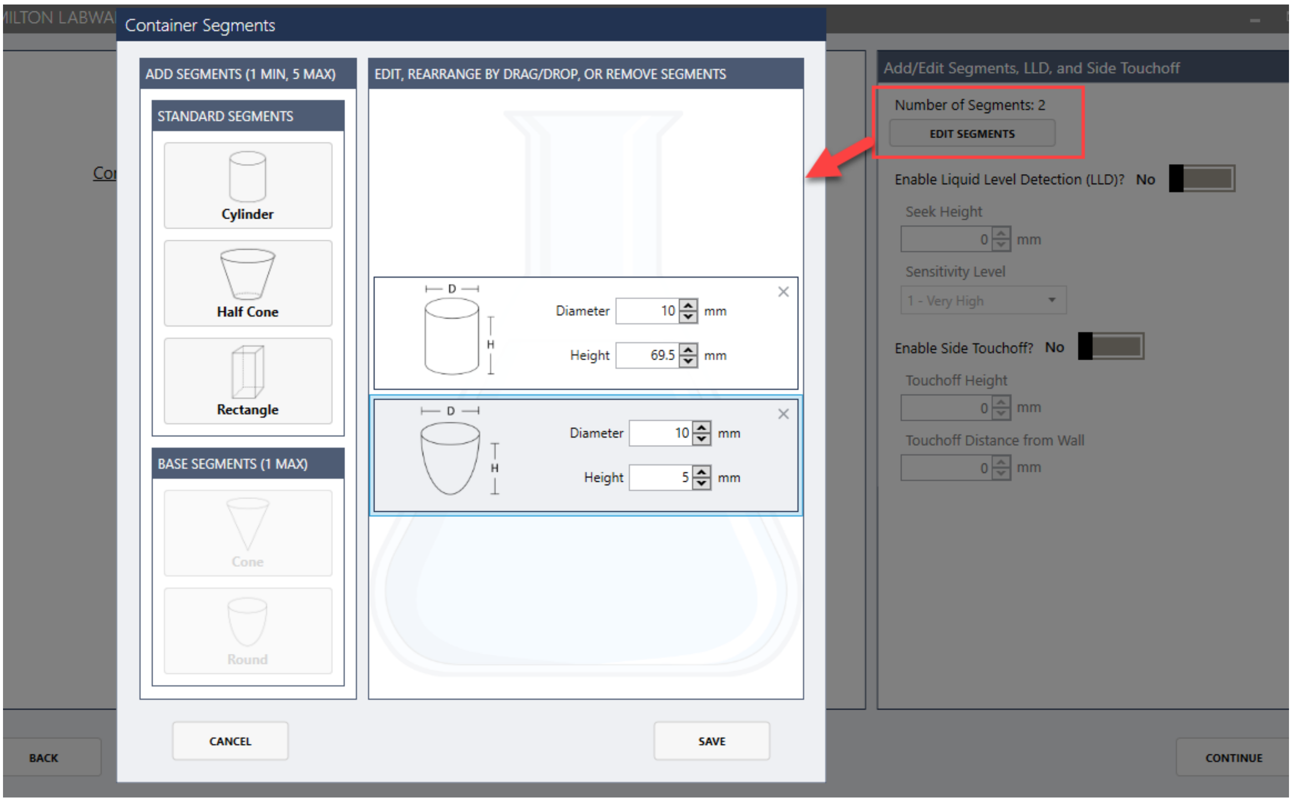


2. Define the container as a round-bottomed tube with an outer diameter of 10.5 mm, with a total container length of 75 mm and a material thickness of 0.5 mm.

   Remember that the Base Point for all further references is the inside bottom point.



3. Click "**Add Segments**" to define the 2 segments that the tube has: a cylindrical segment and a round-bottomed base segment.

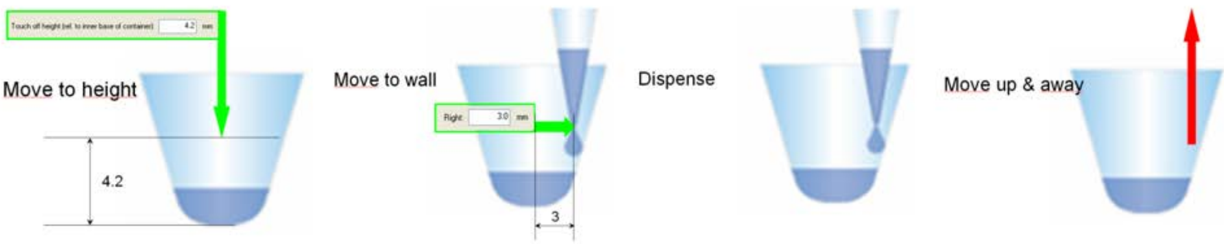


4. Enable Liquid Level Detection and specify the **Seek Height** at 70 mm. This is the height in mm above the inner bottom of the container at which the probe will start searching for liquid.

   A warning will be shown at runtime if liquid is detected above the seek height.

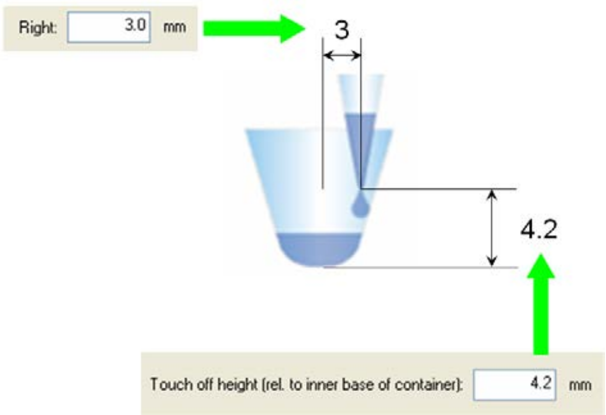
5. Set the detection **Sensitivity Level.** Use higher sensitivity for smaller containers.
6. If a side touch dispensing on the container is not wanted, leave the "**Side Touch-off**" disabled. A **Side Touch Dispense** is a special dispense mode to prevent droplets on the tips end.

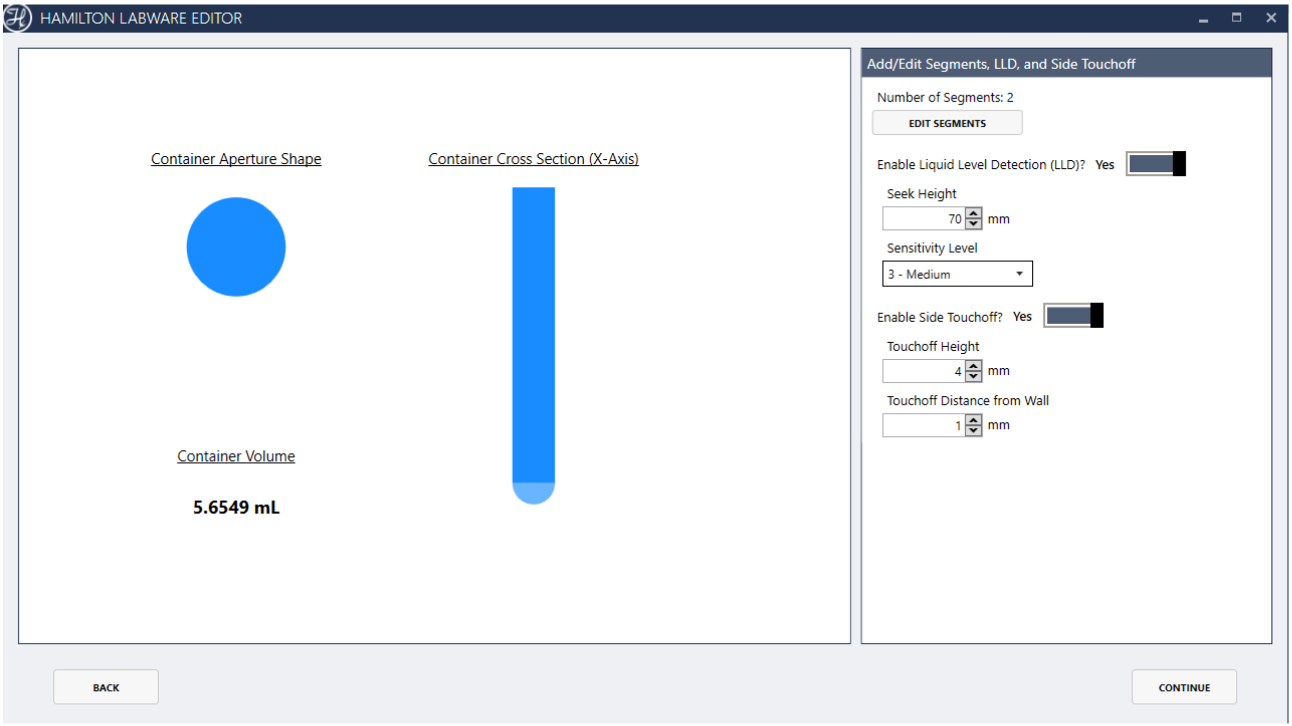
   The tip will move in the center of the container to the specific touch off height. From there, a Right move is performed. Then, the liquid will be dispensed and the tip or needle moves up and away.



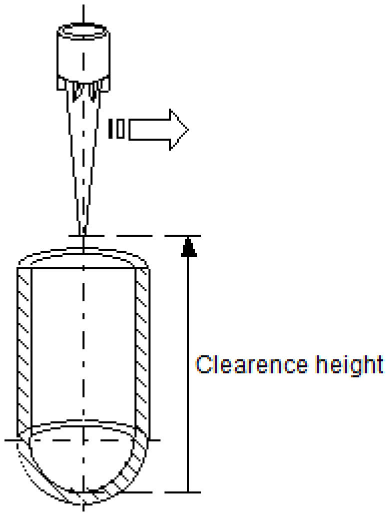
7. To perform a side touch dispense with the Microlab instrument, the values for the touch off height and the right move must be specified. Enable "**Side Touch-off**" to enter the desired values.

   Either calculate or measure these values before testing on a real instrument. An example on how to measure the values is being illustrated below.

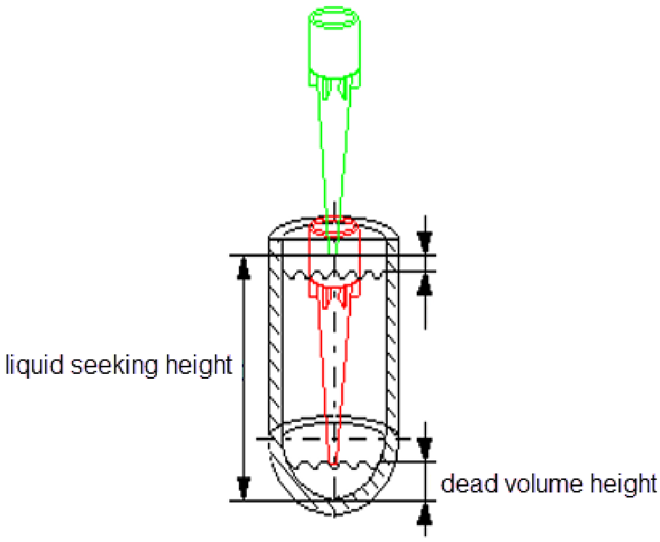




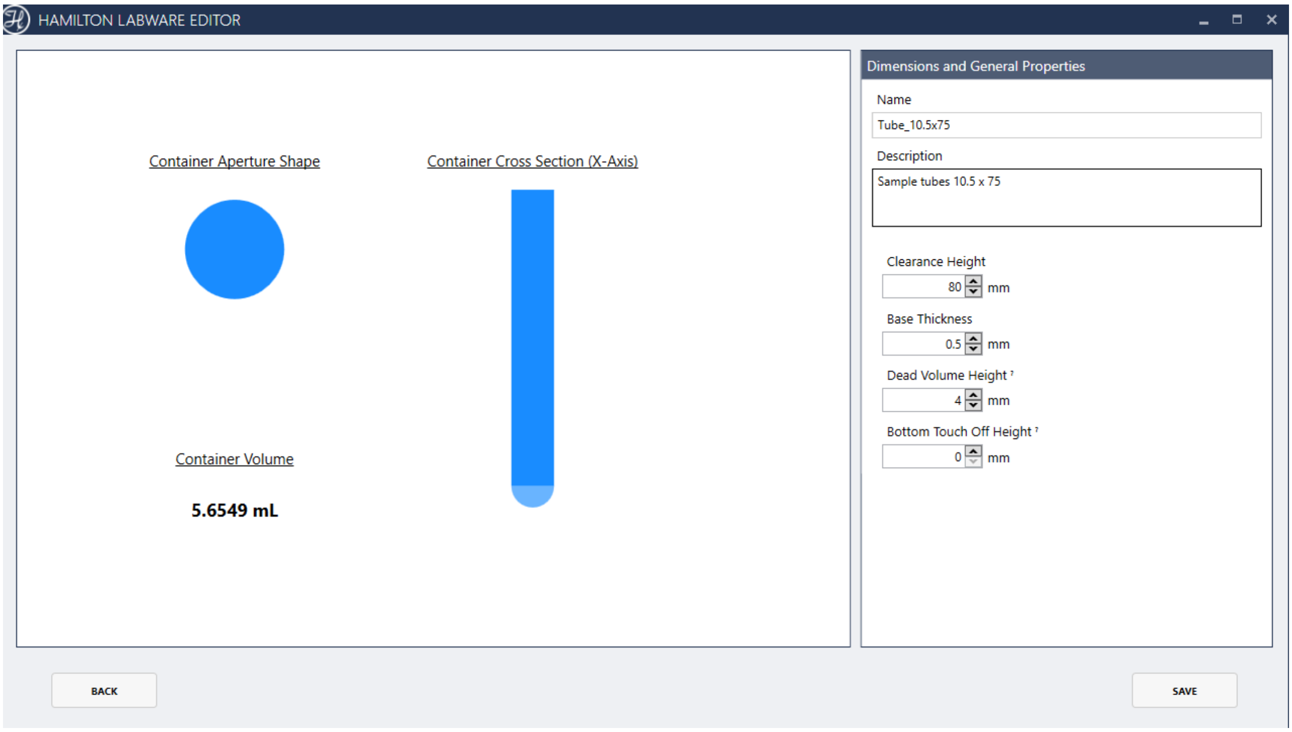
8. Click Continue.
9. Set the **clearance height** at 80 mm (75 mm for the container length plus additional 5 mm for the traveling); measured from the container inner base (Base Point). The clearance height is the height at which the pipetting arm can pass over the container without touching it.



10. Set the **dead volume height** (minimum height above tube bottom) counted from the container bottom to 4 mm. This restricts the tip to not go down to a position lower than 4 mm above the tube bottom (this gives the "**Dead Volume**", the volume that cannot be reliably aspirated). The default value of 0 mm allows the tip to move to the very bottom of the container.



11. Set the "**touch-off at bottom height**" to 0 mm.
12. "**Touch-off at bottom height**" is the position of the tip when dispensing with "**touch off**" into an empty container.

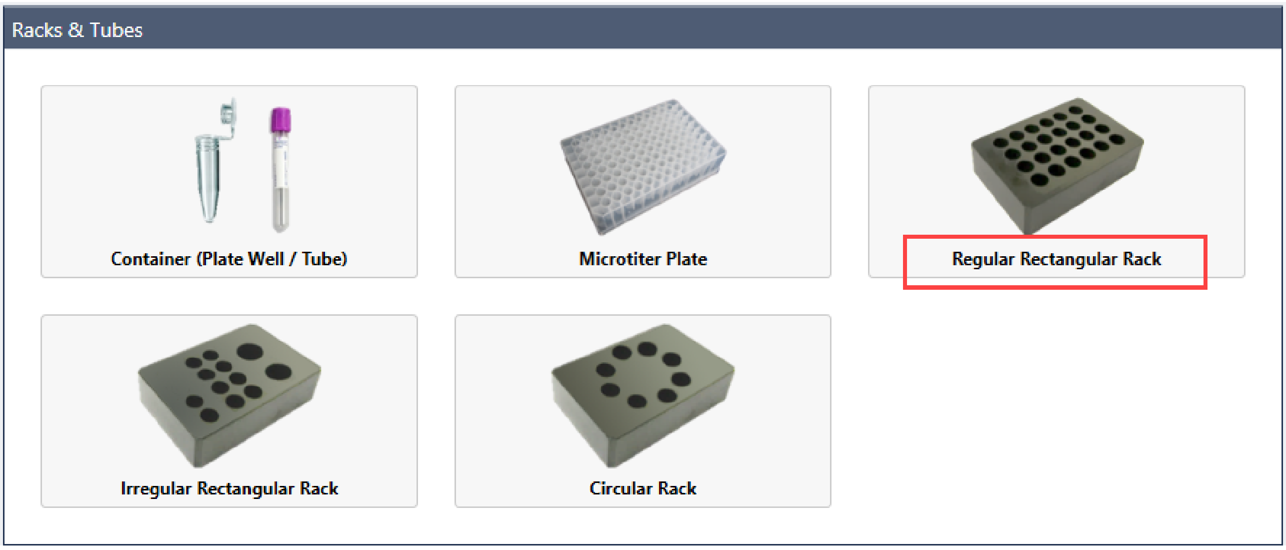


13. Enter the name and description.
14. Click Save and choose a destination folder for the container (.ctr) definition file.

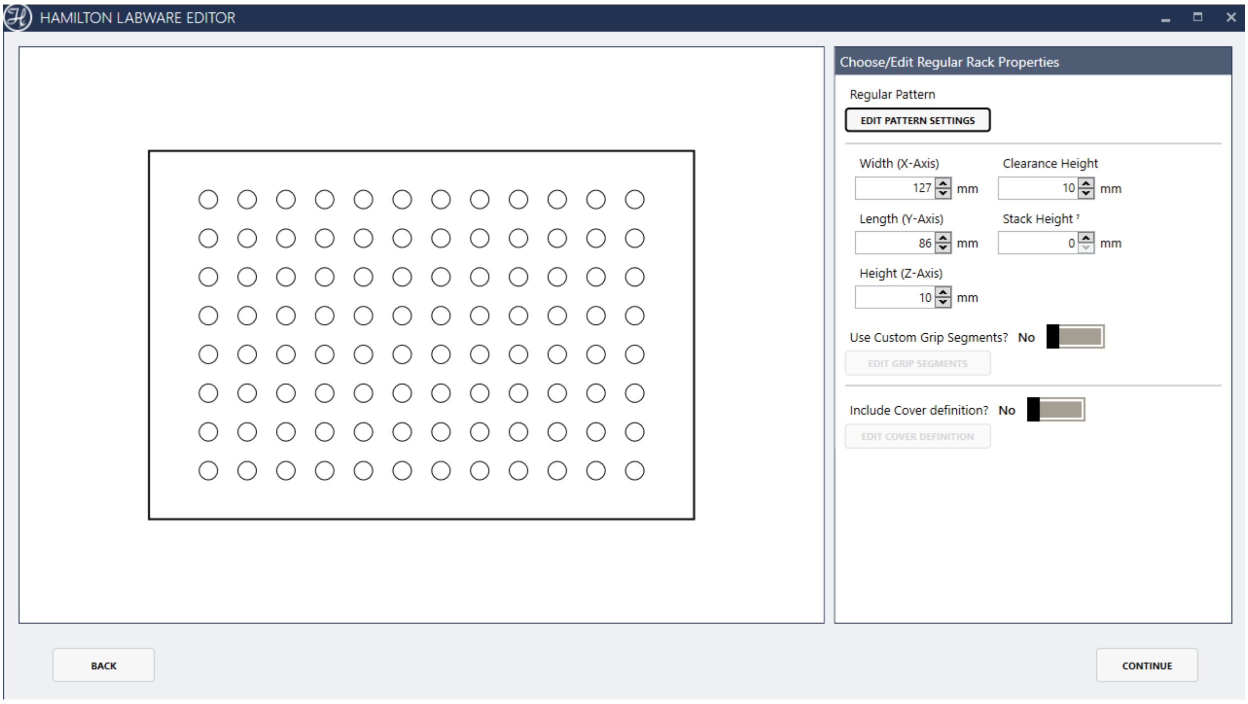
10.1.2 Defining a Rectangular Rack

For ease of use, in defining a Rack for the Containers, ensure that all Racks and Containers defined have clear and distinct names.

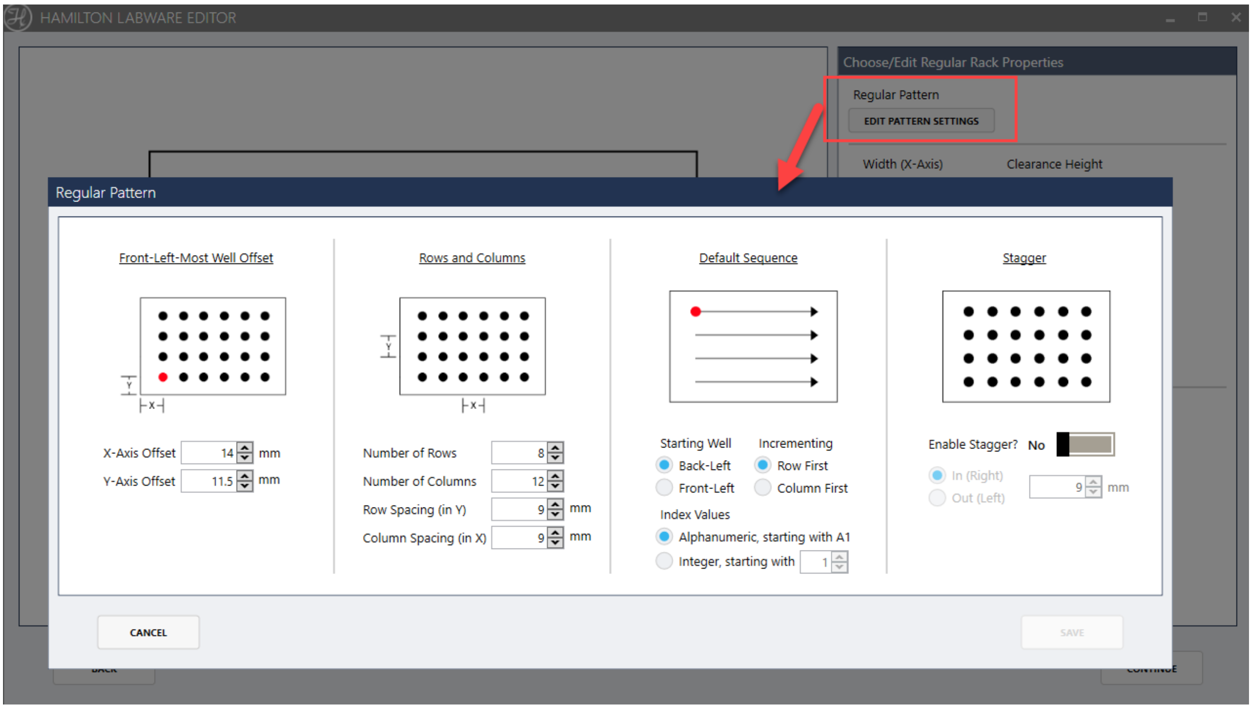
1. Select "**Create New Labware → Rectangular Rack**"
   A series of screens will appear, allowing the programmer to design the Rack.



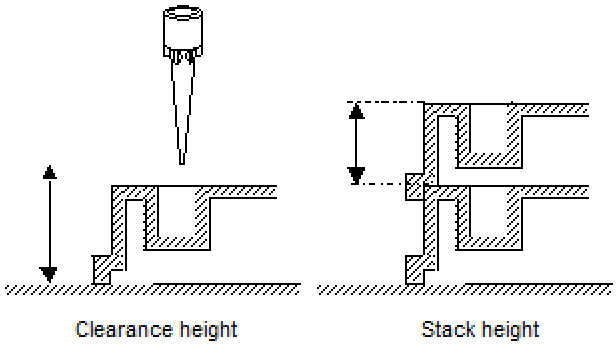
- Always choose "**Regular rectangular rack**" to define or modify rectangular Racks and Micro Plates. Choosing "**Microtiter plate**" requests only a subset of the information relevant for the Microlab instrument.



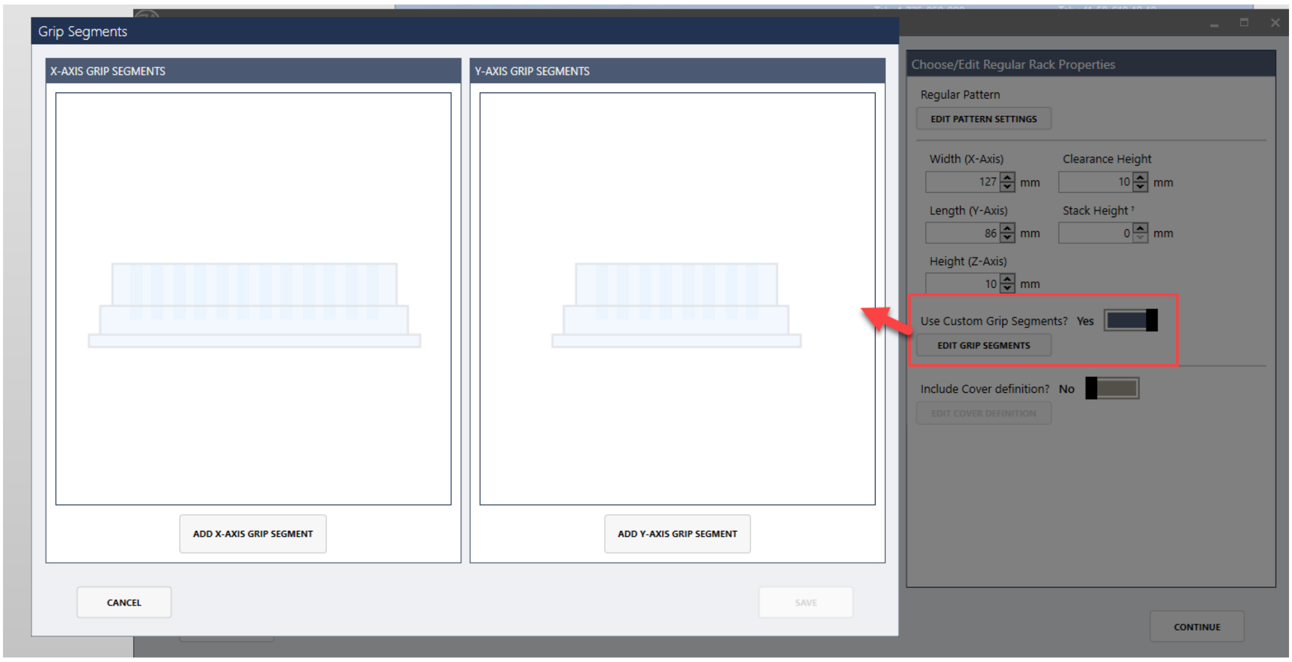
2. Click **Edit Pattern Settings** to edit the wells grid: distance to the Rack edge, number of rows, columns and Wells, default sequence order, spacing and staggering.



3. Enter the Rack outer X-axis, Y-axis, and Z-axis dimensions.
4. Specify the clearance and stack height.



5. If the Rack consists of more than one segment with different XYZ dimensions, enable **Use Custom Grip Segments**.



6. If defining the cover is intended, check the “**Include cover definition**” option.

   To define the rack cover, specify the following:

   - “**Covered Rack Stack Height**”, i.e. the stack height if covers without racks are stacked
   - Distance from “**Rack base to Cover Base**”
   - “**Stack Height**”, if racks with covers are stacked
   - “**Thickness of Cover**” (in Z dimension)
   - Cover “**Dimensions**” in x, y, and z
   - A cover name and description. Select an icon bitmap for the graphical representation of the cover, if one exists, and a 3D model file for deck layout rendering, if available.

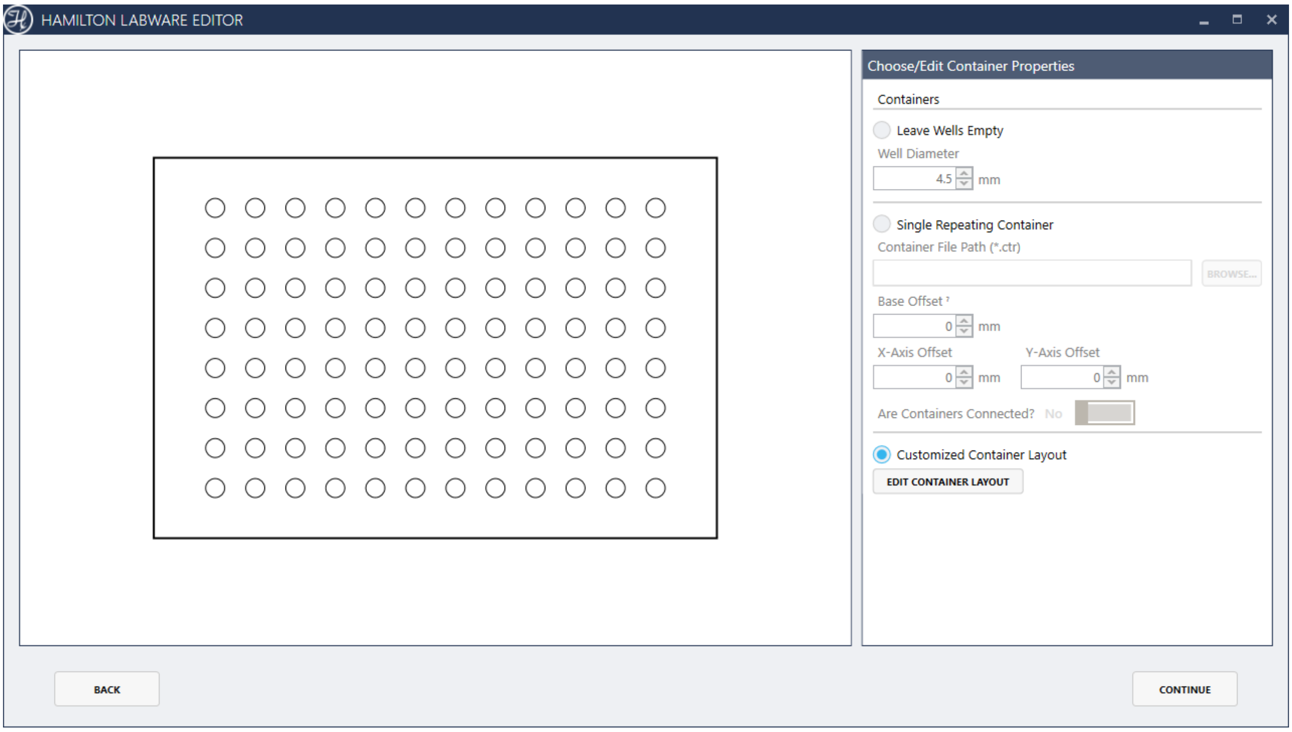
7. Click Continue.
8. Define the container definition(s) used for the wells in the rack:

   a. **Leave Empty**: no container definition will be used. Specify the well diameter for deck layout representation.

   b. **Single Repeating Container**: use the same container for all wells.
   Set the **base offset** (distance above or below the bottom of the Rack). This value

is negative if the Container's bottom is below the Rack's bottom; otherwise, is positive.

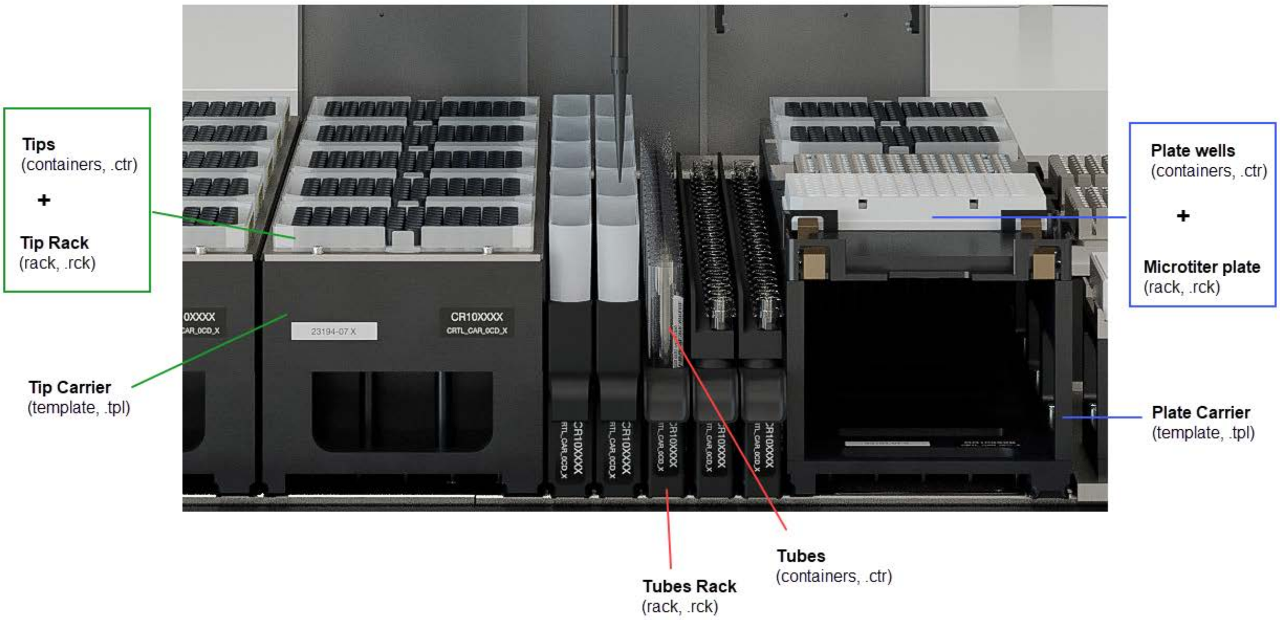
c. **Customized Container Layout:** customize the container and offsets for each well.



9. Click Continue.
10. Enter the Rack name and description. Select an icon bitmap for the graphical representation of the Rack, if one exists, and a 3D model file for deck layout rendering if available.
11. Click the "**Background Color**" Box to select a Rack background color of preference.
12. Disable **"Shall the labware be visible"** if the Rack should be not be visible in the deck layout whenever is not loaded or used.
13. Click Save and choose a destination folder for the Rack (.rck) definition file.

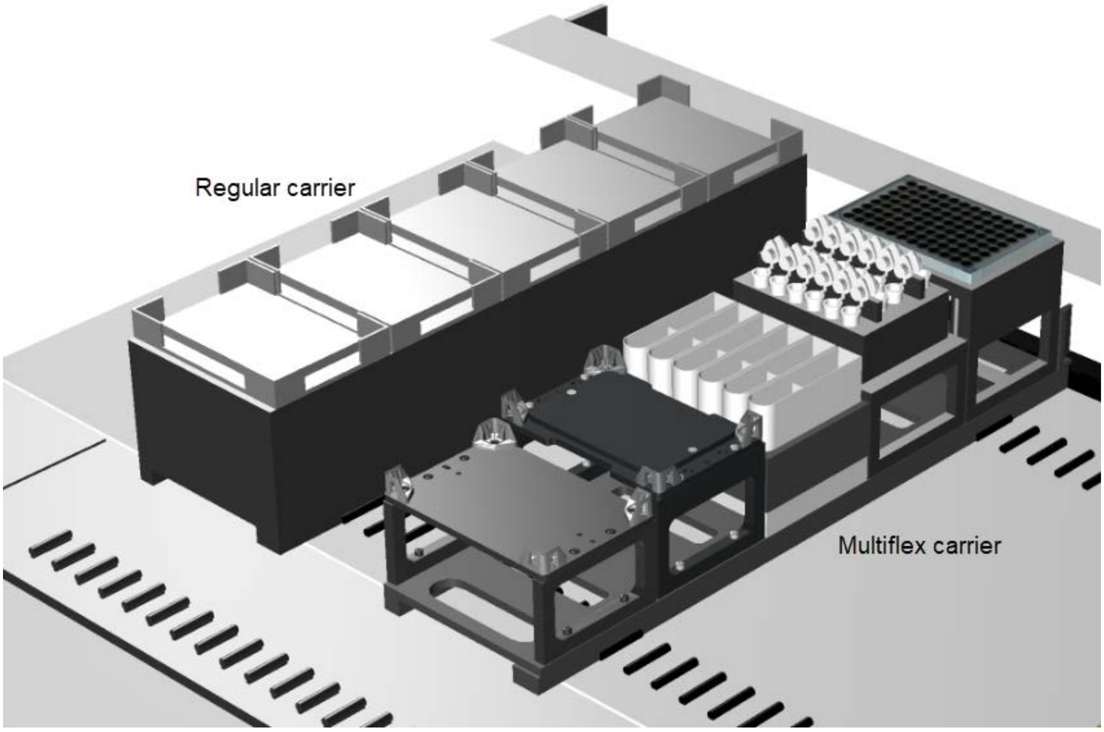
10.2 Defining a Carrier (Template)

Carriers are placed onto the Microlab instrument deck. They have sites for labware, such as Plates, Troughs or Tubes. A carrier file is defined as a template (*.tpl).



There are two types of Carriers:

- **Regular Carriers**, where all the carrier sites having the same dimensions
- **Multiflex Carriers**, where each carrier site can be a different module with different dimensions and holding a different Rack type.



The labware editor offers three options to create Carriers:

- **Custom Template**: use this for all regular Carriers
- **NIMBUS**: use this for NIMBUS multiflex Carriers
- **STAR**: use this for STAR Multiflex Carriers

In the following, a regular Carrier that is pre-loaded with flat 96-well Nunc Micro Plates is defined.

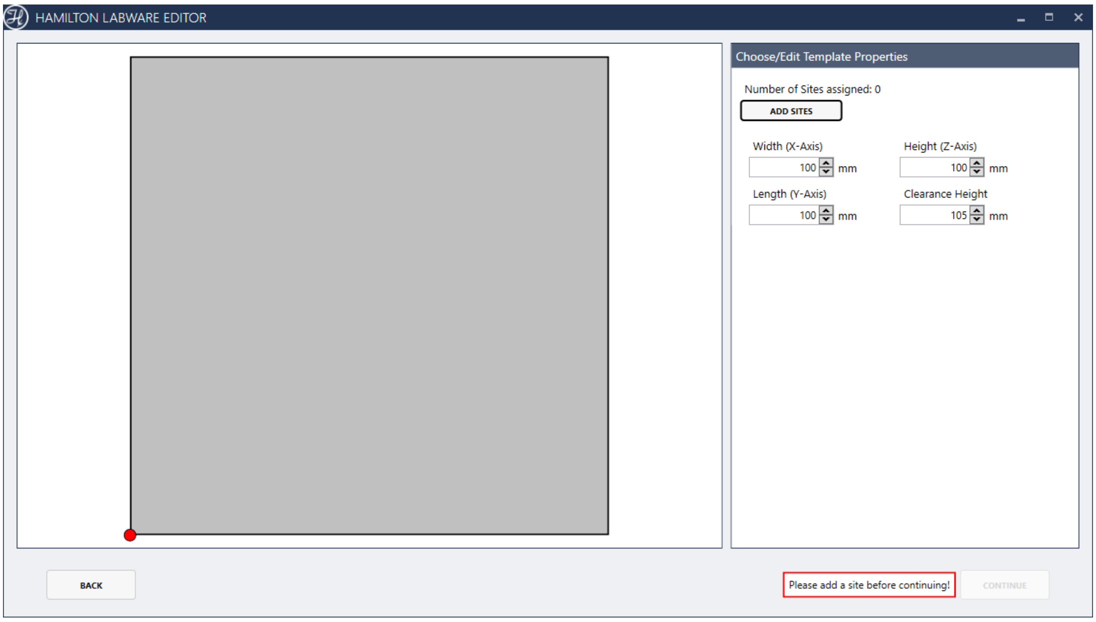
NOTE

A Tube Carrier in the sense of labware is not a Carrier (template). It is a Rack which directly fits the track geometry of the Microlab instrument and therefore can directly be loaded onto the instrument deck.

To define a Carrier:

1. Select "**Create New Labware → Custom Template**".

2. The "**Template Properties**" screen opens.



- Enter the Carrier dimensions (width, length, height)
- The "**Width**" of the Carrier is derived from:

$$width\,[mm] = \#\,[track] \cdot 22.5\,[\frac{mm}{track}]$$

- In the case of a Plate Carrier which is 6 tracks wide, the result is 136 mm. The length of a Microlab STAR track is always 497 mm.

- Set the clearance height (the height at which pipetting channels can safely move above the Carrier without colliding it)

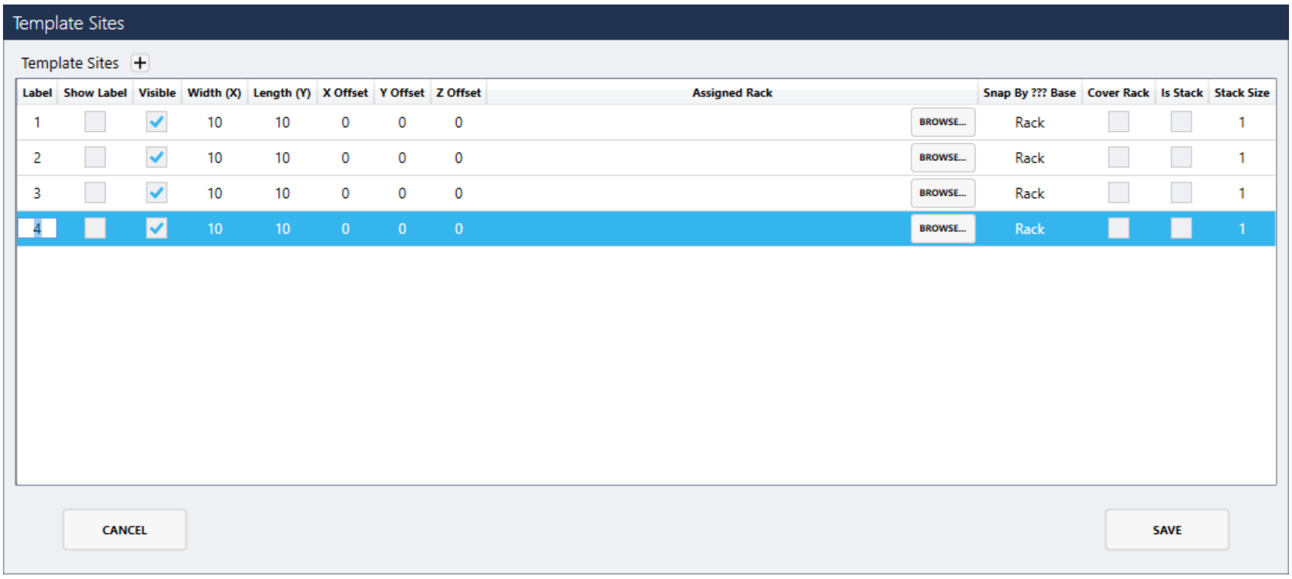
ATTENTION

The clearance height defines – the traverse height at which the pipetting channels can move without collision.

Clearance height should not exceed 140 mm.

Incorrect clearance height definitions can lead to serious damage.

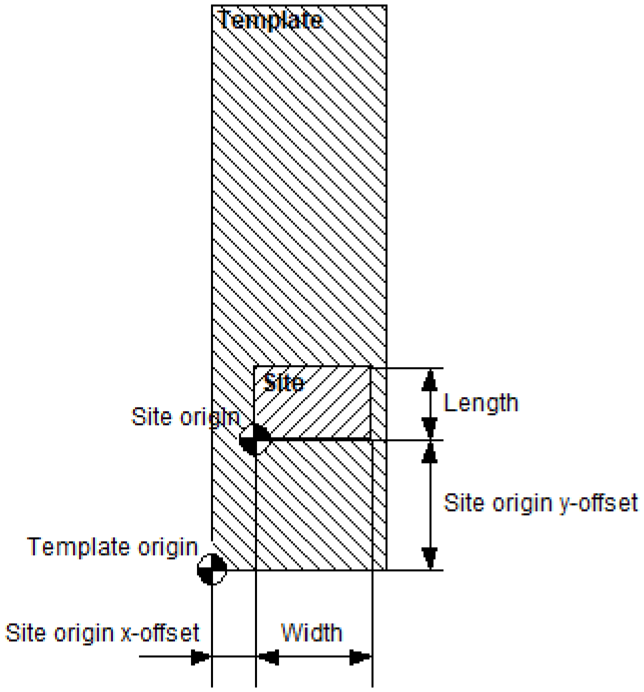
3. Click ‘Add Sites’ to define the sites hosting the plates have to be defined:



4. For each site, define:
   - **“Show label”** If checked, this will display the site label in the System Deck Layout.
   - **“Visible”** If checked, this will draw the site boundary in the System Deck Layout, otherwise the site will not be visible.
   - For “**Snap by base**”, select ‘Rack’ because a micro plate (rack) fits with its bottom onto the plate carrier, to enable a good electrical coupling for capacitance-based LLD.
   - All plate types should be defined with the same ‘footprint’ of 127 mm x 86 mm. Only this size will snap onto a default Carrier.
   - **“Assigned Rack”** defines the rack to be preloaded in a site when the template is added to the deck layout. Click on **[Browse]** to search for the corresponding labware to add pre-defined or user-defined labware to the site.
5. After filling out all the fields, click **Save** to go back to the Main Template Properties screen
6. Click **Continue**.
7. Enter the template Carrier name and description. Select an icon bitmap for the graphical representation of the rack, if one exists, and a 3D model file for deck layout rendering, if available.
8. Click the “**Background Color**” Box to select a template background color of preference.

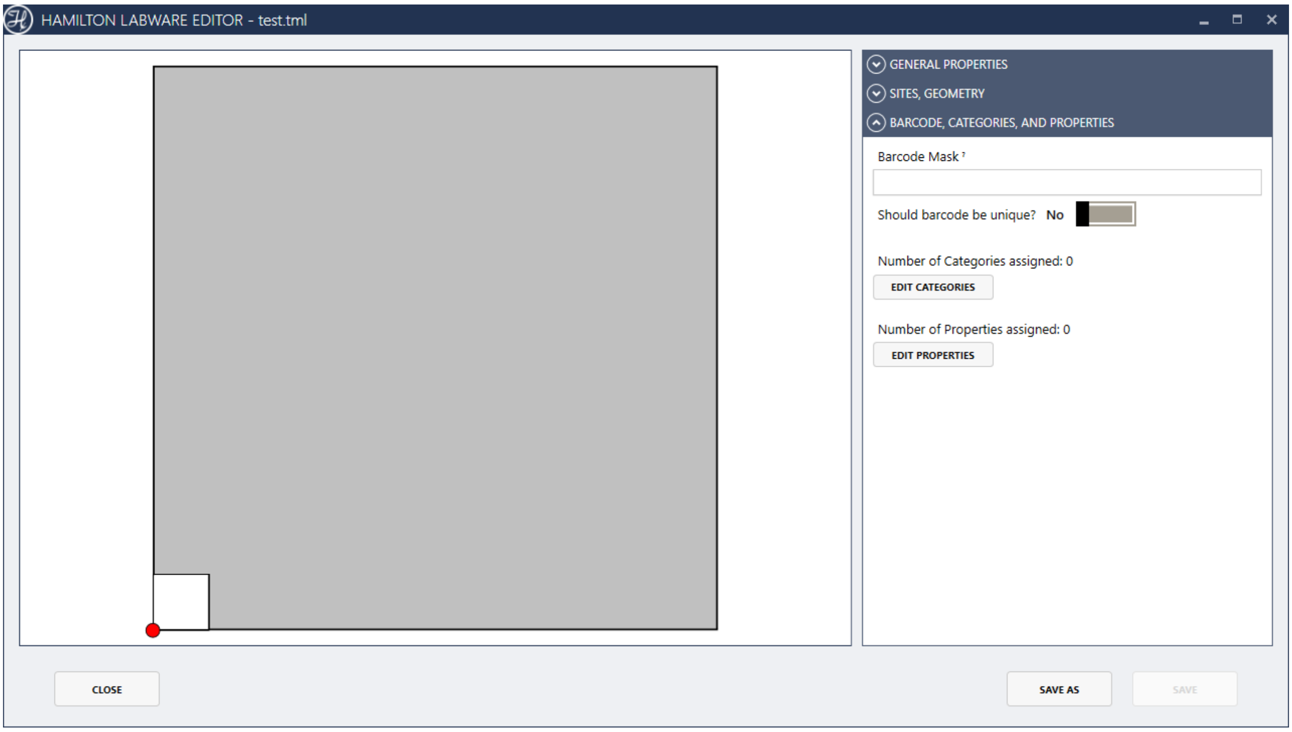
9. Disable **“Should the labware be visible”** if the template should be not be visible in the deck layout, whenever it is not loaded or used.
10. Click **Save** and choose a destination folder for the template (.tml) definition file.
11. The sketch below illustrates the different “**Origin**” coordinates and “**Dimensions**”.

    Whenever this Carrier is added to the deck layout it will be pre-loaded with a plate and the custom rack (sites) on top.

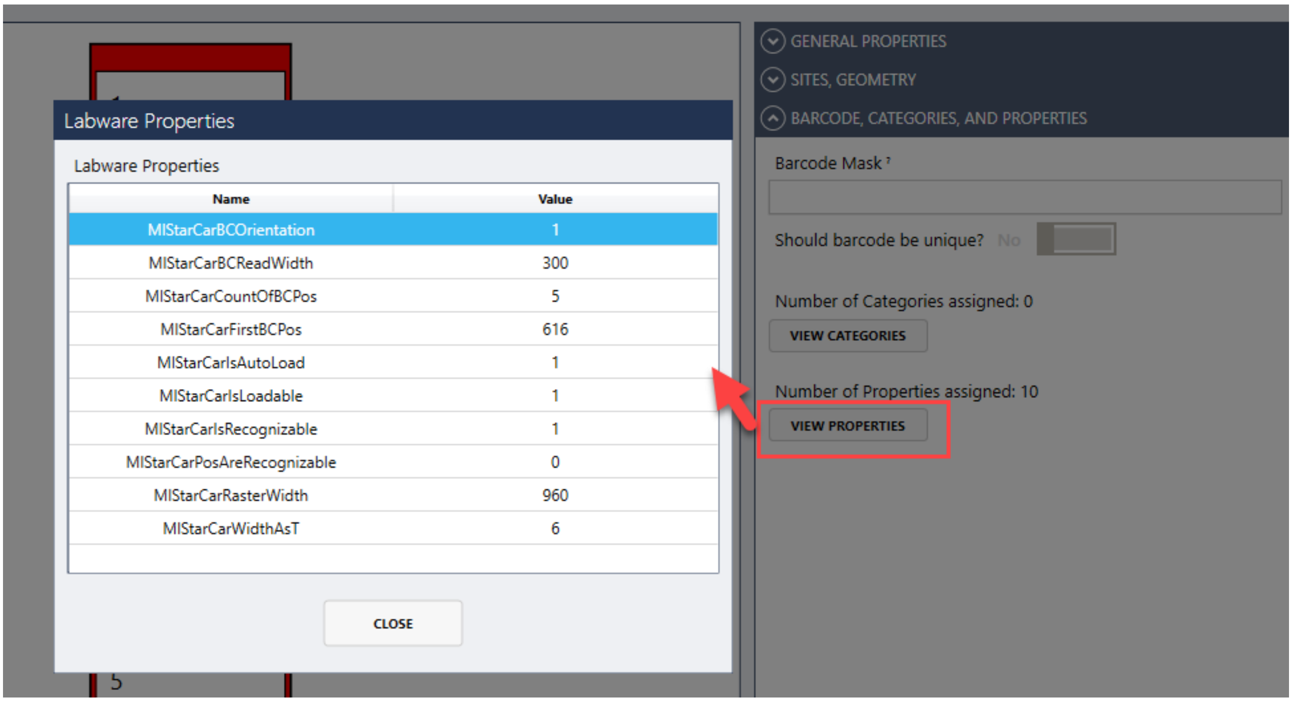


The decision whether a plate (or tip rack) fits in a site of a Carrier is made, depending on the length and width of the site: all plates that have the same boundary measurements (length 86 mm, width 127 mm) can be placed on the site.

12. After saving the template, the section to assign a barcode mask, labware properties and categories becomes available.



13. Click “**Edit Properties**” to define the labware-specific properties of the template carrier.



Example properties of a PLT_CAR_L5AC Template Carrier

NOTE

Always start with the given settings of a standard Carrier and then apply changes in a step-by-step manner. Do not change the properties’ names or their spelling. These names are system properties.

14. Click "**Edit Categories**" to define the labware categories to be assigned to the template Carrier. This selection will define where to display this template Carrier in the labware selection tree of the System Deck Editor (see more details in the Labware Categories section of this manual)

    Drag the desired categories to be assigned to the list on the right.

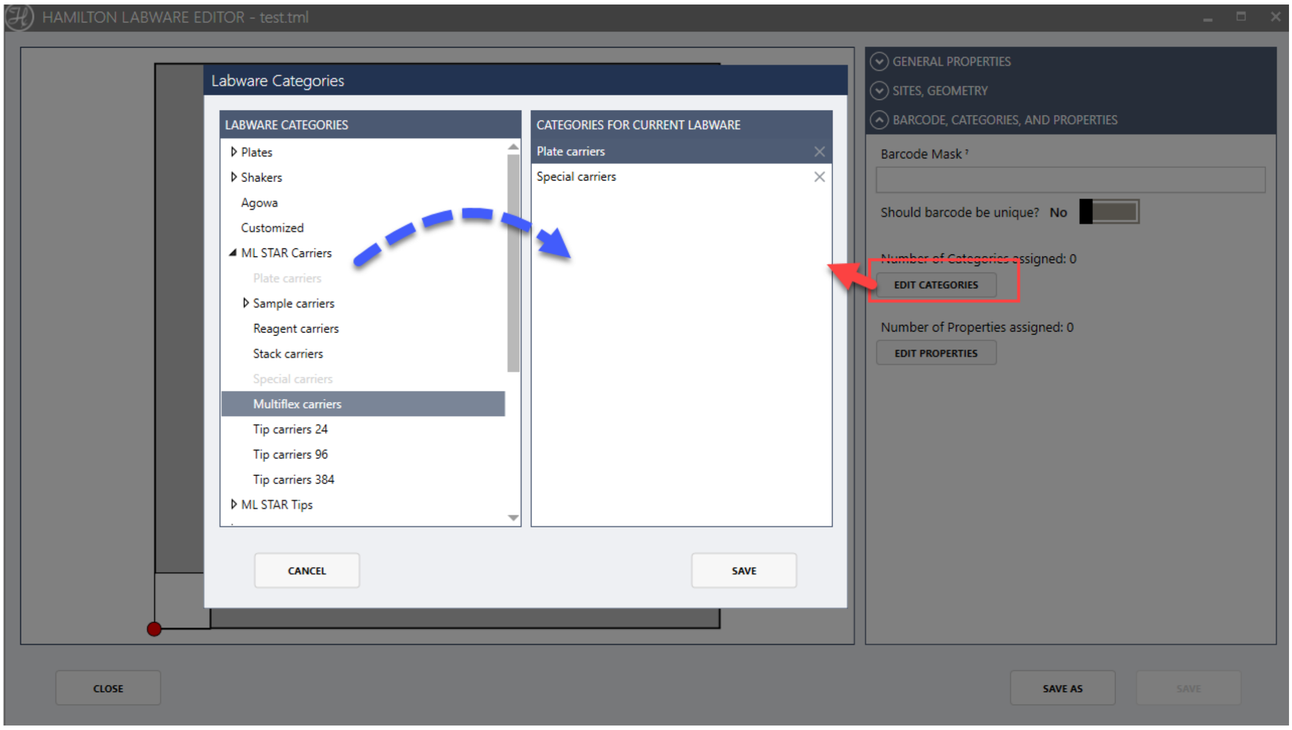
10.3 Labware Categories and Filters

To be able to group the labware in the selection tree of the System Deck Editor, each labware has to be assigned to a labware category first. For the labware that comes with the default installation, this has been done already.

However, in creating a new labware, the connection should be done manually.

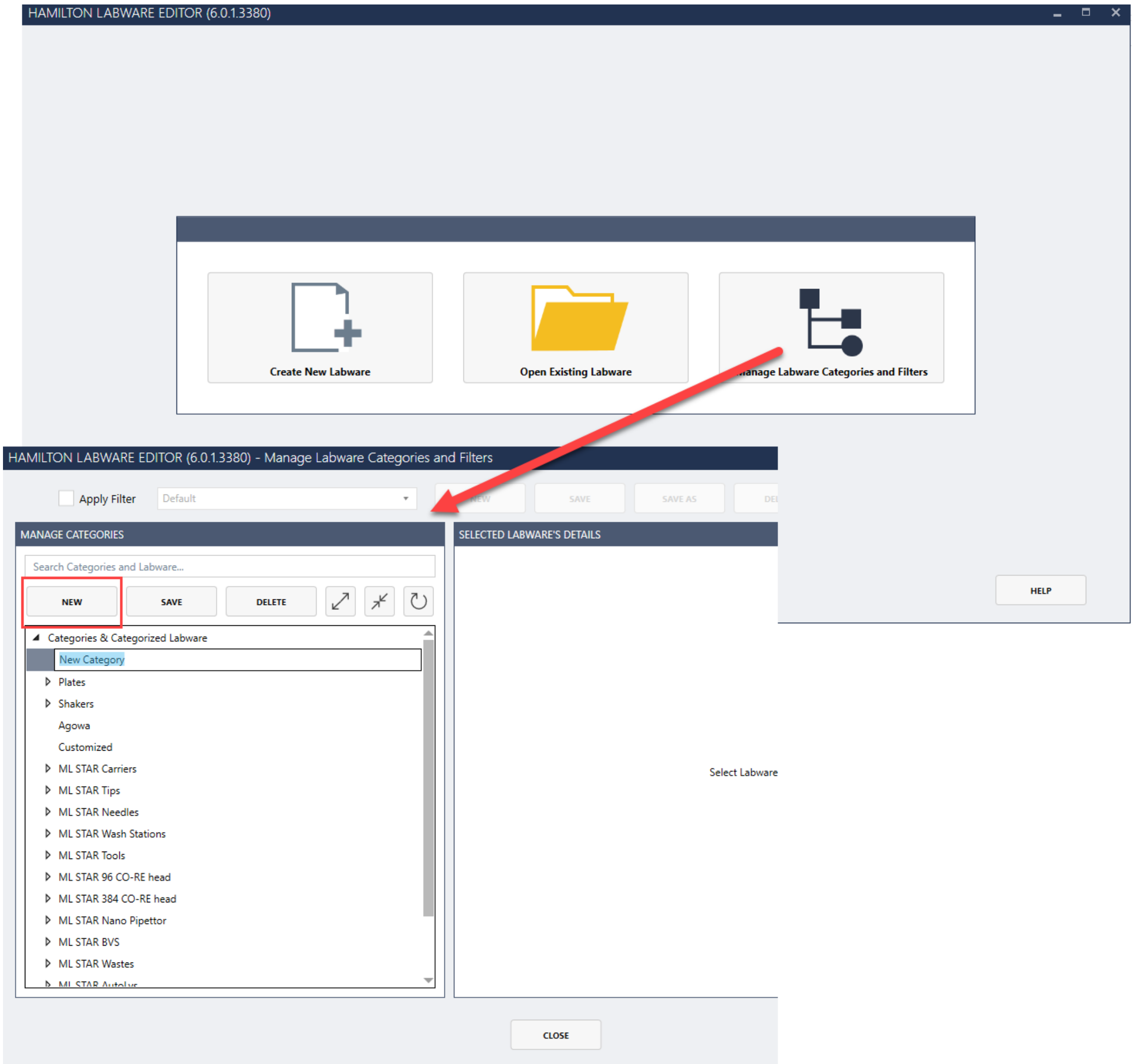
1. Open the Rack Template in the Labware Editor.
2. Go to the **"Barcodes, categories and properties"** section.
   Click "**Edit Categories**" to define the labware categories to be assigned to the Template Carrier.

   Drag the desired categories to be assigned to the list on the right.

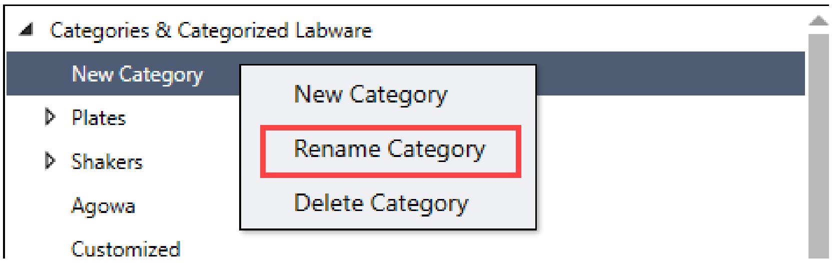


3. Assigning a labware for more than one category can be done. For instance, a 96-well Deep-Well Plate can be in the category "**Plates → 96 Position Plates**" and in the "**Deep-Well Plates**" Category.
4. In order to apply the changes into the software, restart the Method Editor / System Deck Editor.

5. If an appropriate category for the labware cannot be found, click **“New”** to create a new labware category, then click “**Save**”.



6. **Right-click** the new category and rename as needed.

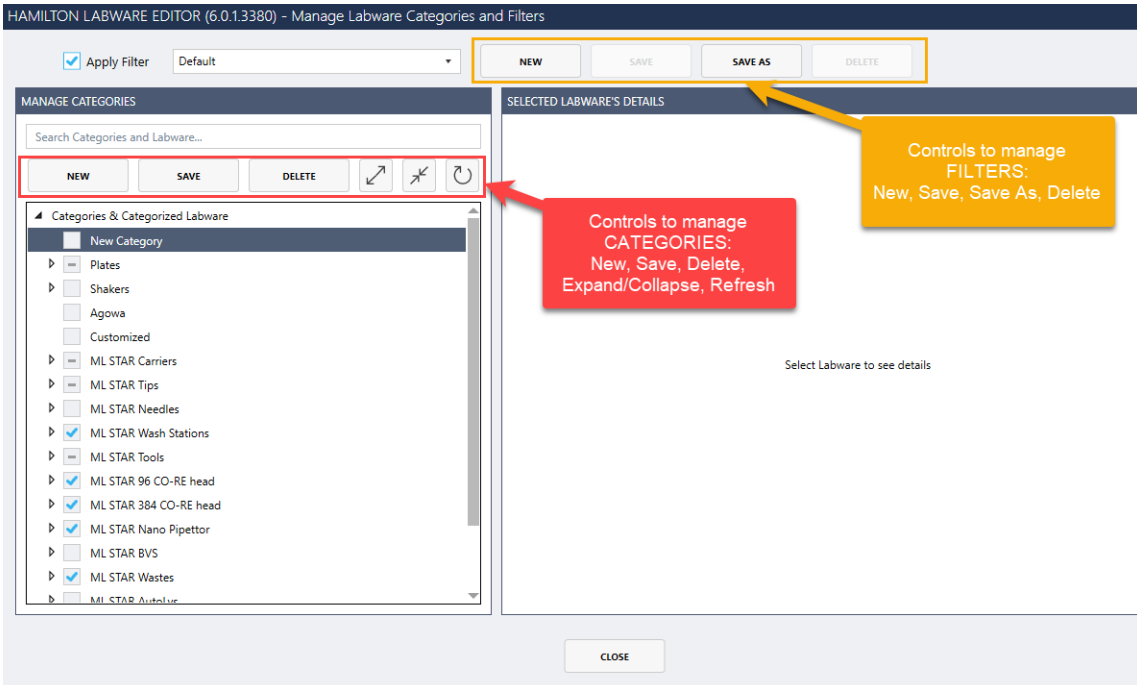


From this same contextual menu, a new child category can be created, or delete the entire category.

Remember to assign labware to the newly added category in the labware tree of the System Deck Layout. Empty categories will not be displayed. Edit the appropriate labware and assign it to the new category.

Additionally, **labware filters** can be applied to only show/hide specific categories in the Method Editor/ System Deck Editor. To enable this functionality, from the category manager click "Apply Filter".

Note that there are different controls for the categories and the filters. Different filters can be created for different needs (e.g., show only the labware available in the laboratory).



10.4 Labware Properties

Within the labware, some flags and settings (labware properties) are defined to determine handling of the labware elements. Labware properties should not be changed by the user, although they are accessible.

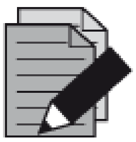
These properties can be divided up into the following groups:

- Information for the handling with the Autoload unit.
- Information for the accessories like tip handling, waste, etc.
- Information to support the reduction of selection during edit time.

Structure

The properties always combine a key name and a value. The key names are case-sensitive. All values are in integers (i.e. no decimal points).

To find the following tables, right-click an easy step and select "**what's this**". Click "**miscellaneous**" to expand and finally choose "**Definion of labware properties**".



NOTE

Key names of labware properties are case-sensitive.

Definition of labware properties

Labware definition may contain several labware properties to specify handling of the labware elements. Microlab STAR® specific properties are defined within the following tables.

Caution: *Key names are case-sensitive.*

General carrier properties:

Key name	Type	Default	Range [unit]	Description
`MlStarCarWidthAsT`	long	-	1..xx [T]	Width of carrier in loading track positions (1T = 22.5mm).
`MlStarCarrierBarcodePos`	long	43	0..4700 [1/10 mm]	Barcode position of carrier (Distance between rear of carrier to middle of carrier barcode position).
`MlStarCarrierBarcodeWidth`	long	85	1..999 [1/10 mm]	Width of carrier barcode read window.
`MlStarBarcodeReadSpeed`	long	1281	15...1600 [1/10 mm/s]	Barcode reading speed.
`MlStarCarCountOfBCPos`	long	-	0..32	Count of barcode positions of cups or racks
`MlStarCarBCOrientation`	long	0	0..1	0 = Vertical, 1 = Horizontal barcode read direction.
`MlStarCarFirstBCPos`	long	-	1..4700 [1/10 mm]	Distance between rear of carrier to middle of first barcode position.
`MlStarCarRasterWidth`	long	-	1..4700 [1/10 mm]	Distance middle of barcode position to next middle of barcode position.
`MlStarCarBCReadWidth`	long	-	1..999 [1/10 mm]	Width of barcode read window.
`MlStarCarIsRecognizable`	long	0	0..1	1 = TRUE, 0 = FALSE (true, if Carrier has a magnet)
`MlStarCarIsLoadable`	long	0	0..1	1 = TRUE, 0 = FALSE (true, if Carrier is loadable)
`MlStarCarIsAutoLoad`	long	0	0..1	1 = TRUE, 0 = FALSE (true, if carrier is loadable with autoload)
`MlStarCarPosAreRecognizable`	long	0	0..1	1 = TRUE, 0 = FALSE (true, for sample carriers with recognizable tubes only!)
`MlStarCarNoReadBarcode`	long	0	0..1	1 = Don't read barcode, 0 = read barcode If set to 1, only carrier barcode is read.
`MlStarReadBarcodeOnPositions` For more information UniqueBarcodeAssignment	string	""	"1=2;4=3;" or empty or "1=2;4=3.1"	Positions of read barcode positions assigned to labware item position. If string is empty or does not exist, all barcode positions are read *Example / Notes:* • "1=1;4=7;6=2.A1" This positions must have a valid barcode. The 2.A1 is the position from a rack on a template. • The separator between the positions **must be** a semicolon '**;**' • A separator at first position is not accepted • Spaces are not accepted • Position 1 is always at rear of carrier • If `MlStarCarNoReadBarcode` set to 1, no barcodes are read.
`MlStarCarLabelName`	string		CarrierName	Contents the name of the carrier. If exist and not empty, this name will be displayed within error dialogs. If this key does not exist or the string is empty the carrier name of deck layout will be displayed within error dialogs.

Special carrier properties:

Key name	Type	Default	Range [unit]	Description
Carrier with calibration hole				
`MlStarCarCalibrateX`	long	-	0..30000 [1/10 mm]	Distance between left margin of carrier to middle of measure hole.
`MlStarCarCalibrateY`	long	-	0..6500 [1/10 mm]	Distance between carrier front to middle of measure hole.
`MlStarCarCalibrateZ`	long	-	0..3500 [1/10 mm]	Distance between deck to the top of measure hole
Incubator Carrier				
`MlStarCarIsTemperated`	long	0	0..1	1 = TRUE, 0 = FALSE
`MlStarCarIncubatorNumber`	long	-	1..2	Temperate incubations station 1 or 2

Rack properties:

Key name	Type	Default	Range	Description
`MlStarIsDefaultWasteRack`	long	0	0..1	0 = the rack is not a default waste 1 = the rack is a default waste rack A default waste must be a rack with reusable positions. If more than one default waste rack is defined the first one which will be found is used a default waste. If property `MlStarDefaultWasteVolume` is defined, the default waste rack with the highest volume is used as default waste. **Requirements**: For channels: • If this property is set, the corresponding property for channels `MlStarIsWasteRack` must be defined too. • The rack must define all Y positions in the same X position. • The first and last y positions are used for calculate the Y position for all channels. For CO-RE 96 Head: • If this property is set, the corresponding property for CO-RE 96 Head `MlStarIsCore96WasteRack` must be defined too. • The rack must be a rack with 96 positions in a raster of X = 9mm and Y = 9mm. For CO-RE 384 Head: • If this property is set, the corresponding property for CO-RE 384 Head `MlStarIsCore384WasteRack` must be defined too. • The rack must be a rack with 384 positions in a raster of X = 4.5mm and Y = 4.5mm.
`MlStarIsWasteRack`	long	0	0..1	1 = Rack may be used to eject tips/needles with channels.
`MlStarIsXLWasteRack`	long	0	0..1	1 = Rack may be used to eject tips with 5ml-channels.
`MlStarIsCore96WasteRack`	long	0	0..1	1 = Rack may be used to eject tips with CO-RE 96 Head.
`MlStarIsCore384WasteRack`	long	0	0..1	1 = Rack may be used to eject tips with CO-RE 384 Head.
`MlStarIsNanoWasteRack`	long	0	0..1	1 = Rack may be used to flush the Nano Pipettor.
`MlStarDefaultWasteVolume`	long	0	positive value	Volume of waste in mm3 Note: This value is only used for search the default waste with the highest volume.
`MlStarTipRack`	long	-	0..xx	Value specifies the tip type hold within rack. The rack may be either a tip rack, needle rack or a rack for a tool. If this property is set, the rack can be used to pickup the specified tip type with channels. *Note: Set only a tip type which is intended to be used with channels. See Microlab STAR® Operating Manual for more information.*
`MlStarXLTipRack`	long	-	0..xx	Value specifies the tip type hold within rack. The rack may be either a tip rack or a rack for a tool. If this property is set, the rack can be used to pickup the specified tip type with 5ml-channels. *Note: Set only a tip type which is intended to be used with 5ml-channels. See Microlab STAR® Operating Manual for more information.*
`MlStarCore96TipRack`	long	-	0..xx	Value specifies the tip type hold within rack. If this property is set, the rack can be used to pickup the specified tip type with CO-RE 96 Head. *Note: Set only a tip type which is intended to be used with CO-RE 96 Head. See Microlab STAR® Operating Manual for more information.*
`MlStarCore384TipRack`	long	-	0..xx	Value specifies the tip type hold within rack. If this property is set, the rack can be used to pickup the specified tip type with CO-RE 384 Head. *Note: Set only a tip type which is intended to be used with CO-RE 384 Head. See Microlab STAR® Operating Manual for more information.*
`MlStarCore384TipRackFourToOneTip`	long	-	0..1	1 = Tip rack contains "R O C K E T" Tips to be used with CO-RE 384 Head.
`MlStarCore384TipRackMode96`	long	-	0..1	1 = Tip rack may be used from CO-RE 384 Head with reduced pattern mode.
`MlStarNeedleWashRack`	long	-	0..xx	The value specifies the needle type hold within a washer rack. *Note: Set only a needle type which is intended to be used with channels. See Microlab STAR® Operating Manual for more information.*
`MlStarWashStationNumber`	long	-	1..3, 4..6	WashStation1: 1..3, WashStation2: 4..6

MlStarIsSpecialEject	long	0	0..1	1 = This rack needs a special movement to eject tips. Only one rack with this property set must be on deck. This property is reserved for special rack (e.g. waste2.rck) and may not be used elsewhere.
MlStarCore384SpecialEject	long	0	0..1	1 = Special eject of CO-RE 384 Head Needle-Adapter respectively CO-RE 384 Head Pin Tool.
MlStarIsSpecialTipPickup	long	0	0..1	1 = Keep a residual air volume, when picking up tips from this rack.
MlStarIsCoreHeadSpecialTipPickup	long	0	0..2	0 = Pick up tips without aspirate air before (normal mode) 1 = Aspirate air before tip pick up and blowout during move up (used on washer). 2 = Aspirate air before tip pick up (A dispense with 'Dispense mode' (0) Blowout tip must be use for blowout the tip on a waste or washer).
MlStarIsCore96WashRack	long	-	0..1	1 = Rack can be used for wash step with CO-RE 96 Head.
MlStarIsCore384WashRack	long	-	0..1	1 = Rack can be used for wash step with CO-RE 384 Head.
MlStarIsNanoWashRack	long	-	0..1	1 = Rack can be used for wash step with Nano Pipettor.
MlStarCore96WashRack	long	-	0..xx	Value specifies the tip type allowed to be washed on the CO-RE 96 Washer. *Note: Set only a tip type which is intended to be used with CO-RE 96 Washer. See Microlab STAR® Operating Manual for more information.*
MlStarCore384WashRack	long	-	0..xx	Value specifies the tip type allowed to be washed on the CO-RE 384 Washer. *Note: Set only a tip type which is intended to be used with CO-RE 384 Washer. See Microlab STAR® Operating Manual for more information.*
MlStarPumpStationNumber	long	-	1..3	Value specifies the firmware node of pump station. 1 = Node HW, 2 = Node HU, 3 = HV
MlStarPumpUnitChamberNumber	long	-	1..2	Value specifies the chamber number of a Dual Chamber Head Washer. 1 = chamber one (rear chamber), 2 = chamber two (front chamber)
MlStarIsCore384TipLifter	long	0	0..1	1 = Tip rack may be used from CO-RE 384 Head for pick up a single column or two neighboring columns. If this value is set to '1', the value of key MlStarCore384TipLifterLiftUpDistance must be defined too.
MlStarCore384TipLifterLiftUpDistance	long	-	positive value	Value specifies the distance [1/10 mm] the tip lifter will move up tip column(s).
MlStarCore384TipLifterType	long	0	0..1	0 = Tip lifter can lift up column(s). 1 = Tip lifter can lift up row(s) *Note: This key is optional, at the moment lift up of row(s) cannot be used.*
MlStarCoReGripCheckPlateYDelta	long	-	positive value	Defines the relative distance [1/10 mm] the CO-RE grip clamp will close before plate check.
MlStarCoReGripCheckPlateZDist	long	-	positive value	Defines the distance [1/10 mm] the CO-RE grip will: - move up from labware surface after check plate is done before 'get plate' step is started - move up from labware surface after 'place plate' step before check plate is started.
MlStarRackSTubeZDistance	long	-	positive value	Defines the distance [1/10 mm] from the top rim of the S-Tube to the center of the bayonet cap where the capper may start capping/decapping process.
MlStarTeachingNeedlePosition	long	-	0..X	Defines the position on which the channel (1000µl or 5ml) is search the teaching needle. Value greater than 0, MoveProbe is search the teaching needle in this labware on the defined position X. X = must be between 1.. max position of used labware. Dependent of 1000µl channel or 5ml channel the additionally corresponding key MlStarTipRack or MlStarXLTipRack must be defined too.

NOTE

*At edit time, the user does not have to define the tip type in the "**pick-up tip or needle**" Step. The information is taken from this property.*

*At edit time, the user does not have to define the tip type in the "**eject tip or needle**" Step.*

At edit time, the user can only select sequences out of the list of sequences with underlying racks having these properties.

At edit time in the aspiration or dispense step, the user can only select sequences having underlying racks without these properties out of the sequence list.

Needles can only be ejected into a rack with the property MlStarIsWasteRack and corresponding Microlab StarTipRack settings.

NOTE

There are only two CR Wash Stations allowed on the same deck.

Each Wash Station must have a unique identifying number.

At edit time during the wash step, the user can only select sequences with underlying racks having these properties out of the sequence list.

10.4.1 Properties for Temperature Controlled Carrier (TCC)

Key	Default	Range	Description
MlStarCarIsTemperated	0	0 or 1	The carrier can be temperature-controlled 1 = TRUE, 0 = FALSE
MlStarCarIncubatorNumber	-	1 or 2	Number of temperature incubation stations 1 or 2

NOTE

There are only two TCCs allowed on a deck.

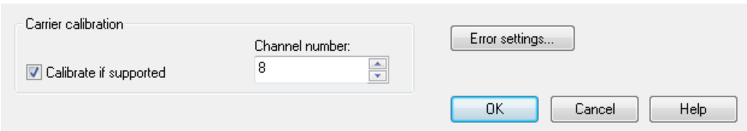
At edit time in the incubator step, the user can only select sequences with underlying Racks having these properties out of the sequence list.

Properties for “**calibrate carrier**” Single Command (Carrier 1536):

NOTE

If a Carrier can be calibrated, all values should be set.

The origin is the zero point of the zero position of the Carrier (front, left, down).



The calibrate action is executed only if the calibrate property is set.

11 Advanced Programming

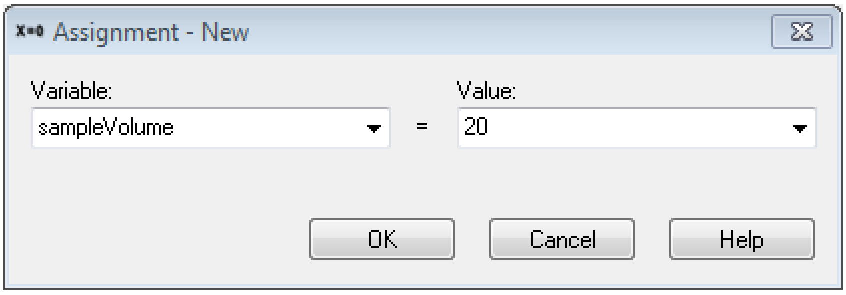
11.1 Variables and Return Values

11.1.1 Variables

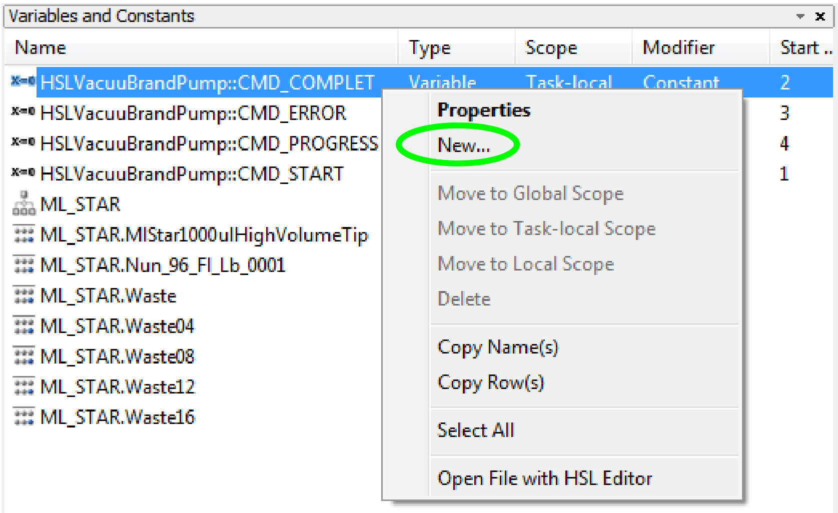
Within a method, the programmer will naturally define and process variables.

There are two ways to specify variables:

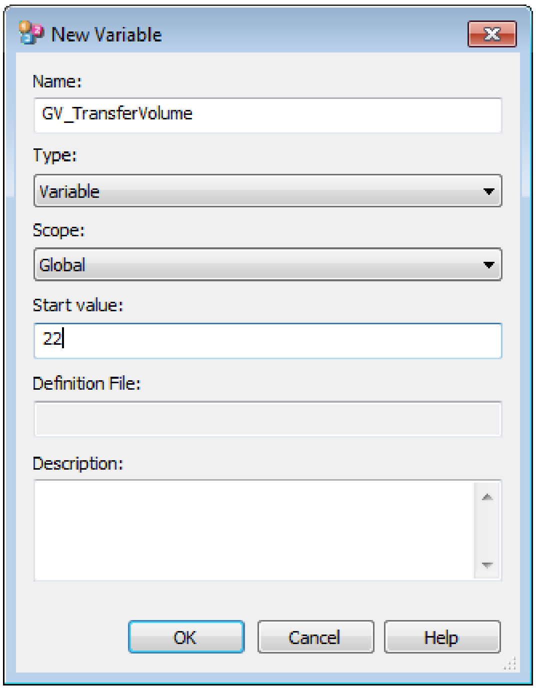
1. Use the “**Assignment / Assignment with calculation steps**” from the “**General Steps**”.
2. Type in the variable and the value needed and confirm with **[OK]**.



3. Activate the “**Variables and Constants**” Screen shown below in the “**View → Variables**” Menu or with **[ALT + F7]**.
4. To create a new variable, trigger the Context Menu with a right-click in the Variables and Constants screen, and select “**New**”.



5. Type in the name, type, scope, start value and the description (optional). Confirm the entries with **[OK]**.



6. The new variable is now available in the method.

The Method Editor supports three kinds of variables with differing scopes:

Local variables: Valid only within a sub-method. All variables declared inside a sub-method automatically have a local scope. Local variables can be promoted to "**Task local variables**" through the "**Method →Export local variable**" Menu.

Task local variables: Valid within a method. All sub-methods have access to variables of this type. Local variables and Task local variables with identical names can coexist however, it is recommended to use naming conventions or prefixes. Such variables are generated when they are declared within the main program, or for example, within the method. When the method is scheduled, each task will have its own task local variable.

Global variables: Can be accessed by all tasks using a method (see the Scheduler Manual). They can be declared in the Method Editor through the "**Method → Global variables**" Menu. This menu option can also promote a Task local variable to a global variable.

If a workflow contains a task local variable and a global variable with the same name, a name conflict occurs. In this case, a syntax error hint will be signaled.

NOTE

It is important to avoid mixing and confusing local and global variables. If possible, use naming conventions allowing the identification of each variable type:

GV_*<variable name> for global variables used for all tasks of the same method*

TV_*<variable name> for task local variables used in one method*

LV_*<variable name> for local variables used in a sub-method*

It is not possible to define variables which are valid in a method without being also valid in the related sub-methods. For this requirement, define task local variables with reserved names; e.g.:

MLV_*<variable name> for local variables used in a method.*

The Graphical Method Editor provides variables which can accommodate three sub-types of variables without explicit declaration:

- Integers (numbers)
- Floats (floating point numbers)
- Strings (text variables and characters)

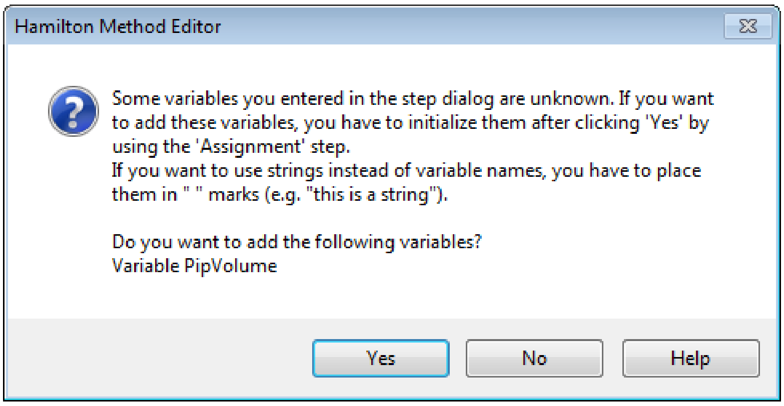
Variables can be newly defined within the dialogs, if they occur in the following:

- Left-hand side of an equation (e.g. x=0, or x=t+1), or
- Within entry fields that hold only one variable

If not requested (as is the case for the "**Open File**" Dialog), the variable types are identified automatically. For example, f=5 will generate a variable f of type integer.

NOTE

To define a string (text) variable through the assignment step, the text must be enclosed in quotation marks, e.g. ***"s"*** *=* *"**Test**". The variable* ***"s"*** *is then of type string. Otherwise, the contents of a variable named "**Test**" and its contents will be assigned to s. If the "**Test**" variable does not exist, a warning will be prompted.*



Variables once defined appear in the drop-down list of all input fields of a method.

Syntax

For variable names, **do not**:

- place numbers at the beginning
- use blanks or signs other than the underscore (“_”)

Variable names are case-sensitive.

To define a path within a string variable in the course of programming, use the double backslash (“****”) - e.g. “c:\\programfiles\\hamilton\\methods\\“. To define a path at run time by typing it into an input box, use single backslashes (“****“). The reason for this is that variables defined in the course of programming are analyzed for escape sequences (for example single backslashes “****”), and single backslashes followed by n (“**\n**“) means “**new line**”.

11.1.2 Return Values

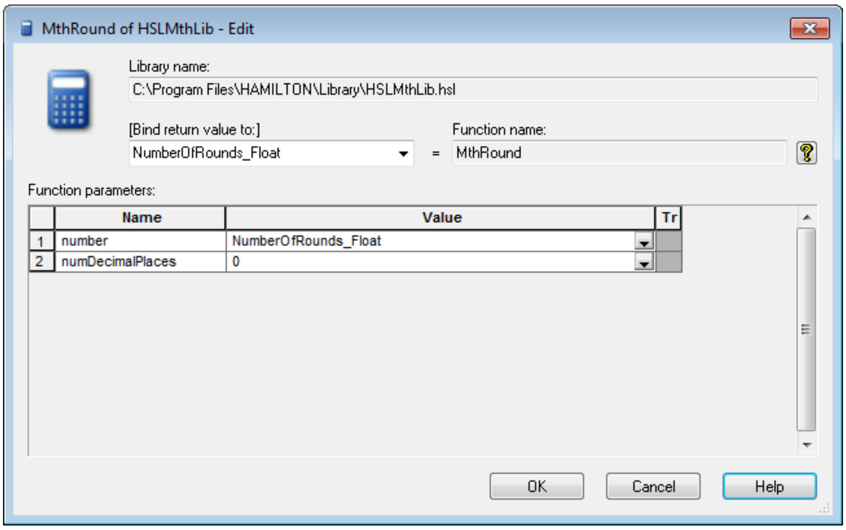
Often times, the steps used will return data when the step is executed. A typical example would be a mathematical function, like a simple assignment x=0.

Dialog functions or instrument steps can also return data:

- **Dialog functions** will provide a code of the button pressed upon exit enabling the method to continue depending on the button pressed in a particular dialog.
- **Instrument steps** can, for example, return a temperature for the TCC (Temperature-Controlled Carrier) or a pressure value for the BVS / CVS.

Example 1 (Function Call)

Using the "**Round**" Function of the HLSMth library leads to the dialog below displaying an entry field "**Bind return value to:**" which by default is empty. Select a variable listed in the drop-down list or enter a variable name, even one that does not yet exist. Leaving this field empty means that there is no variable where the rounded value of *NumberOfCycles_Float* will be taken from.



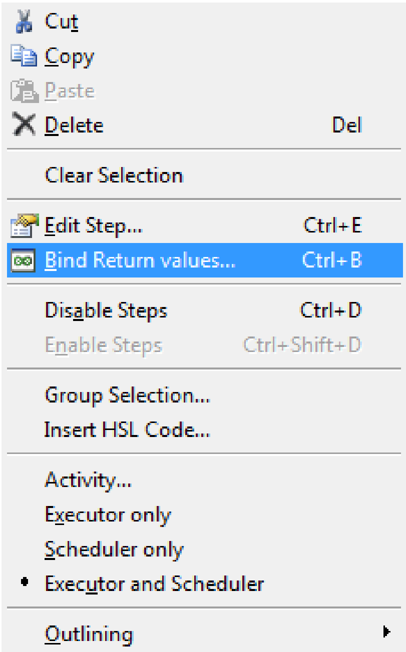
Example 2 (Instrument Step Return Values)

In a similar way you can have access to the additional return value of, for example, the "**GetTemperature**" Function of the TCC. Some functions return more than one item of information.

If you use this "**GetTemperature**" Function, you will only be asked for the TCC you wish to use. The value of interest and the returned temperature, needs to be bound in a separate task.

To do this, perform these steps:

7. Right-click on the desired step.
8. Select "**Bind Return values...**" from the Context Menu that opens.

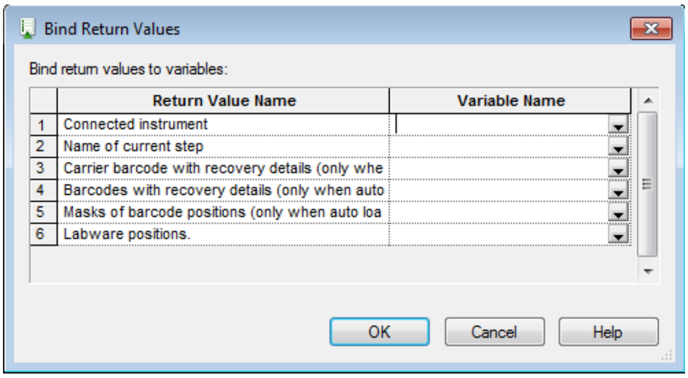


The “**Bind Return values…**” Function is also available as a function button in the toolbar. The function is active for method steps which have additional return values.

NOTE

*The “**Easy Steps**” and “**Smart Steps**” do not have the “**Bind Return values…**” Function.*

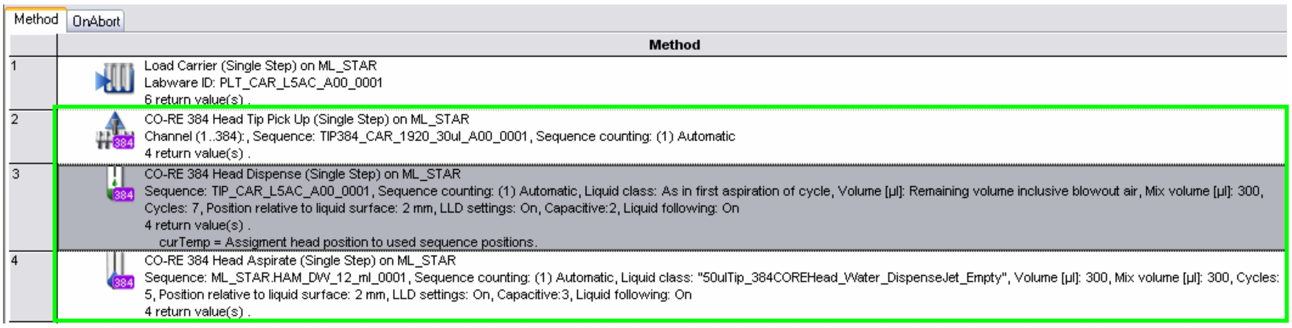
These pieces of information as presented in the image below can be stored in variables for further processing or to be used as output. The dialog generally lists the return values, each with an editable field “**Variable Name**”, to define the variable the return value is to bound to.



11.1.3 When to Bind Variables

It is up to the programmer to choose whether to bind variables or not. Often times, the variables need to be handled in a different way from what is shown in the examples. For user-defined error handling, there is a necessity to bind return values.

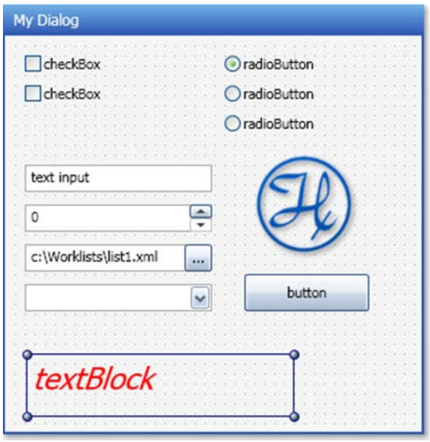
Functions with additional return values which have to be bound by the procedure, as shown above can be simply recognized in a method. The number of return values is given, even if it is not going to be handled (see in the topmost label of method 2, "**Tip Pick Up**" in the image below). The return value(s) assigned to the variables will be shown also (see the method Steps 3 and 4 in the image below).



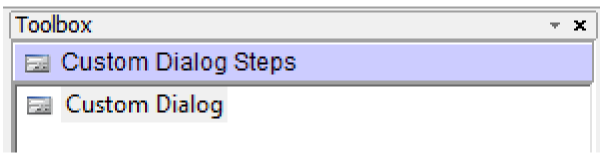
11.2 Customized Dialog

What is the Custom Dialog Step?

The custom dialog step allows for the creation of user dialogs that contain multiple and different input controls such as Drop-Down Lists, Checkboxes, Radio Buttons, Text Input Controls, File Browse Controls, Numeric Input Controls, Pictures, Buttons and Text Blocks. All of these controls can be arranged in any desired way.



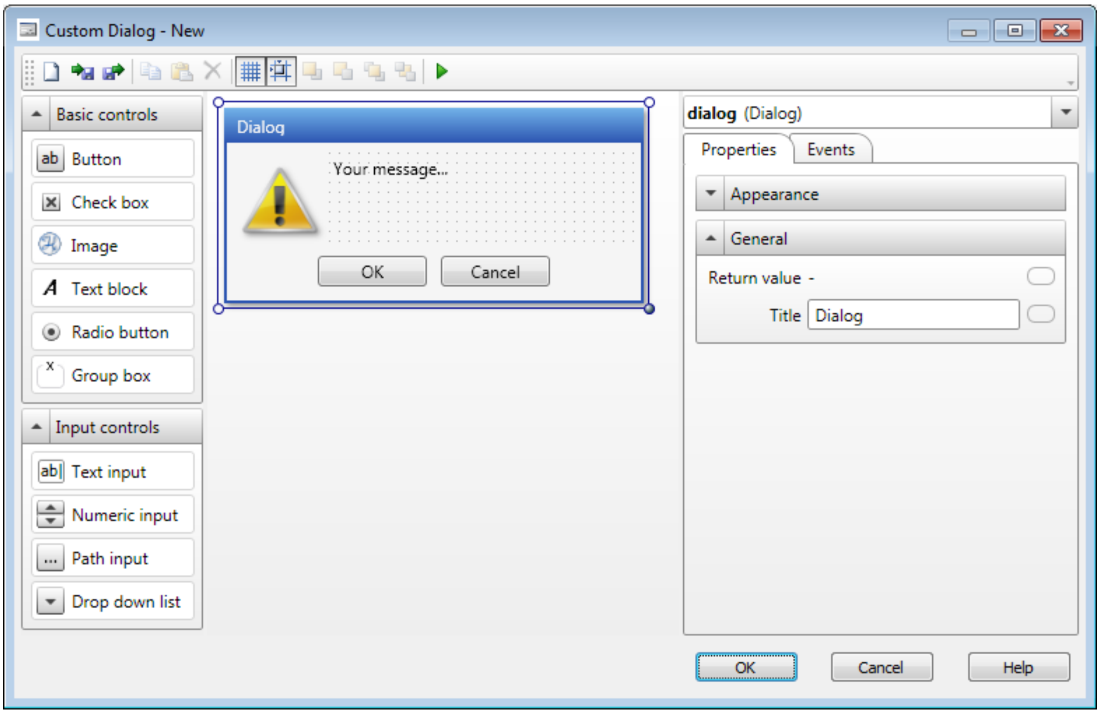
The custom dialog step is located in a separate "**Custom Dialog Steps**" Tab.



After a “Drag-and-Drop” to the method, the selection for templates is displayed.

Choose either blank or User Output: 1 Button, User Output: 2 Buttons or User Output: 3 Buttons to start with a predefined dialog that contains existing buttons.

After this selection, the editor starts (In this example: User Output: 2 Buttons)



Step 1: Add controls to the custom dialog

Customize the dialog box by adding basic controls and input controls from the toolboxes located on the left of the window. The elements can be added through a simple “Drag-and-Drop”. It is then possible to arrange the elements added into the dialog. The dialog itself can be adjusted in size.

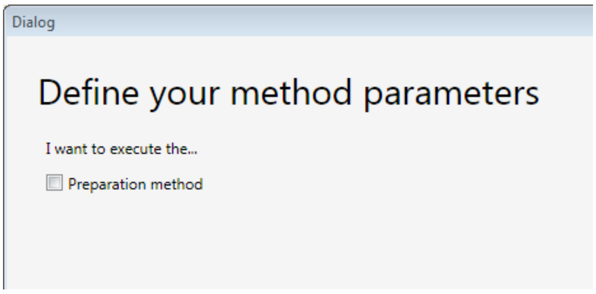
Step 2: Adjust the Properties of the Added Elements

To make changes on an element, simply select the element and choose ‘Properties’. Depending on the type of the element, the appearance, the behavior, and the font can be adjusted. All elements contain ‘General’ properties that are used to specify the variable binding and containing the handler of the element which is the ControlID. This ID is used to link events to the elements and makes the dialog ‘intelligent’.

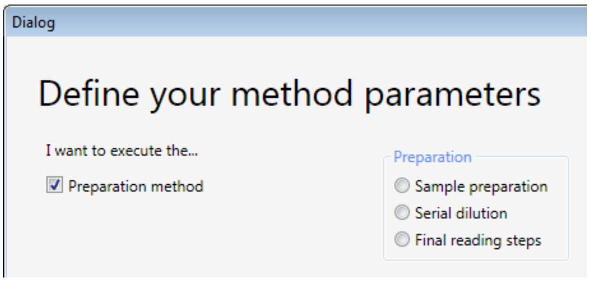
Step 3: Add Intelligence to the Custom Dialog

Using the ‘Events’ on the custom dialog can make the dialog intelligent. This means that specified elements of the dialog are only shown if e.g. a checkbox of another element is set. To learn more, refer to the example to below.

Example:



The details of the 'Preparation method' are only visible if the appropriate checkbox is activated. If deactivated, the user cannot make erroroneous inputs and is not confused by an overloaded dialog.

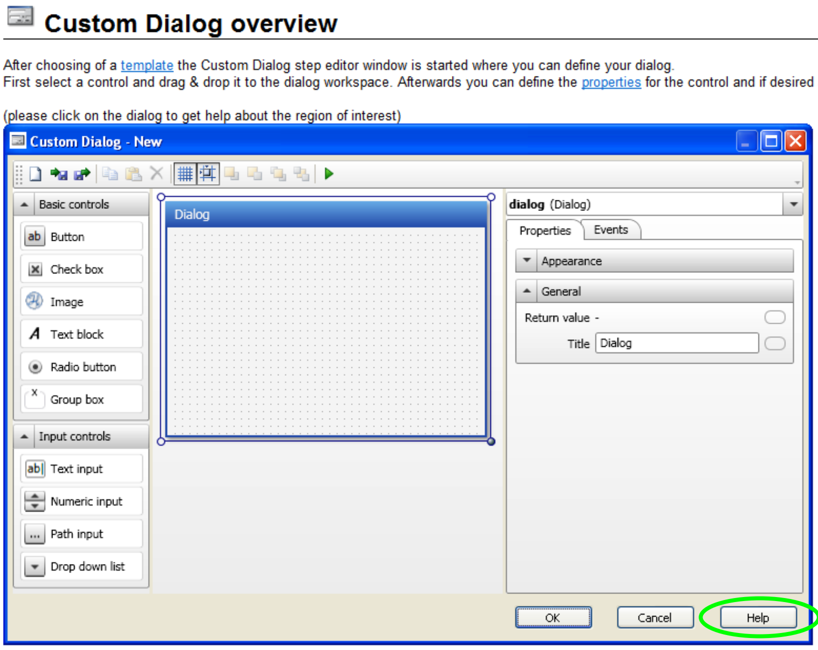


Getting Help for the Custom Dialog

Available within the Help of the dialog are the explanations about the following:

- how to link variables
- handle events on elements and
- change the behavior at runtime

Simply click on the **[Help]** Button at the bottom-right corner. This will open the "**Custom Dialog Overview**", where a simple click on the desired area will activate the related Help section.



11.3 Dispense on the Fly

VENUS Software offers a dispense on the fly (Single Step). This step does not stop the x-movement while pipetting which allows for a very fast dispensing of liquid, e.g. to fill a plate with reagent.

ATTENTION

This feature is available for 1000 µL-pipetting channels and 5mL-pipetting channels on Microlab STAR, STAR^{V} and VANTAGE.

This feature is also available for MagPip channels on Microlab STAR^{V} and VANTAGE.

Refer to the dialog box shown below while reading this section. The following parameters of the dialog box must be filled.

Sequence

The sequence positions where to dispense the previously aspirated liquid.

Sequence Counting

Select “**Automatic**” to update the current position of the sequence or choose “**Manually**” to not adjust the current position of the sequence.

Volume

The volume that will be dispensed per well. Note: The volume per pipetting channel is constant.

Dispense On the Fly Mode

Complete plate will sort the sequence accordingly.

Sequence order uses a sorted sequence (must fit within the selected pipetting channel pattern).

Labware Surface Distance

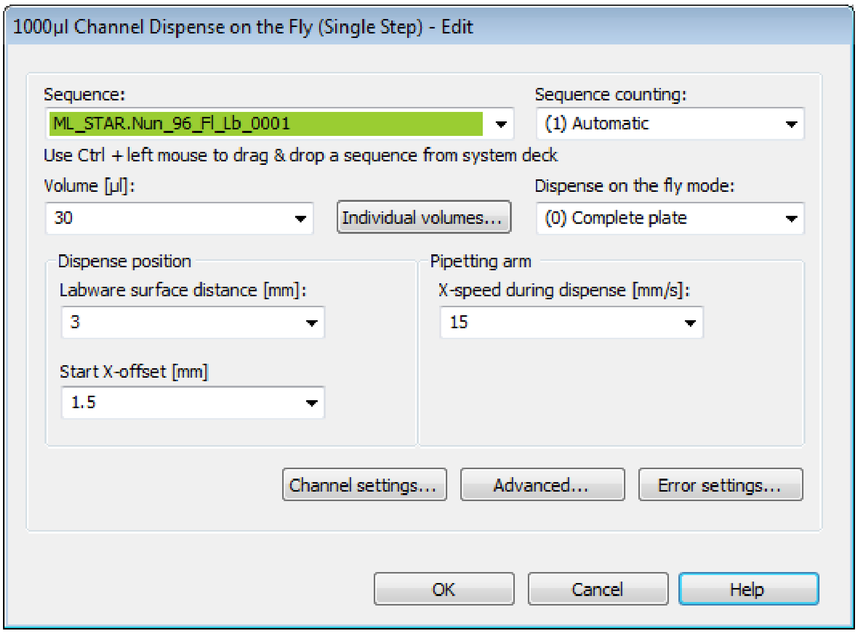
The position above the labware surface where the liquid is dispensed (travelling height).

X-Speed During Dispense

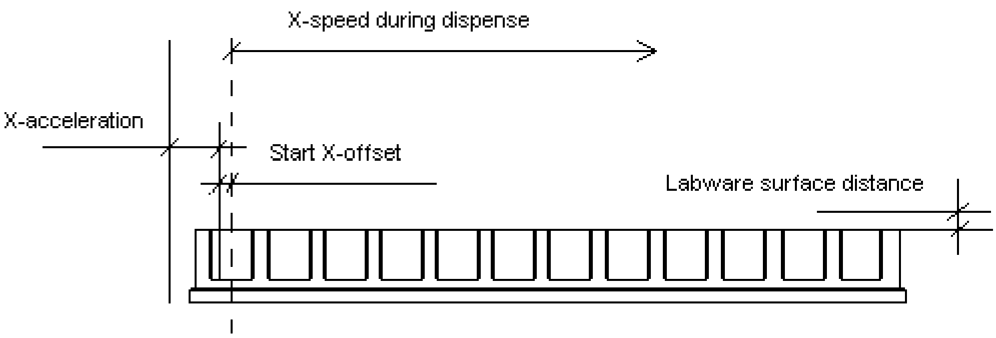
This value specifies the x-move speed. This value is dependent upon the volume to be dispensed, the flow rate in the liquid class and the distance between the wells.

Start X-Offset

The dispense normally starts in the center of the well. For larger volumes, shift the start point to the left to make sure the full volume is dispensed into the well.



The display below shows distances to be specified.



In the Advanced section, the following parameters can be changed. Please refer to the image below whil reading this section.

Liquid Class

It is possible to use a different liquid class than in the aspirations step.

X-Acceleration Distance Before First Shoot

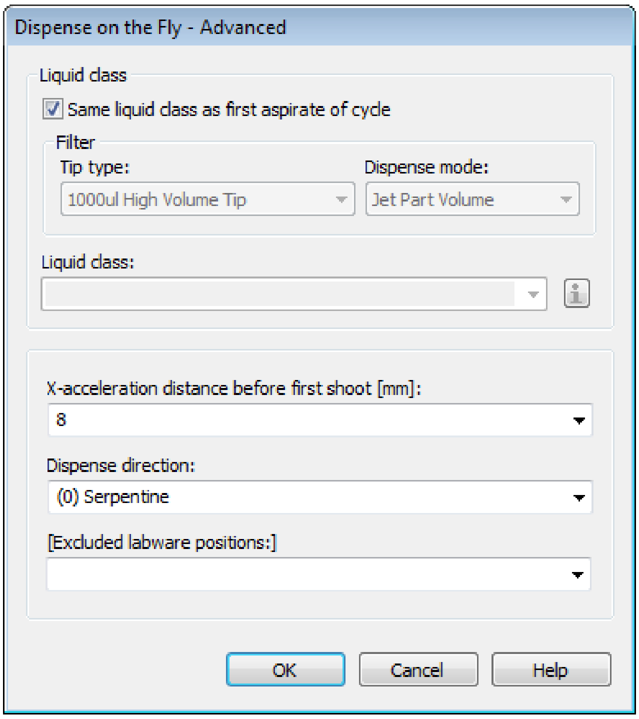
The distance at which speed for pipetting should be reached (see graphic above).

Dispense Direction

Choose “**Serpentine**” (for quick dispense) or “**From left only**”

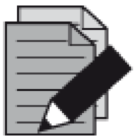
Exclude Labware Positions

This option allows to not dispense to specified positions.



Example: A1 / B1 must be left empty since it is a blank position on the reader plate.

Add a separate dispense step after dispense on the fly, using the dispense mode ‘Drain tip in jet mode’ to dispense the rest of the volume out of the tip.



NOTE

*Make sure that the liquid class in the aspiration is set to DispenseJet Part Volume, otherwise the “**Dispense on the Fly**” Step will display an error.*

*Run some tests on the instrument to properly adjust the x-speed during dispense. Set the value to “**0**” to let VENUS calculate the x-speed.*

The x-speed during dispense depends upon the volume, the liquid class parameters and the labware geometry.

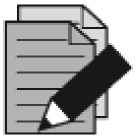
A table guide to find the correct settings.

Liquid: Water

Tip Size: High volume 1000ul, no filter

Liquid Class: High_Volume_Water_DispenseJet_Part

Shots per pipetting channel: 12 (full plate)



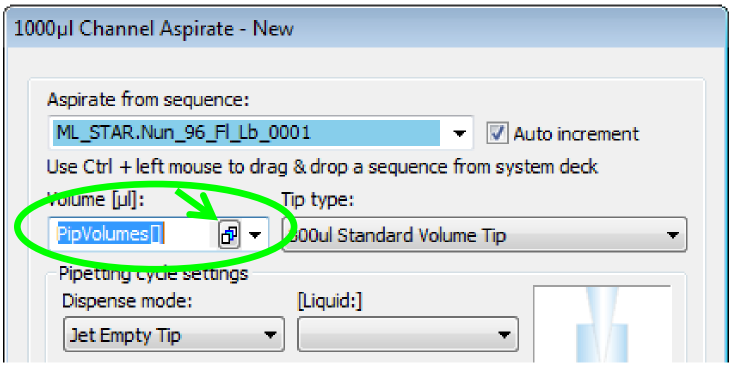
NOTE

Different tip sizes, liquid classes and shoot numbers may require different settings.

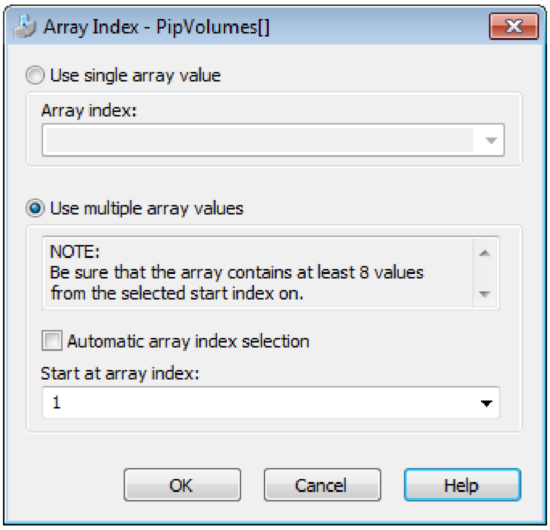
Volume	Speed	Start-X Offset
10	20	1
20	20	1
30	15	1.5
40	14	1.5
50	14	2
60	13	2
70	12	2.5
80	10	2.7

11.4 Arrays in Steps

Arrays can be used in steps, for example in fields like "**Sequence**", "**Volume**" and "**LC**". The existing arrays are listed together with the variables in the drop-down list. Once an array has been selected, the **[Array Index Dialog]** Button is displayed next to the data field containing the array (see green arrow).



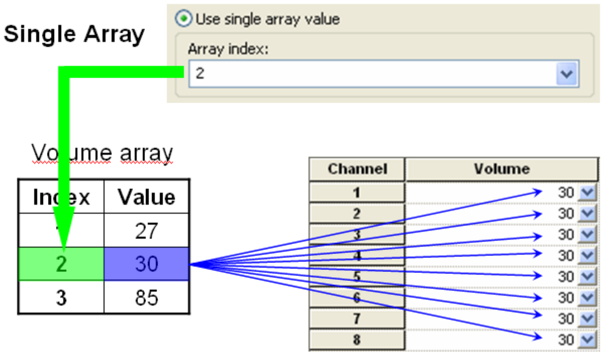
Clicking the button opens the "**Array Index Dialog**" as shown below.



In this dialog, the following options can be selected:

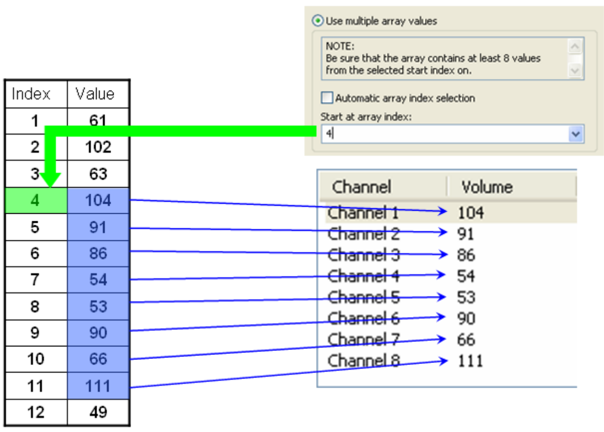
Use Single Array Value:

Enter the index of the specific array element (a digit or a variable). This selection is disabled when only a series of elements is allowed.



Use Multiple Array Values:

Enter an array start index, to select a series of array elements. For example, in the field of an aspiration step, to set the individual volumes per pipetting channel, starting at a specific index (see picture below).



- Activate automatic array index selection. This function is available for arrays of well-defined functions, where the usage of the array index was specified precisely. For example, a smart step processing some well-defined positions uses an array with all of the volumes of the individual positions. In this case, the step needs only the array without index, because the index is, for example, defined by the currently processed position.

11.5 1000 µL-Pipetting Channels versus 5 mL-Pipetting Channels

Working with the 5 mL-pipetting channel on the Microlab instrument is almost the same as working with 1000 µL-pipetting channels. The same CO-RE technology (in a different size) is used to pick up tips and tools, cLLD and pLLD can be performed and the same pipetting features (TADM, MAD, ADC) as on the 1000 µL-pipetting channels are available on the 5 mL-pipetting channels.

However, although the familiar appearance of the Single Steps and Easy Steps for the 5 mL-pipetting channel are kept the same, there are some differences to keep in mind when working with the 5 mL-pipetting channel.

Pipetting: 5 mL-Pipetting Channel Pattern

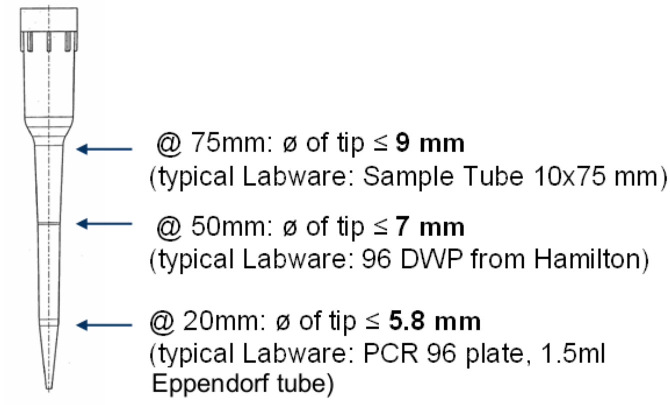
Due to its large capacity, the 5 mL-pipetting channel needs more space on the y-rail. Therefore, the closest distance from one tip end to the neighboring tip end is 18 mm (this is true if the pipetting channels are mounted on a double rail arm. On a single-rail arm, the distance is 36 mm). This has an influence on the sequence, for example, on plates.

Using the stamp tool for the 5 mL-pipetting channel, a sequence on a plate will look like this:

This means that four 5 mL-pipetting channels can work in parallel on a landscape plate.

Pipetting: Well / Tube Diameter

Depending on the immersion depth, the 5 mL-pipetting channel may collide with well or tube walls. See image guide below for useable diameters and heights, in oder to avoid hardware crashes.



11.6 Tip Handling Details

This section describes essential tip handling topics. Although the process of tip handling is stable and safe, some precautions must be considered when handling special labware.

VENUS Software offers a tip recognition feature based on the different tip geometries. It is available for both disposables and needles and it is activated during installation by a Service Engineer. This feature increases the Microlab instrument´s security, especially when different tip types (e.g., low- and standard volume tips) are used. In addition, all tip racks have color-coded labels and are bar-coded to be identified by the Autoload or ILD option (if installed).

For distinguishing disposable CO-RE tips, 50 µL and CO-RE tips and 300 µL, a special library is needed. Please consult a local Hamilton Representative.

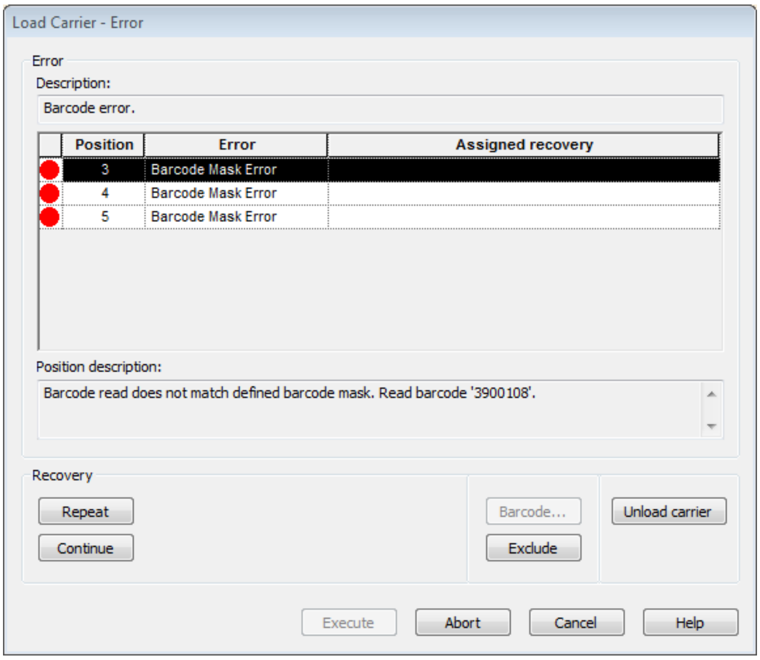
NOTE

All new or special tip types require additional settings (such as configuration file entries, liquid classes etc.) in the VENUS Software. Please consult a local Hamilton Representative for the implementation data of non-standard tips.

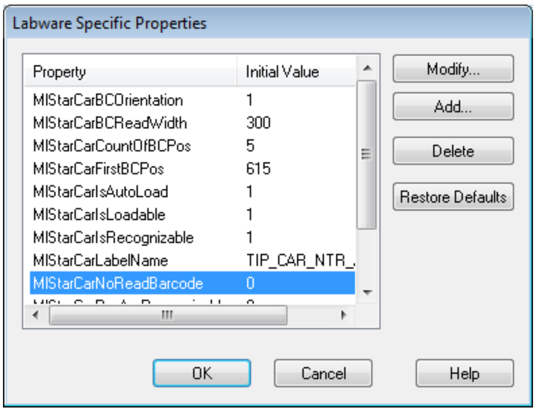
11.6.1 Tip Recognition with Different Tip Types on the Same Deck Layout

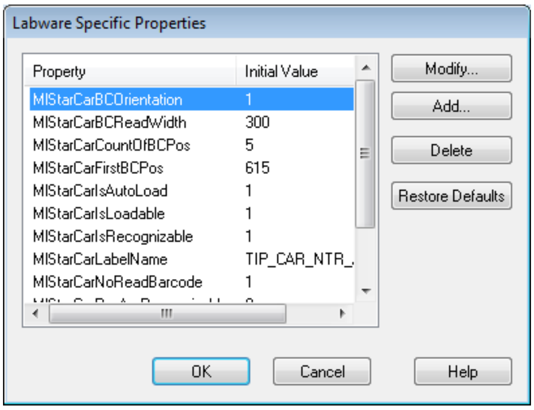
The tips used in a pipetting procedure must match with the suitable pipetting channel or pipetting head to prevent damage to the device. Proper precautions must be taken when loading tips in order to prevent confusion with the tips or loading incorrect tip types.

Whenever possible, use ***loading with Autoload*** (Micolab STAR) or ***loading with ILD*** (Microlab STARV and VANTAGE) and make sure that only tip carriers with the suffix “…BC…” (e.g. TIP_CAR_480**BC**_ST_A00) are used. In case of an incorrect tip rack, the system, while loading will prompt an error which looks like the image shown below.



When using another tip carrier (e.g. the 50 µL tip carrier) switch to the Deck Layout and change the properties MlStarCarBCOrientation to 1 and the MlStarCarNoReadBarcode to 0.





When ***loading manually***, pay extra attention to the tip types which are similar in design and cannot be distinguished by the tip recognition feature. When CO-RE tips, 1-50 µL and CO-RE tips, 10-300 µL are used on the same Deck Layout, a ***special library*** is needed. Please consult a local Hamilton Representative. In addition, check the label of the tip rack. The volume of the tips can be found at the top-left corner of the barcode label:

When ***loading manually*** and there is ***no library available*** for distinguishing the different tip types used, ***visual inspection*** of the barcode label is the only option. This is the case for Slim Tips, 10-300 µL used in combination with CO-RE Tips, 10-1000 µL.

NOTE

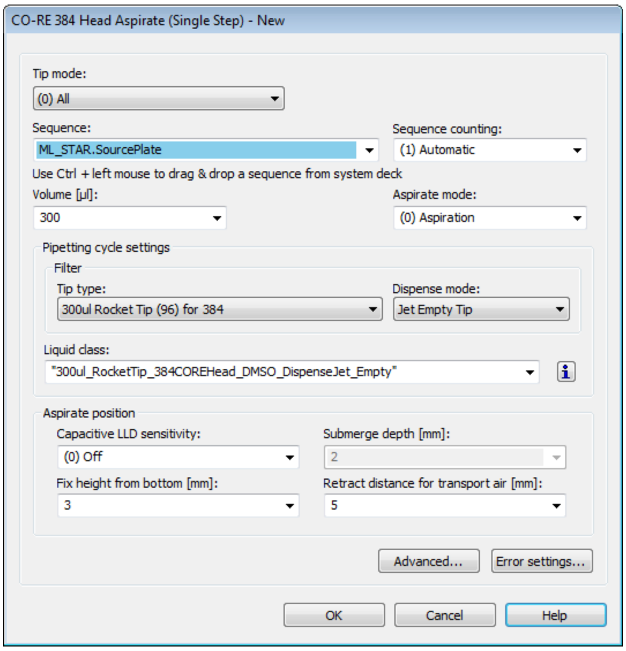
Slim tips and 1000 µL tips have the same head sizes therefore it can only be checked with the Barcode Reader.

11.6.2 Rocket Tips

It is possible to use Rocket Tips to pipette larger volumes than 50 µL using the CO-RE 384 Head. Rocket Tips combine a 300 µL tip body with a 4-channel head adapter.

To take advantage of Rocket Tips, the following settings have to be considered:

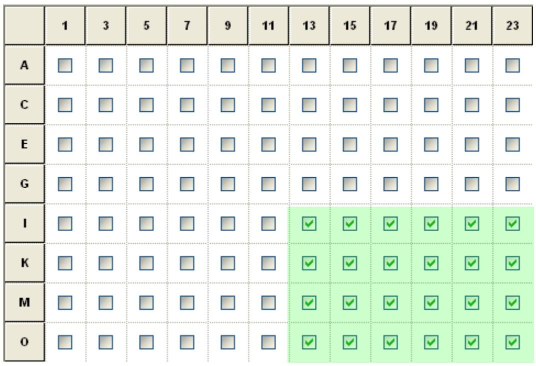
- Use the tip mode: (1) 96 Tips / Rocket Tips in all the steps → See NOTE
- Set the tip type to 300 µL Rocket Tip (96) for 384
- Use a Rocket Tip Liquid Class



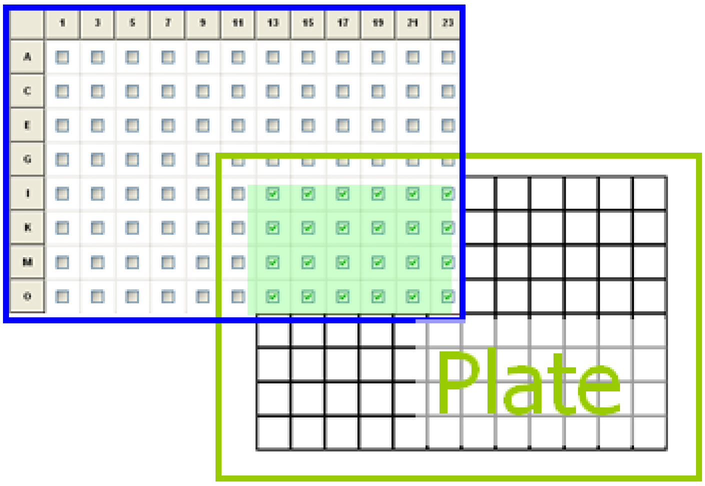
To pick up only single row(s), column(s) or a quarter:

- Use the reduced pattern mode: (1) Tips / Rocket tips in all the steps → See NOTE
- Click the "**Customize**" Tab and select "**Channel settings**"
- Select all, one channel, quarter, row(s) or column(s), depending on the pattern that will be used.

The picture below shows the head (**not the plate**) and which tips will be picked up.



The following is an example of how the head will be located over the plate if the plate sequence is sorted accordingly.



For more details on the different pattern, see Section 8.2 Stamp Tool.

NOTE

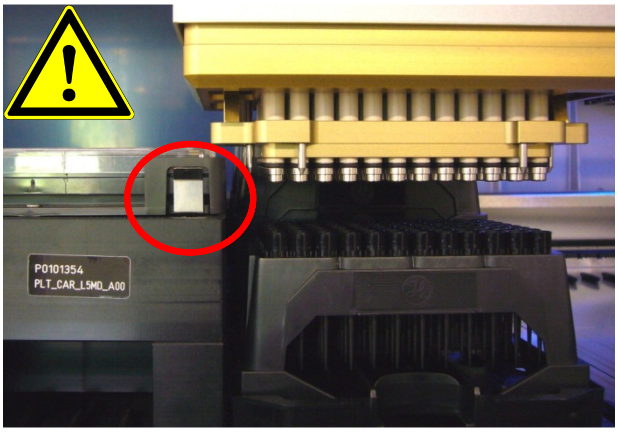
Rocket tips can only be used on the 50 µL CO-RE 384 Probe Head.

The reduced pattern mode has to be set in all pipetting steps.

- *When using Single steps: in the Tip Pickup, Aspirate, Dispense and Tip Eject*
- *When using Easy Steps: in the Easy Aspirate and Easy Dispense*

11.6.3 Tip Pick-up with the CO-RE 96 Probe Head

The Multi-Probe Heads need more space to pick up tips than the single pipetting channels. Therefore, it is important to use an appropriate Deck Layout. This will avoid collisions at run-time.



Example:

Picking up tips with the CO-RE 96 Probe Head out of e.g. the Nestable Tip Rack (NTR) will not work in the lower positions, if a high Plate Carrier, a Wash Station etc. is placed **directly beside** the Tip Carrier.

Solution:

Leave a space of **one track** to the left and to the right of the Tip Carrier → See white lines.

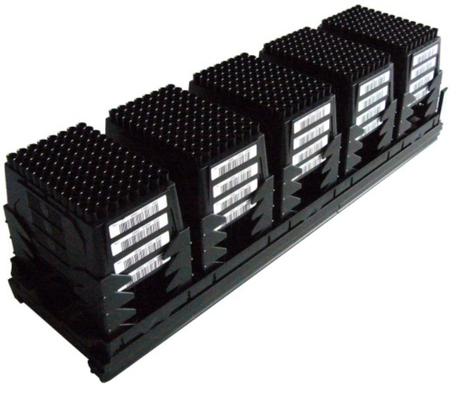
The Multi-Probe Heads are now able to pick up tips without collision with other Carriers.

NOTE

The same precautions have to be taken with all Tip Racks for the CO-RE 96 Probe Head / CO-RE 384 Probe Head and the single pipetting channels.

11.6.4 Nestable Tip Rack (NTR)

The Nestable Tip Rack (NTR) is designed to stack disposable Tip Racks which increase the number of disposable tips by a factor of 4 per SBS position, as compared with one-layer tip locations. NTRs come with 5 stacks of 4 layers each and have to be placed onto NTR Carriers of the Microlab instrument.



Tip pick up directly from loaded NTR Carriers is basically possible. This eliminates the need for intermediate tip pickup positions since pipetting tools such as single 1000ul pipetting channels as well as Multi-Probe Heads (MPH 96 and MPH 384) can pick up tips from the NTR stack in most cases.

A transport system (e.g. iSwap, CO-RE Gripper) is required to remove and dispose empty NTR frames away from NTR Stacks, to allow access with the Pipetting Tools to the next lower layer with filled NTRs.

Forcing Pipetting Tools to go down to lower or to the lowest layer of the NTR stack may have limitations, due to the required space between the Pipetting Tool and the neighboring Carriers, as well as its corresponding labware (also to other stacks, neighbour NTR's, etc.). In these cases, an intermediate Tip pick up position becomes necessary. This intermediate tip pickup position is designed and available as an NTR MFX Module (MultiFlexModule, MULTIFLEX NTR 96 MODULE, and MULTIFLEX NTR 384 MODULE).

The transport system as mentioned above must then be used to move NTRs to these intermediate Tip Pickup positions which enable constant tip pickup conditions in terms of the same X, Y and Z coordinates of a single NTR. Once the Tips out of the NTR frame of the intermediate tip pickup position are used up, the frame needs to be replaced by a new filled NTR.

When single-, row- or column-wise Tip pickup is required with Multi-Probe Heads, an additional tip pickup position equipped with a Tip Adapter becomes necessary. The Tip Adapter may be positioned onto a Tip Rack Carrier, as well as on a MFX Tip Module (MULTIFLEX TIP MODULE).

The recommended workflow is:

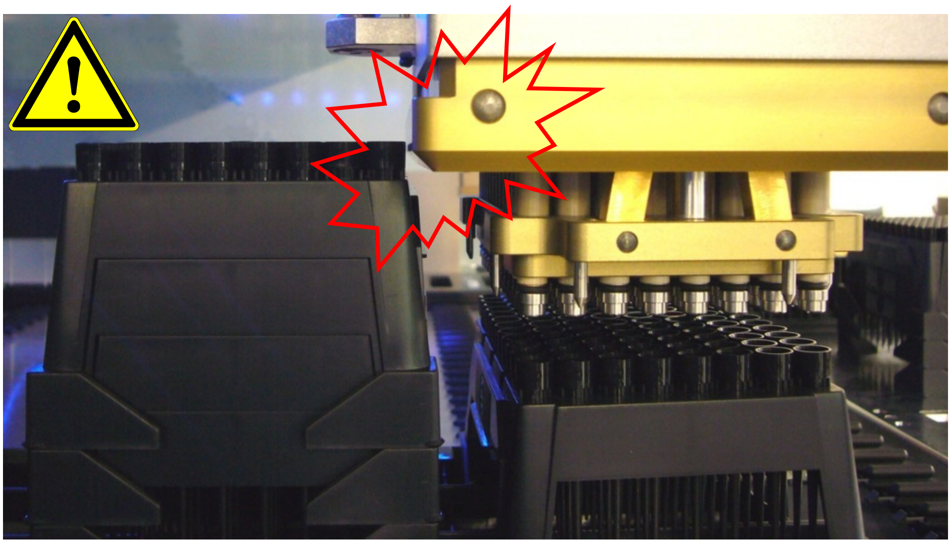
1. Transportation of one NTR from a loaded NTR Carrier to the intermediate Tip pickup position
2. Tip pickup with MPH out of the intermediate Tip pickup position
3. Tip eject into the Tip Adapter
4. As desired; single-, row- or column wise (shifted) Tip pickup out of the Tip Adapter
5. Ready for pipetting

NOTE

An empty NTR can be transported to the waste either by the iSWAP or the CO-RE Gripper. Note that the use of the CO-RE Gripper for transports to the waste needs a special library to grip the rack shifted. Otherwise, it is not possible to move the rack to the default waste position on an Microlab STAR and Microlab STARlet.

Working with the CO-RE 96 Probe Head and Nestable Tip Racks will need some precautions (see also Section 11.6.3 Tip Pick-up with CO-RE 96 Probe Head).

There is danger of collision when the CO-RE 96 Probe Head is picking up tips from lower positions if the neighboring positions are completely filled:



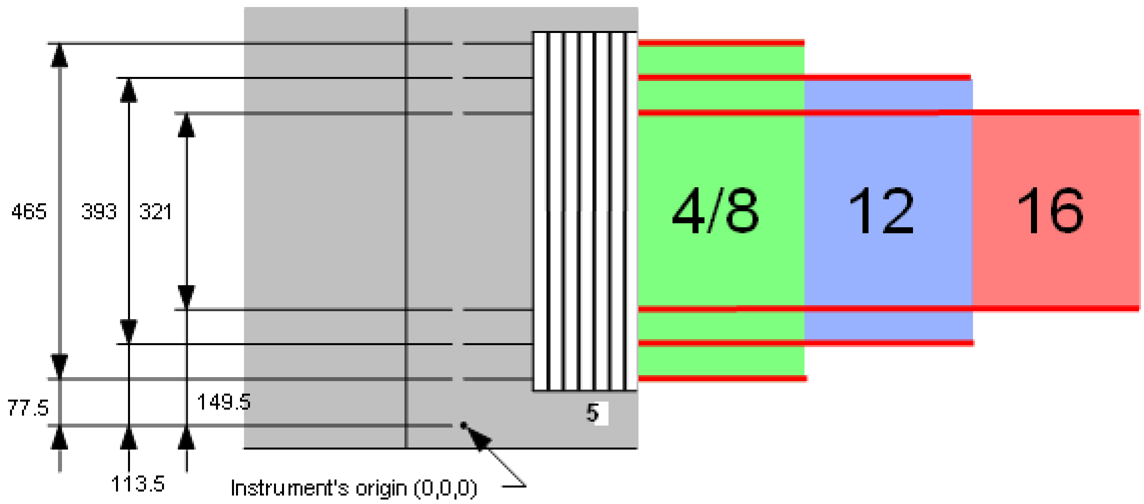
In this case, either work from top to bottom over all stacks or transport the Nestable Tip Rack to a collision-free position and pick up the tips there.

Pos 1 / Rack 4	→	Pos 2 / Rack 4	→	Pos 3 / Rack 4	→	Pos 4 / Rack 4	→	Pos 5 / Rack 4
Pos 1 / Rack 3	→	Pos 2 / Rack 3	→	Pos 3 / Rack 3	→	Pos 4 / Rack 3	→	Pos 5 / Rack 3
Pos 1 / Rack 2	→	Pos 2 / Rack 2	→	Pos 3 / Rack 2	→	Pos 4 / Rack 2	→	Pos 5 / Rack 2
Pos 1 / Rack 1	→	Pos 2 / Rack 1	→	Pos 3 / Rack 1	→	Pos 4 / Rack 1	→	Pos 5 / Rack 1

11.7 Random Access with 16 1000 µL-Pipetting Channels

A 4-Channel or 8-Channel instrument has random access to all positions of the deck. This means that pipetting channel No. 1 (rear-most) is able to aspirate from e.g. well H1 on a plate placed on the front-most plate carrier position. Having more pipetting channels results in a smaller random access area. This is because of the number of pipetting channels that must be moved over. Random access means that (e.g. due to a hit picking) the order of pipetting channels to proceed a sequence may be random.

Deck Areas for Random Access of Different Numbers of Pipetting Channels



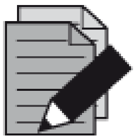
The 16-Channel Instrument is intended to be used as a batch-like processor. This means that all 16 pipetting channels should aspirate and dispense simultaneously, in order to allow maximum parallel organization and highest pipetting speed. In this case, the reduced random access space of a 16-Channel Instrument does not cause any problems.

To achieve a batch-like process, all sequences involved must consist of blocks of at least 8, or better, 16 positions. Sequence positions within a block must have:

- The same x-positions
- Decreasing y-positions from pipetting channel 1 (rear-most) to 16 (front-most)

To program a method using single steps:

- Select the option "**keep pattern**" within the "**Channel Settings**" Dialog in all single steps (tip pick-up, tip eject, aspirate, dispense). Do not use the option "**All sequence positions**".



NOTE

The limitation of the random access area on an instrument deck is also valid for instruments equipped with 5 mL-pipetting channels and any combination of 1000 µL-pipetting channels and 5 mL-pipetting channels. Other equipment such as the Tube Gripper or the camera channel will also reduce the area for random channel access.

Even if an error occurs and “**Exclude Channel**” is selected as an error recovery, the pattern will be kept.

To program a method using Smart Steps, perform the following:

- Select “**Copy Pattern**”
- Do not select “**Exclude Erroneous Positions**”
- If sequences can be reloaded (in run time during the pipette command), deselect the option “**reducible by user**” to maintain the original block-wise structure of the sequences

11.8 Dual-Arm Programming

The VENUS Software is designed to handle tools (e.g. a pipetting tool and a plate handling tool) on different arms. The software monitors each movement of the individual arms of the Microlab STAR, STAR^{V}, and VANTAGE dual arm instruments. If one arm needs priority to pass, the second arm is moved to a safe position first. In this manner, the two arms and the tools on the arms will not crash into each other. The programmer does not have to care about sudden collisions in the system.

VENUS Dynamic Scheduler

For the programming of a dual arm instrument, Hamilton recommends using the VENUS Dynamic Scheduler. Ask a local Hamilton Representative for more information or visit our Home page: www.hamiltonrobotics.com.

The Dynamic Scheduler has several advantages and is specially designed for handling complex parallels in an assay. The administration of other devices, such as Plate Handler, Plate Washer, Reader, etc., is just as easy as the programming of the dual-arm configuration.

Compared to the basic VENUS Software, there are advantages of the additional software package (VENUS Dynamic Scheduler), especially the programming of the parallel task. Each arm is defined as a so-called “**Resource**”. The Dynamic Scheduler allows for defining one resource (arm) as the major and the other as the minor device. The major arm always has the right of way. The Dynamic Scheduler automatically organizes the optimal use of the resources.

Dual arms can also be programmed by the basic VENUS Software of the Microlab instrument. Carriers used by both arms should be placed in the middle of the pathway reachable by both arms. If one arm is moving to the common Carrier(s), the other arm should not be obstructed in its tasks. Also, the programmer must decide whether the tasks of the two arms work in parallel or not (e.g., one arm must wait until the other has finished its task).

11.9 Sample Tracker

The Sample Tracker is a system for recording all actions related to labware and liquids (such as pipetting and moving plates). It can generate a file for each piece of labware on the Microlab instrument, recording information about its history (pipetting patterns, barcodes, pipetting errors, etc.) during one method.

With this information from the Sample Tracker, the user can:

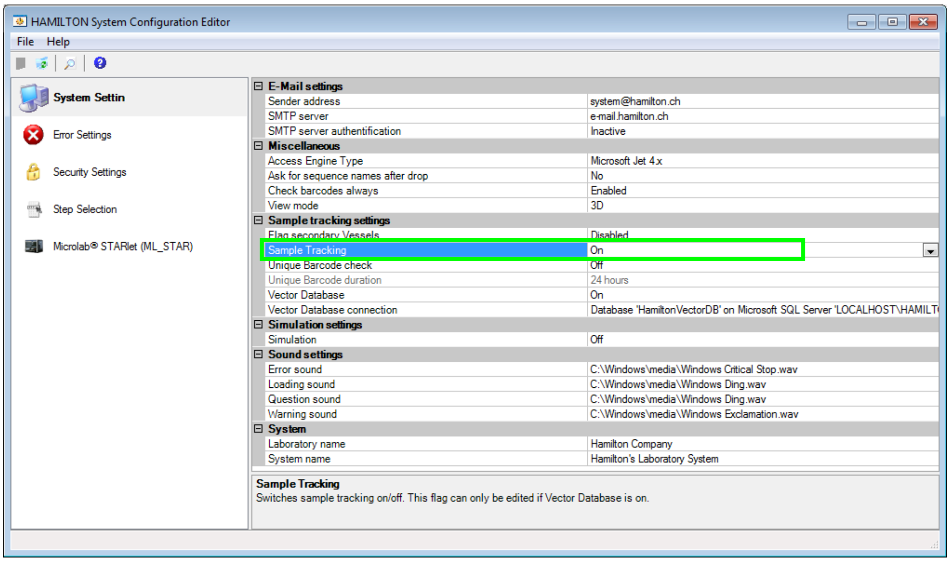
- Know what happened on the Microlab instruments during the runs
- Know whether the Microlab instrument did its work correctly or not
- Use the output to link it to a LIMS
- Use the output file for documentation (GLP)

NOTE

To configure, collect and visualize tracker data, use the Generate Mapping File step from the Data handling step collection.

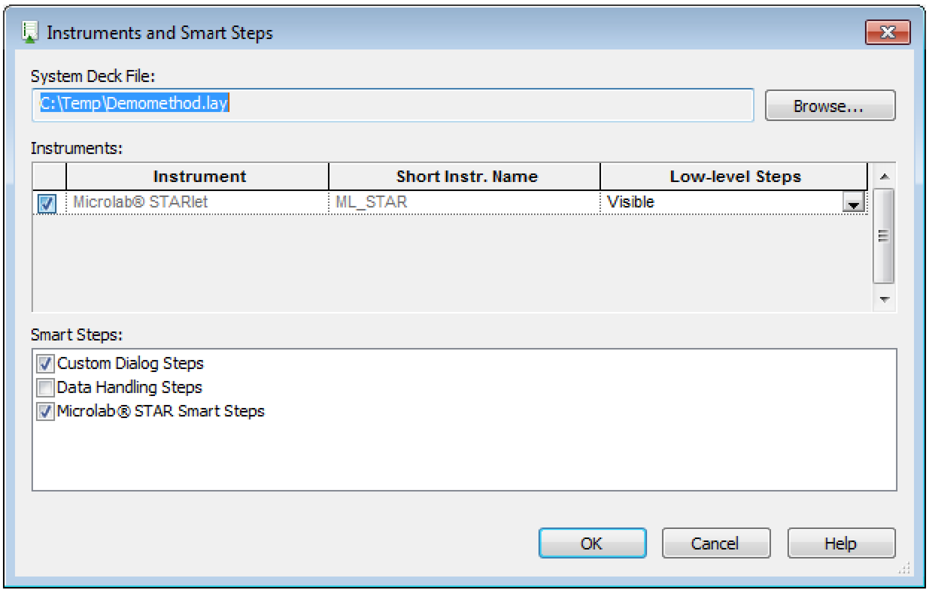
Enabling the sample tracker through the System Configuration Editor creates a database where all relevant information will be stored.

To change the settings, click from the Method Editor "**Tools → System Configuration Editor**" Menu. It can be also accessed from the VENUS application > System Tools > System Configuration Editor.

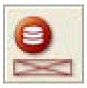


Mark the "**Data Handling Steps**" Checkbox when linking the instrument or layout, to be able to use the functions accessing the sample tracker information (see Section 7.2 Linking a Deck to a Method).

Once the access functions are activated for a method, the group "**Data Handling Steps**" in the toolbox window of the Method Editor will contain the sample tracker steps.



11.9.1 Data Handling Steps Toolbar

Command	Icon	Action Performed
Generate Mapping File		Generates a mapping file with information for selected sequence(s).
Set Labware Barcode		Set barcode for labware which was (or cannot be) read by the built-in reader.
Remove Labware		Remove labware data from the database stop generation of mapping information.
Import Worklist		Imports an external worklist into the internal Hamilton database. It works in combination with "Load and Match".
Update Job Status		Changes the status of a job on the Hamilton database.

11.9.2 Generating a Mapping File

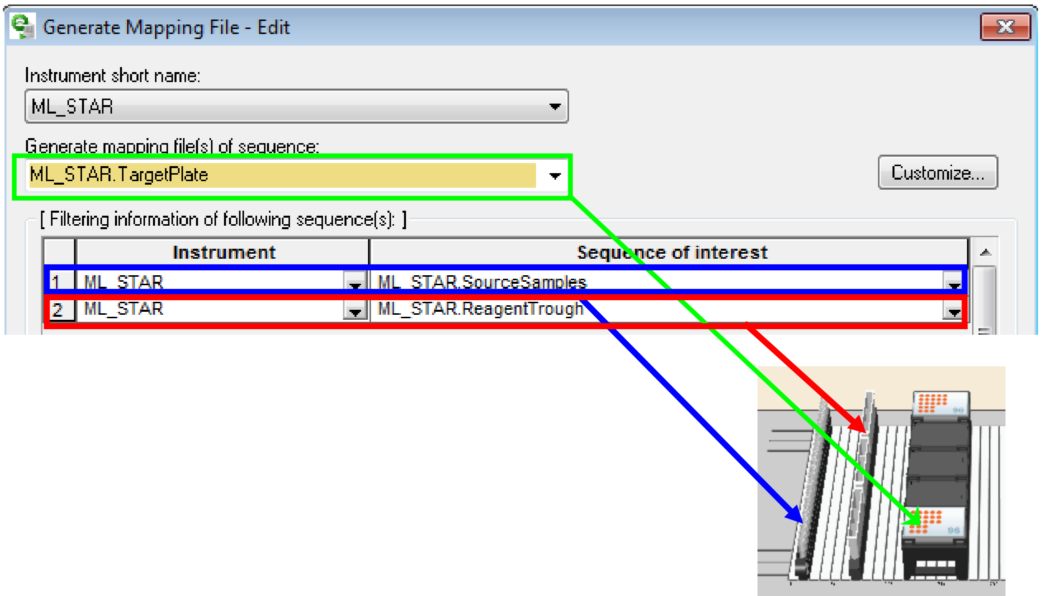
To use this step, select the following:

- instrument
- Sequence over which the data should be generated

The step is able to filter (expand) the stored information by selecting additional sequences in such a way that the additional information is stored which are affected by the sequences.

Define the path and filename where the collected information will be stored. If there is no path specified, it will automatically be stored in the **HAMILTON\Logfiles** directory by default.

Make sure that the labware is loaded before calling this command.



The "**Generate Mapping File**" Step collects data regarding the TARGET sequence and "**looks**" backwards as to what was pipetted into this sequence. Specify "**Sequences of interest**" to filter the information.

Example: Pipette from a tube carrier into the target plate (red to green). Afterwards, cover the samples in the plate with the reagent (blue to green).

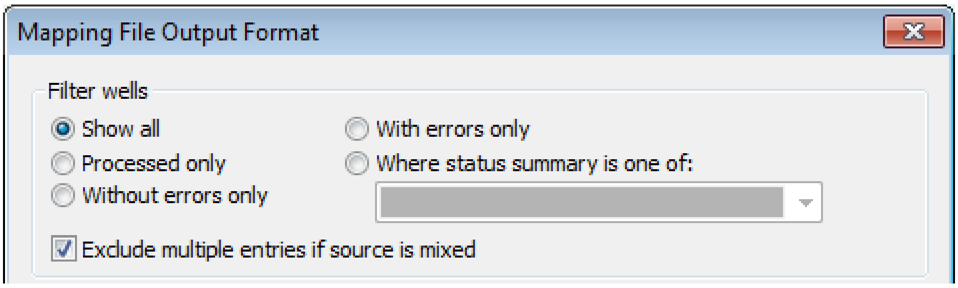
To track both pipetting, both of the source sequences have to be added as "**Sequences of interest**".

To link only the barcode of the sample tube to the well ID of the target plate, only add the sample tube sequence to the "**Sequences of interest**".

11.9.3 Customizing the Mapping File

The **[Customize]** Button offers multiple options to customize the output mapping file. It is possible to:

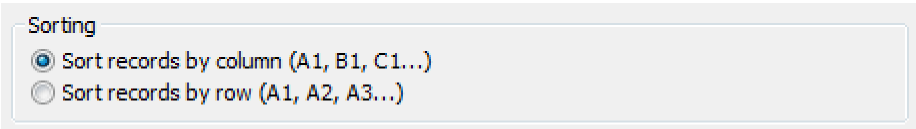
- Filter the data by using the status of the sample.



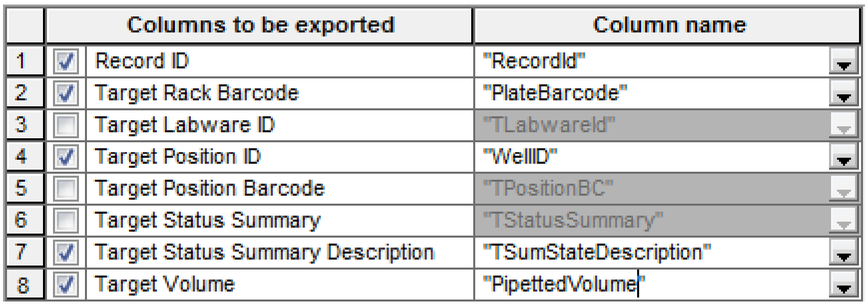
NOTE

*When the aspirate step is mixing the liquid, it will create an entry in the database. To suppress this liquid transfer, check the checkbox "**Exclude multiple entries if source is mixed**". Otherwise, at least two lines per sample will be found.*

- Sort the data in line wise (A1 – A12) or column wise (A1 – H1).



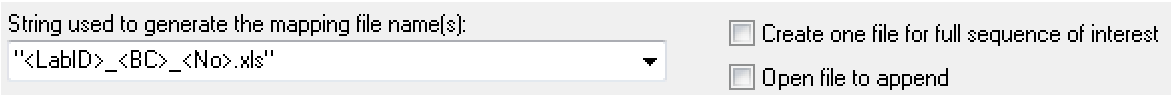
It is possible to link the default column name of the mapping files column to a user-defined column name (see lines 2, 4, 8 in the image below). Simply change the name in the right column. Make sure that the quotation marks are present. Alternatively, variables can be used in the column name fields.



		Columns to be exported	Column name
1	☑	Record ID	"RecordId"
2	☑	Target Rack Barcode	"PlateBarcode"
3	☐	Target Labware ID	"TLabwareId"
4	☑	Target Position ID	"WellID"
5	☐	Target Position Barcode	"TPositionBC"
6	☐	Target Status Summary	"TStatusSummary"
7	☑	Target Status Summary Description	"TSumStateDescription"
8	☑	Target Volume	"PipettedVolume"

- To hide columns in the output file, deactivate the checkbox (see lines 3, 5 and 6 in the image above).
- Activating the "**Create one file for full sequence of interest**" will create one single output file even if the sequence of interest covers more than one labware. Unchecking this box will create a single output file for every labware of the sequence of interest.

- The “**Open file to append**” will not overwrite an existing file with new data but add the new data to the end of the file.
- Per default, the output excel file is created in the LogFiles folder (under C:\Program Files\HAMILTON\.. → depending on the language of the Operating System). To change this path and the default name of the output file, use the field: “**String used to generate the mapping file name(s)**”.



11.9.4 Set Labware Barcode

The step “**Set Labware Barcode**“ is used for labware that has been positioned on the deck but cannot be identified by the built-in barcode reader.

The labware which the barcode has to be set must be identified either by the LabwareID (created in the Deck Layout Editor) or by the sequence which was defined by the Sequence Editor.

The barcode to be set must either be a constant string or a variable.

At run time, an error will be generated if the barcode passed is empty.

11.9.5 Remove Labware

The command “**Remove Labware**“ removes labware from the access of sample tracking. It is used for the labware which are removed from the deck. After this command has been executed, no more tracking data for the removed labware is accessible.

11.9.6 Import Worklist

The import worklist step is used to import an external database. The file format can be in Access, Excel or a structured text. The data in the external worklist can be assigned to the columns of the internal Hamilton database. This allows for the use of the imported data within the VENUS Software environment and e.g. use the “**Load and Match**” Smart Step to connect the loaded barcodes to a previously imported worklist. Refer to Section 11.11.3 Importing a Worklist for more information.

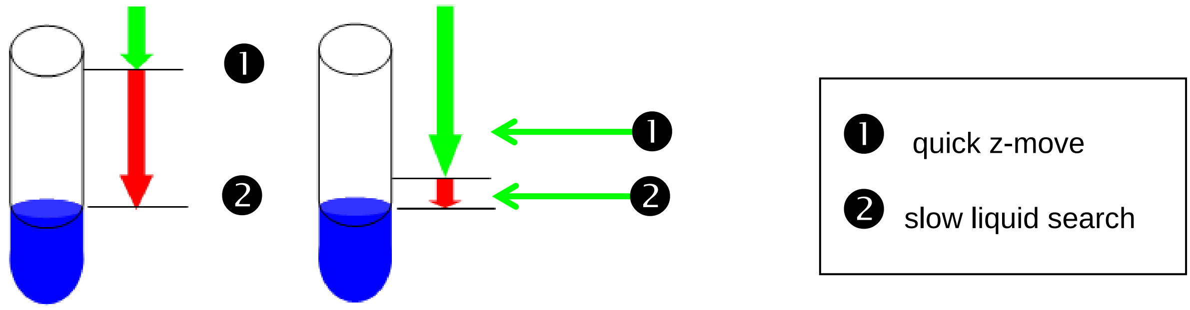
11.9.7 Update Job Status

This command allows for the changing of status of a sequence position from 'Assigned' or 'Unprocessed' to ‘Processed’. The update is done on the current position of the given sequence.

11.10 Speed Up cLLD Detection

Using the cLLD (capacitive Liquid Level Detection) can result in extended pipetting times when used on labware with high walls. The reason is that the cLLD search starts at the LLD Search height, which is defined in the labware, normally very close to the upper border.

Having only small amounts of liquid in such labware, the pipetting channel starts almost at the upper border and travels a long way down to find liquid (see picture below with the "**first pipette**" label).



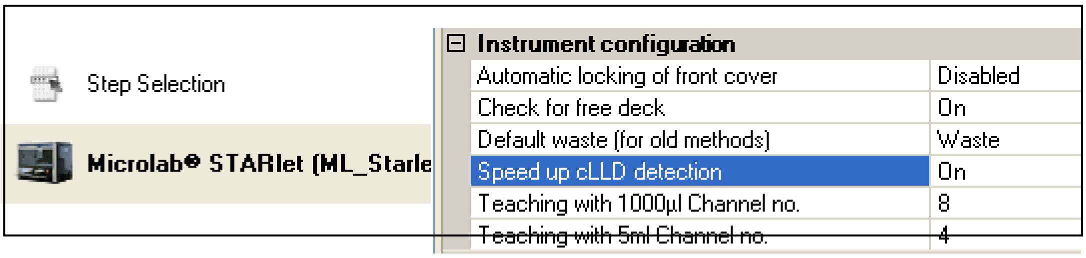
first pipette following pipettes

Using the "**Speed up cLLD detection**" Function, the following aspirate / dispense steps will use a calculated height from the previously detected level. This reduces the total time for pipetting due to the decreased search time.

To enable Speed up cLLD detection apply the following settings:

Set the Sample Tracking in the System Configuration Editor (System Settings) to **ON.**

Set the Speed up cLLD Detection in the System Configuration Editor (Instrument) to **ON**.



Make sure that the labware is higher than 40 mm. The smaller labware will not benefit from this feature.

Make sure to use the cLLD in the desired steps (aspiration and/or dispense).

NOTE

The system will calculate the amount of liquid in the tube when using the instrument. That means that when pipetting into a tube with pipetting channels, CO-RE 96 Probe Head or other pipetting devices, the level will be adapted to the new height.

*If the tube is filled up manually, the level is NOT adapted. The cLLD search in the liquid may start already and prompt an "**Unexpected liquid found**" error message.*

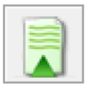
11.11 Working with Worklists

11.11.1 File Formats

Frequently, there will be a need to import and export worklists. The “**Open file**” Command of the method editor is designed to enable such file handling. The data source type (format) is identified by the extension of the worklist file name. The editor supports formats listed in the table presented below. No other extensions are supported.

Microsoft Excel	.xls
Microsoft Access Database	.mdb
Structured Text Files	.txt, .csv, .tab, .asc.
Unstructured Text Files	

11.11.2 Command Overview

Command	**Icon**	**Action Performed**	**Parameters to be Specified**
File: Open		Open a file before file access	File type, name, handle (reference to file), mode, data exchange variable and other, file-type-dependent parameters
File: Read		Read data out of a file*	File handle
File: Write		Write data into a file*	File handle
File: Set Position		Set the file-pointer to a specific position for reading	New position parameters
File: Close		Close a file after file operation(s)	File handle
Import Worklist		Imports worklist data from files into the supported database columns.	Wizard-guided; filename, filetype, delete old data Y/N, table name, column-mapping

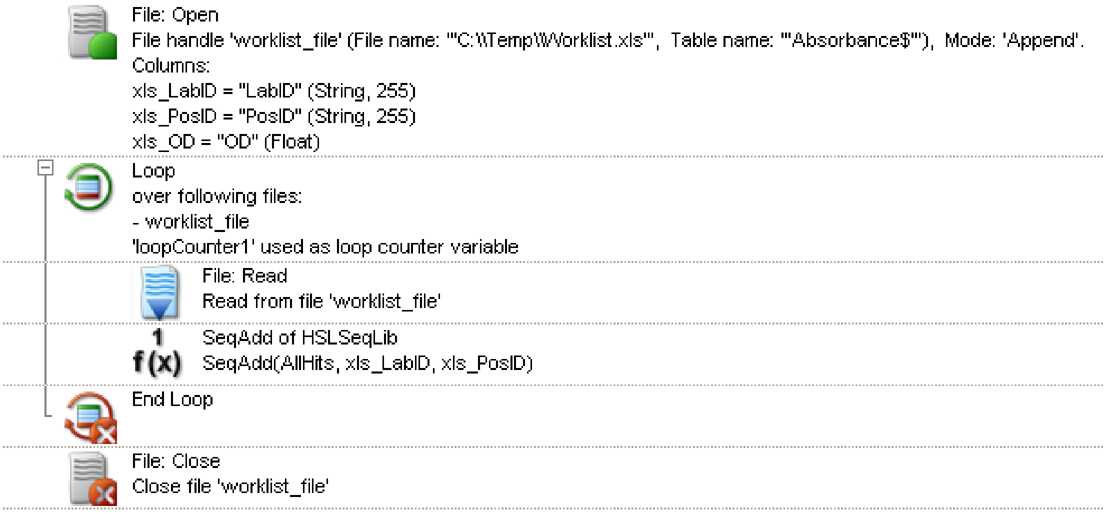
The FileRead / WriteSteps will read or write only ONE LINE of the file. To read through a complete file, use a loop.

The example below shows how to use the file commands. First, the file is opened by using the File: Open step. The path (C:\\Temp\\Worklist.xls) and sheet name (Absorbance$) are specified. Columns (LabID / PosID / OD) are linked to variables (xls_LabID / xls_PosID / xls_OD) respectively.

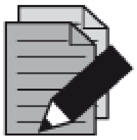
A loop is inserted to make sure that the file is read from the beginning to end. Inside the loop, the File: Read step reads line per line and assigns the values from the file into the variables.

In this case, the LabID and PosID are read out and are used to create sequence positions through the SeqAdd step.

To end, the file is closed (otherwise, it remains open and exclusively locked by the method).



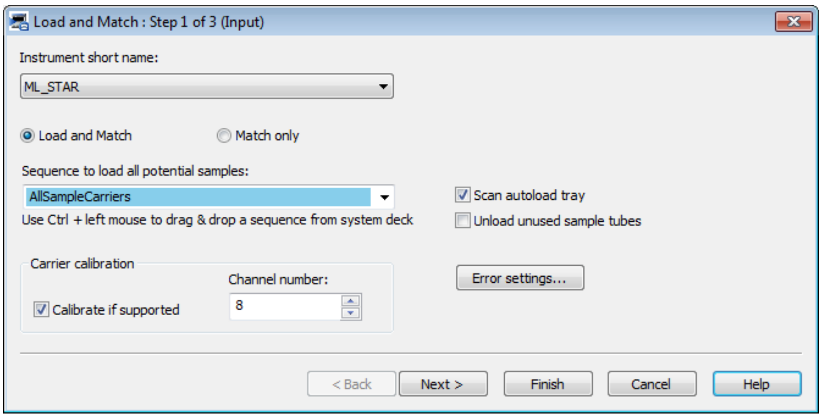
11.11.3 Importing a Worklist



NOTE

Once a Deck Layout has been linked to a method, the instrument-specific steps will be visible in the toolbox.

1. From the toolbox, drag the "**Load and Match**" Command into the method.
2. The "**Load and Match**" Wizard Step 1 will start as shown below.

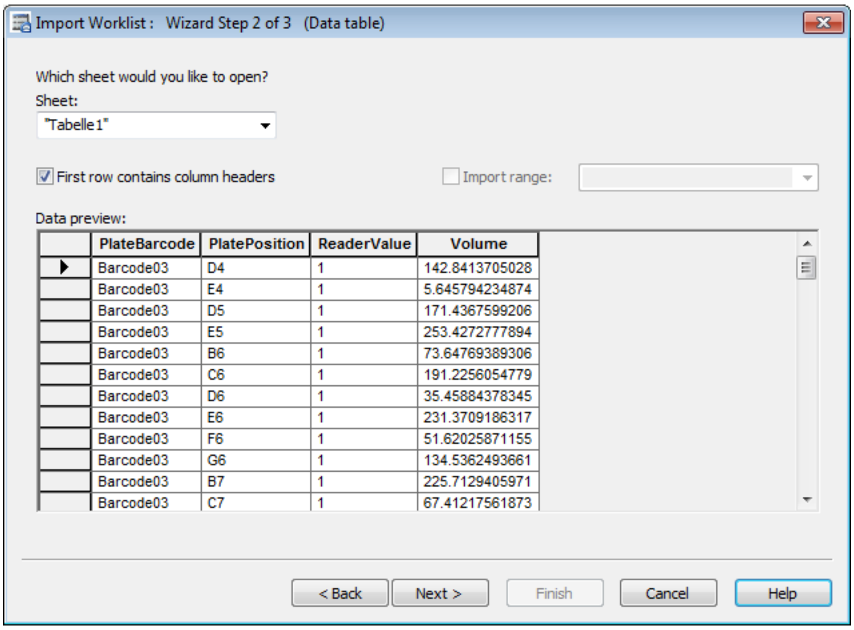


3. Select the instrument and the sequence to load all potential samples. To unload unused samples, which means samples without a worklist entry, check the box "**Unload unused samples tubes**".
4. Click **[Next >]** to continue to the next step of the wizard.

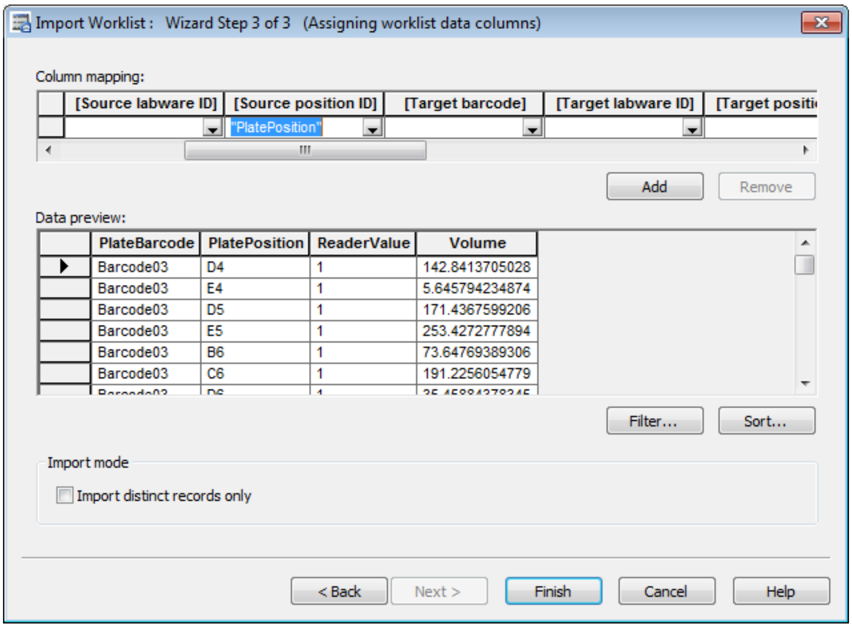
NOTE

*It is recommended to use fixed paths in the "**Worklist filename**" Field. Using fixed paths will allow a preview of the file. Using variables only does not allow a file preview.*

5. Specify the sheet to be opened (when using Excel) and check the data preview.



6. Link the columns of the worklist to the default columns of the database. If needed, use the filter options to restrict the amount of data. A sorting option is available to specify the order of appearance. Click **[Finish]** to complete the worklist import command.



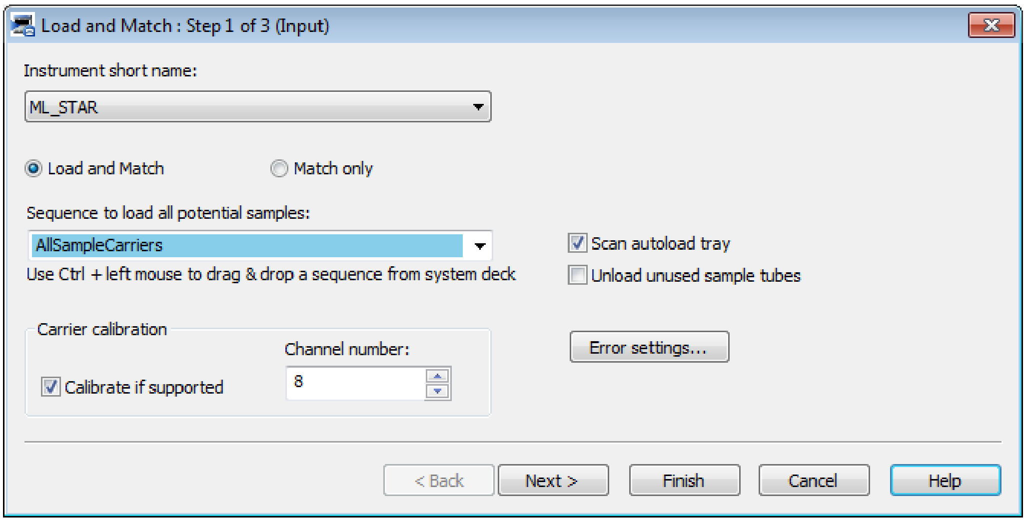
Importing a Worklist Example

This example shows the application of a worklist import.

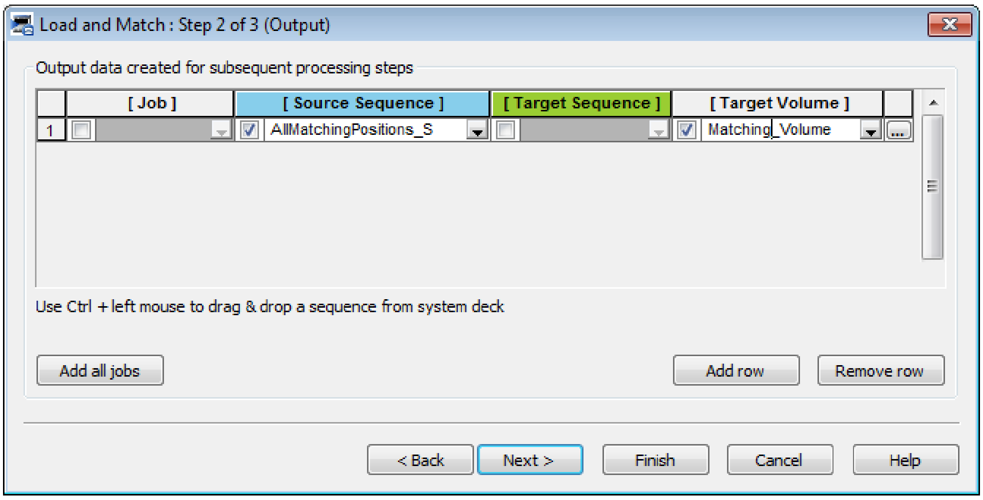
The “**Load and Match**” Command is used to load the Sample Carrier 1 and to read all barcodes. These barcodes are compared to the ones in the database (from the worklist). If the values match, a sequence position will be created in the “**AllMatchingPositions_Source**” Sequence into the target plate. The volume is taken from the array that was created in the Load and Match step. This array can be inserted directly in the Smart Step.

Import a Worklist

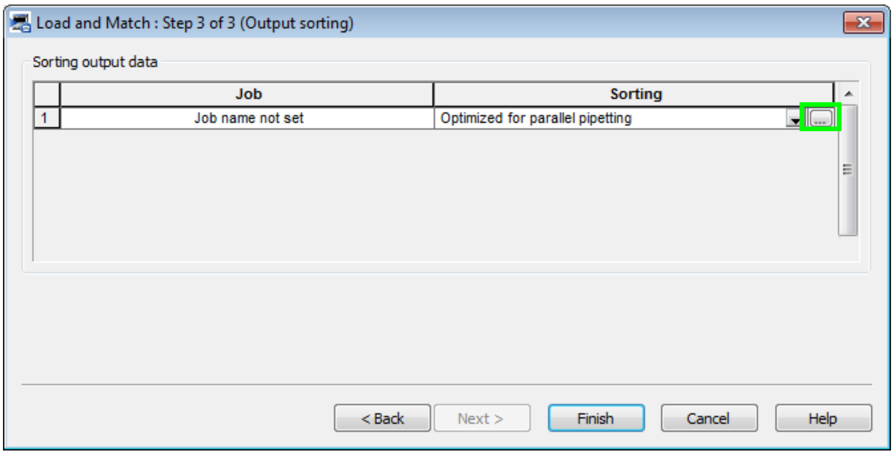
1. First, a worklist is imported with the command Import Worklist, as described above.
2. From the toolbox, drag the “**Load and Match**” Command into the method.
3. The “**Load and Match**” Wizard Step 1 will start, as shown below.



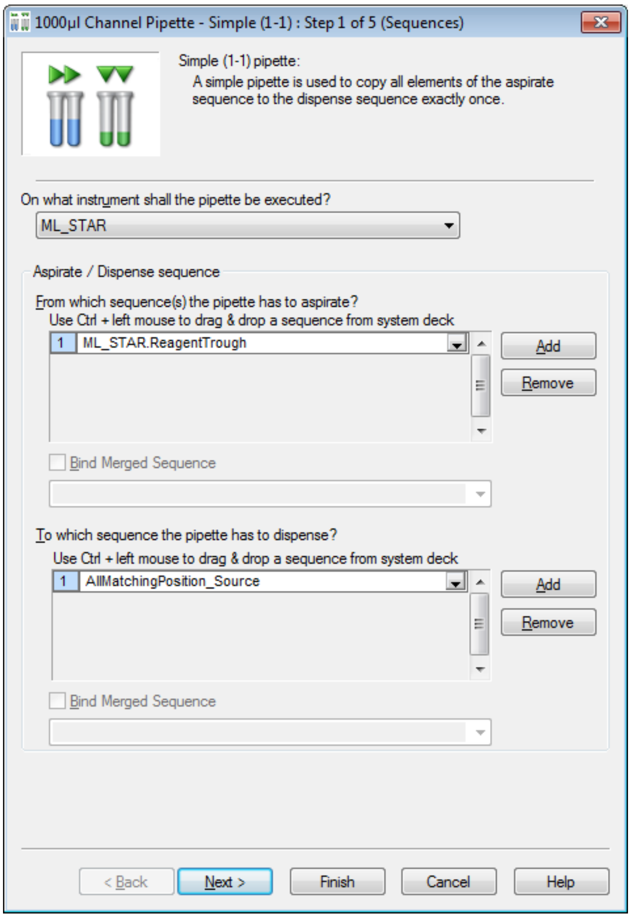
4. Select the instrument and the sequence to load all potential samples. To unload unused samples, which means samples without a worklist entry, check the “**Unload unused samples tubes**” Box. Click **[Next >]** to continue.
5. Select the source sequence ‘AllMatchingPositions_Source‘ with all matching positions and the target volume array ‘Matching_Volume[*]. Click **[Next >]** to continue.



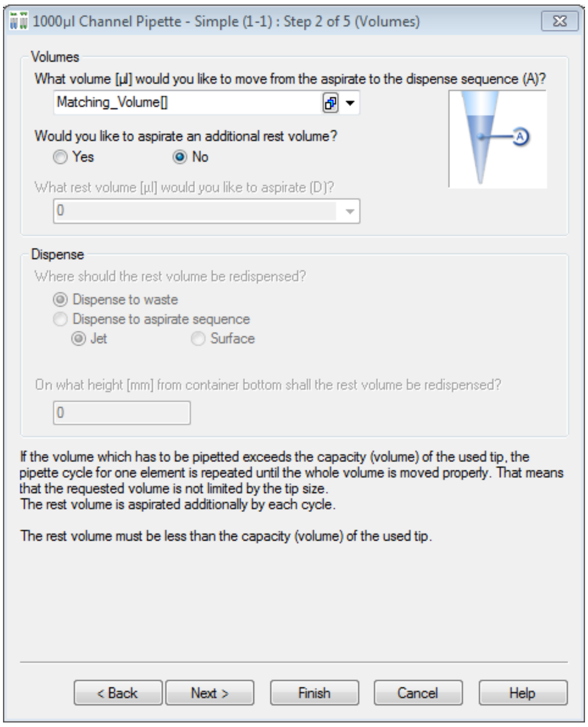
6. The "**Load and Match**" Wizard Step 3 will be shown.



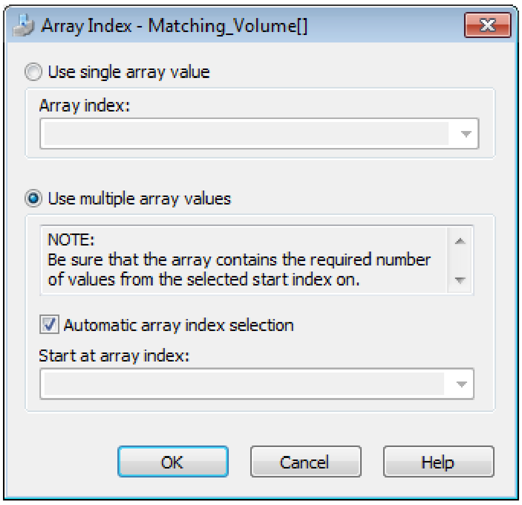
7. Click on the highlighted button shown above. Select the "**Optimized for parallel pipetting**" Sorting Method. Click **[Finish]** to end the wizard.
8. From the toolbox, drag the "**1000µl Channel Pipette – Simple (1-1)**" Command into the method.
9. The "**1000µl Channel Pipette – Simple (1-1)**" Wizard Step 1 will start.
10. Select the instrument "**ML_STAR**", the sequence to aspirate "**ML_STAR.ReagentTrough**" and the sequence to dispense "**AllMatchingPositions_Source**". Click **[Next >]** to continue.



11. The “**1000µl Channel Pipette – Simple (1-1)**” Wizard Step 2 will be shown.

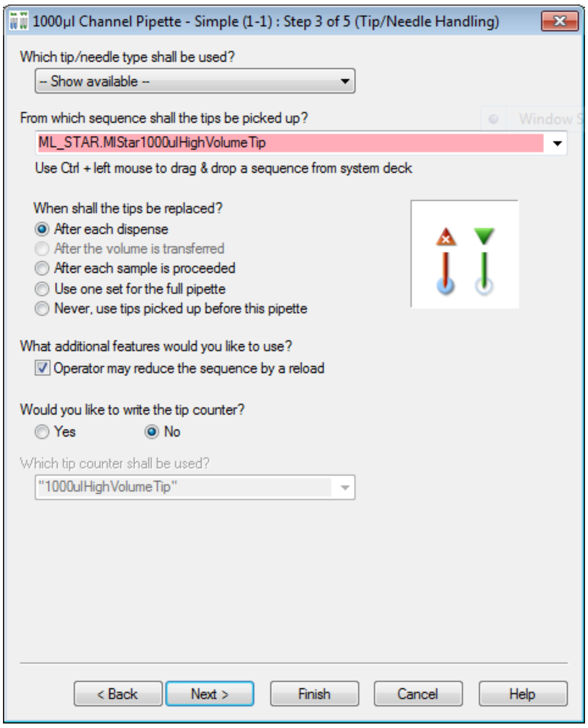


12. Select the volume array “**Matching_Volume**” and click on the array assistant to select the ‘Use multiple array values’. Make sure the Automatic array index selection box is ticked.

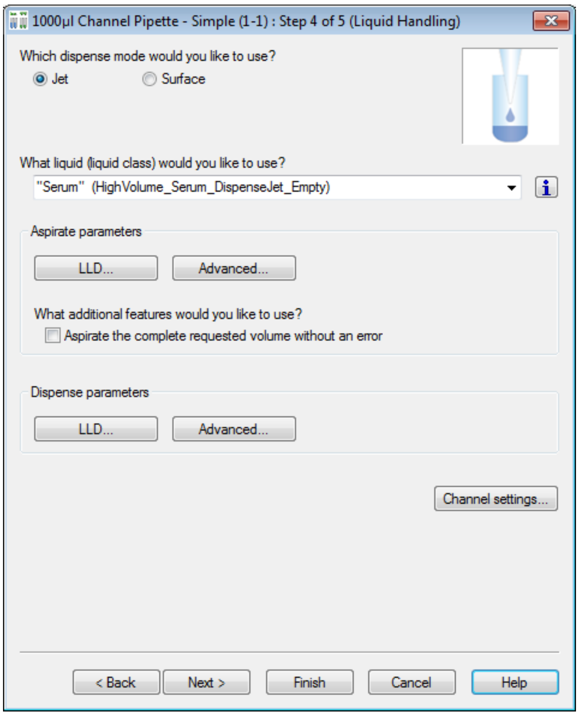


13. Click on **[OK]** to close the window.
14. Select **[No]** to the question if an additional rest volume should be aspirated.
15. Click **[Next >]** to continue.

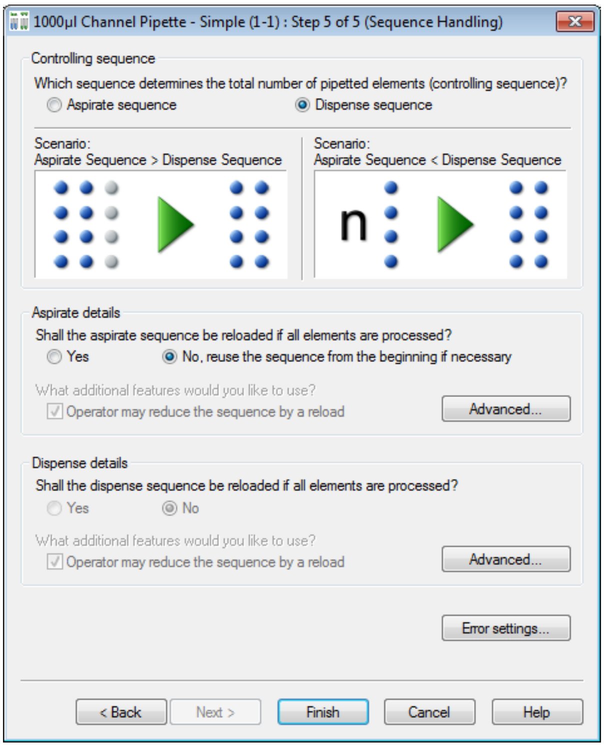
16. The "**1000µl Channel Pipette – Simple (1-1)**" Wizard Step 3 will be displayed.



17. Select the desired tip type. The Tip Sequence Window will display all available sequences of this tip type.
18. Decide if there is a need to replace tips and activate the corresponding radio button. Leave the rest of the values in the dialog with the default settings. Click **[Next >]** to continue.
19. The "**1000µl Channel Pipette – Simple (1-1)**" Wizard Step 4 will be activated.



20. Select the desired dispense mode and an applicable liquid class from the list. Leave the rest of the values in the dialog with the default settings. Click **[Next >]** to continue.
21. The "**1000µl Channel Pipette – Simple (1-1)**" Wizard Step 5 will be shown.



22. Select the "**Dispense**" Sequence as the controlling sequence (to finish pipetting after the AllMatchingPositions_Source sequence is used up). Leave the rest of the values in the dialog with the default settings. Click **[Finish]**, to complete the Pipette – Simple (1-1) command.

11.11.4 Worklist Handling with Microsoft Excel

Specifying a subset of the available data can be done when a Microsoft Excel workbook file is open. It is possible to open one of the following:

- A single worksheet
- A named range anywhere in the workbook
- An unnamed range in a single worksheet

The following table lists the conventions for these settings.

To open this object to read...	**...use this syntax**
Entire worksheet in a workbook file	Specify the sheet as sheetname$, where sheetname is the name of the worksheet. Important: A worksheet name must end with a dollar sign ($).
Named range of cells in a worksheet or workbook file	Specify the named range as NamedRange, where NamedRange is the name assigned to the range in Microsoft Excel. Important: The range in Microsoft Excel must be named before attempting to open it.
Unnamed range of cells in a single worksheet in a workbook file	Specify the sheet to be opened as sheetname$ and the range as FirstCellInRange:LastCellInRange. For example, to access cells A1 through Z256 in a worksheet called Sales, Sales$A1:Z256 is specified.
To open this object to write...	**...use this syntax**
Entire worksheet in a workbook file	Specify the sheet as sheetname, where sheetname is the name of the worksheet. Important: Do not end the worksheet name with a dollar sign ($).
To append to an object	**...use this syntax**
Append to an existing worksheet in a workbook file	Specify the sheet as sheetname$, where sheetname is the name of the worksheet. Important: A worksheet name must end with a dollar sign ($).
Append to a not yet existing worksheet in a workbook file	Specify the sheet as sheetname, where sheetname is the name of the worksheet. Important: Do not end the worksheet name with a dollar sign ($).

NOTE

A value in a range that exceeds the maximum number of rows, columns, or sheets for the worksheet or workbook cannot be specified. For more information on these values, refer to the Microsoft Excel documentation.

The format of all entries within a column in the Excel file must be identical: text, number, etc.

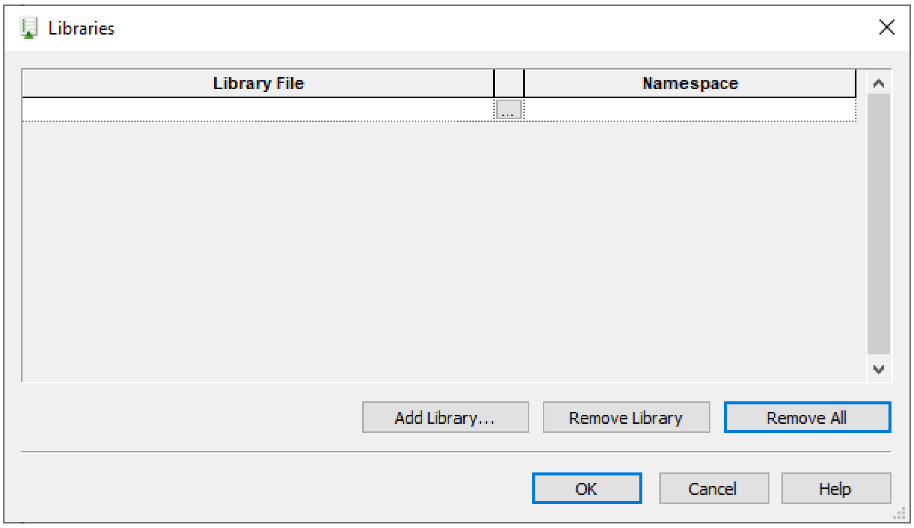
The handling (open/read/write/… file) of Excel files using the VENUS Software files does not require the installation of Microsoft Excel.

11.12 Using Libraries

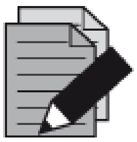
A library is a collection of standard functions which the programmer can choose from when writing methods, in addition to the toolbox elements. The best-known functions are the mathematical functions. For these, VENUS Software provides a mathematical function library. There are several other libraries that can be found to be useful. If the library required hasn't been installed yet, install it through "**Tools → Hamilton Support Software...**".

To use library functions in a method, include the appropriate library. Follow these steps:

1. Select "**Method → Libraries...**" in the Labware Editor to open the dialog seen below.
2. To add a library, click the Browse Button **[...]** at the center of the table highlighted below or click the **[Add Library...]** Button.



3. In the file selection box, select one or more required libraries to be imported. Click **[OK].**
4. Click **[OK]** to close the dialog. The selected library functions are available under their own group tab in the toolbox and can be used like standard commands.
5. To remove libraries, either click **[Remove All]** or select the libraries to be removed and click the **[Remove Library]** Button.



NOTE

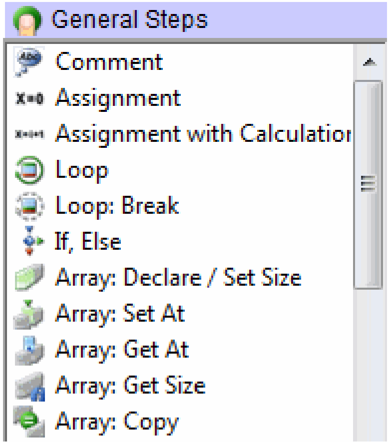
If a method uses at least one function of a library, the library cannot be removed.

VENUS Software supports two kinds of libraries: HSL Libraries (*.hsl) and Sub-Method Libraries (*.smt).

A library function can also be self-written and can be used like any standard library through the following:

- Graphical Method Editor (to create a Sub-method library)
- HSL Method Editor (to create an HSL library)

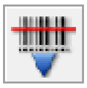
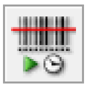
11.12.1 Array Library



An Array Library is an old library. Its functions should no longer be used. All array-related steps can be found in the General Steps. Arrays can be used directly in most steps.

11.12.2 Barcode Library

The HSLBarcodeReader library supports the Metrologic Barcode Scanner Orbit/MS7120 for reading operations over the serial interface RS232.

Command	Icon	Action Performed
Del Com Buffer		Deletes the input buffer.
Read		Gets read barcode from Metrologic barcode scanner Orbit/MS7120.
Set Com Port		Sets the serial port on which the Metrologic barcode scanner Orbit/MS7120 is plugged.
Set Error Recovery		Sets a flag that indicates how to react to a reading error
Set Simulation		Defines whether the Metrologic Barcode Scanner must be simulated instead of reading barcodes
Set Timeout		Sets the new timeout for all read operations of Metrologic barcode scanner Orbit/MS7120.

11.12.3 Deck Visualization Library

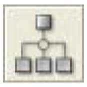
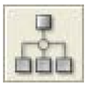
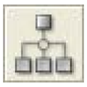
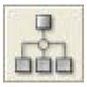
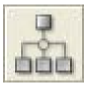
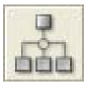
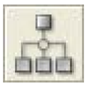
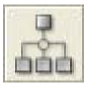
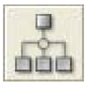
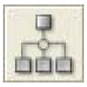
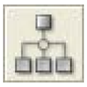
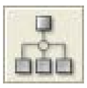
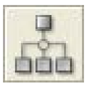
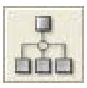
The deck visualization library updates lists and views.

Command	Icon	Action Performed
Update Loaded Labware		Updates the list of loaded labware and updates the view.
Update Used Labware		Updates the list of labware being used and updates the view.
Update Used Positions		Updates the list of loaded labware and updates the view.

11.12.4 Device Library

The device library offers a set of commands to collect labware and device details.

Command	Icon	Action Performed
DevAddContainerTo Rack		Replaces a container on a rectangular pre-loaded rack.
DevAddLabware		Adds given labware to Deck Layout using deck coordinates.
DevAddLabware ToTemplate		Adds given labware to deck site on named template.
DevAddPreloaded Labware		Adds given labware including any preloaded labware to Deck Layout using deck coordinates.
DevAddSequence		Adds a sequence to the collection holding the editable sequences of the device.
DevAddSequence2		Adds a sequence to the collection holding the editable sequences of the device.
DevCompute ContainerVolume		Calculates the volume in ml for the container at the given position and the given internal height. The function does NOT support connected containers.
DevCompute ContainerVolume2		Calculates the volume in ml for the container at the given position and the given internal height. The function supports connected containers.
DevCopyReset Sequence		Reloads a copy of the original deck sequence with the name sequenceName from the Deck Layout file into the sequence object sequenceObj. All indexes, limits and positions are re-initialized. The sequence must exist. The original deck sequence remains unchanged.

Command	Icon	Action Performed
DevEditSequences		Displays the "**Edit Sequence**" Dialog which shows the Deck Layout with all the sequence positions of the sequences set by the DevAddSequence() or the DevAddSequence2() functions. The "**Edit Sequence**" Dialog allows it to enabled / disabled sequence positions on the Deck Layout in a graphical manner by using the mouse.
DevEditSequences2		Displays the "**Edit Sequence**" Dialog which shows the Deck Layout with all the sequence positions of the sequences set by the DevAddSequence() or the DevAddSequence2() functions. The "**Edit Sequence**" Dialog allows it to enabled / disabled sequence positions on the Deck Layout in a graphical manner by using the mouse.
DevGetBarcodeData		Gets the barcode mask for the instance of labware at a position.
DevGetBarcodeData2		Gets the barcode masks for the instances of labware at positions.
DevGetBarcodeData3		Gets the barcode masks for the instances of labware at positions.
DevGetCfgValueWith Key		Returns the configuration value for the instrument mapped to a given key (integer, float, or string).
DevGetDeckLayoutFile Name		Returns the absolute Deck Layout file name (string; may be empty for a device without Deck Layout).
DevGetInstrument Name		Returns the property value for a property key of a labware.
DevGetLabwareData		Obtains the position of the given labware item from the Deck Layout using deck coordinates.
DevGetLabware Position		Obtains the position of the given labware item from the Deck Layout using deck coordinates.
DevGetLabware Positionex		Obtains the position of the given labware item from the Deck Layout using deck coordinates.
DevGetLabware Positionex2		Obtains the position information of the given labware item from the Deck Layout using deck coordinates.
DevGetPosition LabwareNameAt		Retrieves the template site with associated labware name or labware name with associated position name at a given index, retrieved during a previous call to the function Get Positions Labware Names.
DevGetPosition LabwareNames		Retrieves template sites with associated labware names or labware names with associated position names of all positions on the given labware which are referenced by the specified sequence.

Command	Icon	Action Performed
DevGetReleaseVersion		Returns the release version for the instrument, e.g. 3.0.0.630 (string).
DevGetSequence		Gets a copy of the deck sequence with the name sequenceName.
DevGetSequenceRef		Gets a reference to the deck sequence with the name seqId.
DevGetTemplate LabwareNameAt		Returns the labware name with associated template name at a given index.
DevGetTemplateLabware Names		Queries labware names with associated template name.
DevIsValidLabwareFor CurrentDeck-Layout		Checks whether the given labware is valid for the current Deck Layout.
DevPause		Suspends the program execution at the current position.
DevRemoveLabware		Removes given labware from the deck.
DevRemoveLabware FromTemplate		Removes given labware from named template.
DevRemoveSequences		Removes all sequences from the collection holding the editable sequences of the device.
DevResetSequence		Reloads the original deck sequence with the name seqId from the Deck Layout file, all indexes, limits and positions are re-initialized. The sequence must exist.
DevSetBarcodeData		Sets the barcode mask for the instance of labware at a position.
DevSetBarcodeData2		Sets the barcode masks for the instances of labware at positions.
DevSetBarcodeData3		Sets the barcode masks for the instances of labware at positions.

11.12.5 Error Library

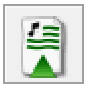
The error library contains functions to display and manage errors.

Command	Icon	Action Performed
ErrClear		Clears all property settings of the err object.
ErrCode		Extracts a numeric value that specifies the code of a phoenix error.
ErrDisplay Error		Displays an error in a message box.
ErrGetDataAt		Returns the element at the specified index from the data stored in the error object. If the index is 0, the function returns the size of the data stored in the error object.
ErrGetDescription		Returns a descriptive string associated with an error.
ErrGetId		Returns a numeric value that specifies an error.
ErrMajor		Extracts a numeric value that specifies the major ID of a phoenix error.
ErrMinor		Extracts a numeric value that specifies the minor ID of a phoenix error.
ErrRaise		Generates a run-time error. The error will be traced automatically.
ErrRaiseLast		Re-generates the current run-time error.
ErrSetDataAt		Sets the element in the data stored in the error object at the specified index (size of data not allowed to grow if 0 < index). If the index is 0, the function sets the size of the data stored in the error object.
ErrSetDescription		Sets a descriptive string associated with an error.
ErrSetId		Sets a numeric value that specifies an error.

11.12.6 File Library

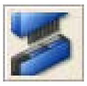
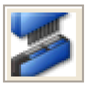
The file library contains all commands necessary for a method to handle files.

Command	Icon	Action Performed
FilEof		Indicates that the end of file (Eof) is reached.
FilFindFile		Starts a file search.
FilFindNextFile		Continues a file search from a previous call to FilFindFile.
FilFormatBarcodeFile		Converts the weakly-formatted barcode file, written as ASCII text file during the LoadCarrier operation, into a strongly formatted barcode file. The strongly formatted barcode file may be an ASCII text file, a Microsoft Excel, or a Microsoft Access file.
FilGetBinPath		Retrieves the path to the Hamilton\Bin folder.
FilGetCommState		Retrieves the configuration information for a specified communication resource. The entries of the structure that retrieves the configuration information must be accessible in the global scope.
FilGetCommTimeouts		Retrieves the time-out parameters for all read and write operations for a specified communication resource. The entries of the structure that retrieve the configuration information must be accessible in the global scope.
FilGetConfigPath		Retrieves the path to the Hamilton\Config folder.
FilGetLibraryPath		Retrieves the path to the Hamilton\Library folder.
FilGetLogFilePath		Retrieves the path to the Hamilton\Logfiles folder where runtime generated log files are stored.
FilGetMethodsPath		Retrieves the path to the Hamilton\Methods folder where all the methods are stored.
FilGetSystemPath		Retrieves the path to the Hamilton\System folder.
FilIsNull		Returns non-zero if the variable is a null value (SQL style Null).

Command	Icon	Action Performed
FilReadString		Reads the next record from the file data source as string-valued data. Row data, but no schema data, is saved to the string. After calling FilReadString, the next unread record becomes the current record, or the Eof property is set to hslTrue if there are no more records.
FilRemoveFields		Removes all fields from a record definition.
FilSearchPath		Searches for the specified file.
FilSetCommState		Configures a communication resource according to the specifications in a structure that contains the configuration information. Each entry in the structure is optional and overwrites the default value in parentheses.
FilSetCommTimeouts		Sets the time-out parameters for all read and write operations on a specified communication resource. The structure that contains the time-out information is as shown below. Each entry in the structure is optional and overwrites the default value in parentheses.
FilUpdateRecord		Updates the current record of the file object with the values of the variable objects specified in the record definition. The current record remains current after calling the FilUpdateRecord function. The provider must support UPDATE.
FilWriteString		Writes a string to the end of the file data source. After calling the FilWriteString function, the new record becomes the current record.

11.12.7 Microlab Instrument DC Wash Station Library

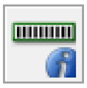
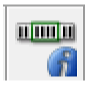
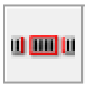
The "**HSLDcWasher**" Prefix has been removed for a better overview.

Command	Icon	Action Performed
WashSettings		Sets independent wash parameters for each DC wash station needle sequence.
NeedleWash		Washes the already picked up needles in the DC wash station.
NeedleWash2		Washes the already picked up needles in the DC wash station. Other than the NeedleWash step the NeedelWash2 step allows to specify further parameters used for dispensing the rest volume.
EmptyFillWashChamber		Either refills or to drains the wash chamber after last wash cycle.

11.12.8 Kit Lot Library

The "**HSLKitLotLib.hsl**" Library provides remarkable functions that handles barcodes of kit lots for the Microlab instrument.

The “**HSLKitLot**” Prefix has been removed, in order to provide a better overview.

Command	Icon	Action Performed
Check Barcode		Check if a barcode fits a particular mask.
Check Kit Lot		Check if kit lot contained in a barcode is valid.
Get Kit Lot Value		Get an arbitrary value from kit lot.
Set Access Repetition		Set the number of denied access attempts for a kit lot file.
SetDateColName		Define the name of the column in the kit lot file which contains the kit lot expiry date.
SetIdColName		Define the name of the column in the kit lot file which contains the kit lot identifier.
SetKitlotFile		Define the name of the kit lot file.
SetKitlotPath		Define the location of the kit lot file.
SetTableName		Define the name of the Excel sheet in the kit lot file which contains the kit lot information.
SetTimeColName		Define the name of the column in the kit lot file which contains the kit lot expiry time.
SplitBarcode		Extract a substring of a barcode.

11.12.9 Labware State Library

The HSL Labware State Library provides labware state management and labware controlling functions.

The "**HSLLabwState**" Prefix has been removed, in order to provide a better overview.

Command	Icon	Action Performed
AddLabware SourceBarcode		Adds an additional barcode as source barcode of the labware at the current position of the given sequence.
GetLabwareBarcode		Returns the barcode of the labware at the current position of the given sequence.
GetLabwareBarcode Mask		Returns the barcode mask of the labware at the current position of the given sequence.
GetLabwareExpNumOf ProcSteps		Retrieves the expected number of processed steps for the labware at the current position of the given sequence.
GetLabwareId		Returns the labware id for the current position of the given sequence.
GetLabwareLastAction State		Retrieves the action state of the last performed action for the labware at the current position of the given sequence.
GetLabwareLast SourceBarcode		Returns the last known source barcode of the labware at the current position of the given sequence.
GetLabwareNumOf ProcSteps		Retrieves the number of processed steps of the labware at the current position of the given sequence.
GetLabwareRelation		Returns the labware id of the chosen parent of the labware at the current position of the given sequence.
GetLabwareSource Barcodes		Returns an array with all source barcodes of the labware at the current position of the given sequence. This function uses the sample tracker. It will cause an error if sample tracking is disabled.
GetLabwareSummary State		Retrieves the current summary state of the labware at the current position of the given sequence.
GetLabwareVisibility		Returns the visibility of the labware at the current position of the given sequence.
GetLabwareVolume		Retrieves the volume of the labware at the current position of the given sequence.
Is Labware Connected		Returns hslTrue, if the labware at the current position of the given sequence is part of a connected container definition, else hslFalse.

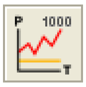
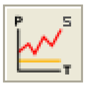
Command	Icon	Action Performed
Set Labware Barcode		Sets a new barcode for the labware at the current position of the given sequence.
Set Labware Barcode Mask		Sets a new barcode mask for the labware at the current position of the given sequence.
Set Labware Exp Num Of Proc Steps		Sets the expected number of processed steps for the labware at the current position of the given sequence.
Set Labware Summary State		Overwrites the current summary state of the labware at the current position of the given sequence.
Set Labware Visibility		Sets the visibility of the labware at the current position of the given sequence.
Set Labware Volume		Sets the volume of the labware at the current position of the given sequence.

11.12.10 Microlab Instrument Library

The “**HSLML_STAR**” Prefix has been removed, in order to a better overview

Command	Icon	Action Performed
AntiDropletControl_1000ulChannel_On		Activates the Anti-Droplet Control (ADC) on the 1000µl-pipetting channels.
AntiDropletControl_1000ulChannel_Off		Deactivates the Anti-Droplet Control (ADC) on the 1000µl-pipetting channels.
AntiDropletControl_5 mLChannel_On		Activates the Anti-Droplet Control (ADC) on the 5 mL-pipetting channels.
AntiDropletControl_5 mLChannel_Off		Deactivates the Anti-Droplet Control (ADC) on the 5 mL-pipetting channels.
AspirationMonitoring_1000ulChannel_Off		Disables the aspiration and clot detection monitoring with the ‘pLLD’ on the 1000µl-pipetting channels.
AspirationMonitoring_1000ulChannel_On		Enables the aspiration and clot detection monitoring with the ‘pLLD’ on the 1000µl-pipetting channels.
AspirationMonitoring_5 mLChannel_Off		Disables the aspiration and clot detection monitoring with the ‘pLLD’ on the 5 mL-pipetting channels.
AspirationMonitoring_5 mLChannel_On		Enables the aspiration and clot detection monitoring with the ‘pLLD’ on the 5 mL-pipetting channels.

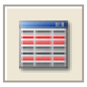
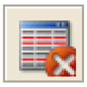
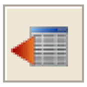
Command	Icon	Action Performed
ClotDetectionMonitoring_1000µlChannel_Off		Disables the clot detection monitoring with the 'cLLD' on the 1000µl-pipetting channels.
ClotDetectionMonitoring_1000µlChannel_On		Enables the clot detection monitoring with the 'cLLD' on the 1000µl-pipetting channels.
ClotDetectionMonitoring_5mlChannel_Off		Disables the clot detection monitoring with the 'cLLD' on the 5 mL-pipetting channels.
ClotDetectionMonitoring_5mlChannel_On		Enables the clot detection monitoring with the 'cLLD' on the 5 mL-pipetting channels.
CreateATBarcodefile		Creates an AT Barcode file for a MTP with the AT Barcode filter program.
DeleteBarcodeFile		Deletes an existing barcode file.
ExecuteWorklist		Executes a worklist written in the Gemini worklist file format (Gemini Worklist File Schema Information).
FormatBarcodeFile		Converts the mixed formatted barcode file 'barcodeFileName', written as ASCII text file by the LoadCarrier step, into a single formatted barcode file 'formattedBarcodeFileName' using the Formatted Barcode File Schema Information.
GetContainerVolume		Retrieves the volume [ml] of the container measured by the n-th channel during the previous MeasureContainerVolume operation.
GetInstrumentType		Gets the type of the connected instrument of type ML-STAR. Use this function only when no global device is defined.
GetInstrumentType_GlobalDevice		Gets the type of the passed instrument.
IsSimulatorMode		Probes whether the past instrument is in simulator mode or whether it is connected to a real instrument.
MeasureContainerVolume2		The MeasureContainerVolume function measures the volume [ml] of the container(s) beginning at the current position of a given sequence and using the given pipetting channel pattern.
SetBarcodeTypes		Sets the barcode types.

Command	Icon	Action Performed
SetPressureThreshold_1000ulChannel		Sets the pressure threshold for the aspiration and clot detection monitoring with the 'pLLD' on the 1000µl-pipetting channels.
SetPressureThreshold_5 mLChannel		Sets the pressure threshold for the aspiration and clot detection monitoring with the 'pLLD' on the 5 mL-pipetting channels.

11.12.11 Mapping Report Library

The HSL mapping report library provides functions to create mapped report files for the labware out of the sample tracking database within a method run.

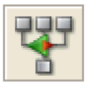
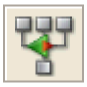
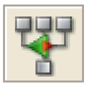
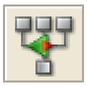
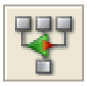
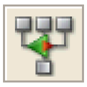
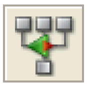
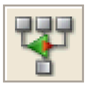
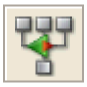
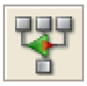
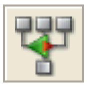
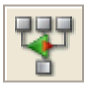
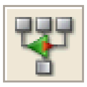
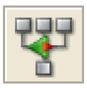
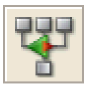
The "**HSLMAPREPORT**" Prefix has been removed, in order to provide for a better overview.

Command	Icon	Action Performed
AddFilterSequence		Adds the specified filter sequence to the list of filters influencing the report file formatter. Once one or more filter sequences are set, only information about positions that match with a position in one of the filter sequences appears in the mapping report file.
RemoveAllFilterSequences		Removes all filter sequences from the list of filters influencing the report file formatter.
GenerateMappingFile		Creates the report-mapping files.
GetListSeparator		Returns the list-separator, used as column-delimiter within the character separated value (.csv) report-mapping file.
DeleteFile		Deletes the specified file.
SetElementBarcode		Sets the barcode to the associated labware within the sample tracking database.
RemoveElement		Removes the given labware from the access of sample tracking.

11.12.12 Microlab Instrument Step Return Library

The step return library establishes functions to parse the result of the most frequently used Microlab instrument step return values.

The "**StepReturn**" Prefix has been removed on each command, in order to provide a better overview.

Command	Icon	Action Performed
GetErrorCode		Returns the error contained in the step return value (integer).
GetNumberOfPositions		Retrieves the number of positions (number of block delimiters) in the given step return value (integer).
GetPosition		Retrieves the num value for a specified position (block delimiter number) (integer).
GetMainError		Retrieves the main error for a specified position (integer).
GetSlaveError		Retrieves the slave error for a specified position (integer).
GetRecoveryButton		Retrieves the ID of the recovery button for a specified position (integer).
GetBarcode		Retrieves the barcode for a specified position (string).
GetBarcodeMask		Retrieves the barcode mask for a specified position (string).
GetBarcodePosition		Retrieves the barcode position for a specified position (string).
GetLastLiquidLevel		Retrieves the last liquid level for a specified position (float).
GetLabwareId		Retrieves the labware identifier for a specified position (string).
GetPositionId		Retrieves the position identifier for a specified position (string).
GetLabwarePositionIds		Retrieves the labware identifiers with associated position identifiers for a specified position (array of variables).
GetPositionFromNum		Searches a step return value for the first match of a specified num value (integer). Returns the one-based position (block delimiter number) of the first match of the specified num value.
GetStepData		Generic function to retrieve the step data for a specified position (string).

Command	Icon	Action Performed
SetFieldDelimiter		Sets a new field delimiter for the step return (initial value is ",").
SetBlockDelimiter		Sets a new block delimiter for the step return (initial value is "[").
SetFieldDelimiterEx		Sets the field delimiter for the step return as specified in the instrument configuration file (initial value is ",").
SetBlockDelimiterEx		Sets the block delimiter for the step return as specified in the instrument configuration file (initial value is "[").

11.12.13 Math Library

The math library establishes functions for both basic and advanced mathematical operations.

Command	Icon	Action Performed
MthACos		Returns the arccosine of a number.
MthASin		Returns the arcsine of a number.
MthATan		Returns the arctangent of a number.
MthBitwiseAND		Performs a bitwise conjunction on two expressions.
MthBitwiseNOT		Performs a bitwise NOT (negation) on an expression.
MthBitwiseOR		Performs a bitwise disjunction on two expressions.
MthBitwiseXOR		Performs a bitwise exclusive OR on two expressions.
MthCeiling		Returns the next highest integer that is greater than or equal to the specified numeric expression.
MthCos		Returns the cosine of an angle.
MthDec		Decrements the value of a number by one.
MthEqual		Returns a Boolean value indicating the result of the comparison.
MthERandDraw		Draws a new random number from the exponential distribution.

Command	Icon	Action Performed
MthERandInit		Initializes the exponential distribution
MthExp		Returns e (the base of natural logarithms) raised to a power.
MthFCeiling		Returns a float value representing the next highest integer that is greater than or equal to the specified numeric expression.
MthFloor		Returns the nearest integer that is less than or equal to the specified numeric expression.
MthFFloor		Returns a float value representing the nearest integer that is less than or equal to the specified numeric expression.
MthGreaterThan		Returns a Boolean value indicating the result of the comparison.
MthGreaterThanOrEqual		Returns a Boolean value indicating the result of the comparison.
MthInc		Increments the value of a number by one.
MthLessThan		Returns a Boolean value indicating the result of the comparison.
MthLessThanOrEqual		Returns a Boolean value indicating the result of the comparison.
MthLog10		Returns the base 10 logarithm of a number.
MthLogicalAND		Performs a logical conjunction on two expressions.
MthLogicalNOT		Performs a logical negation on an expression.
MthLogicalOR		Performs a logical disjunction on two expressions.
MthLog		Returns the natural logarithm of a number.
MthMax		Returns the greater of two supplied numeric (or string) expressions.
MthMin		Returns the lesser of two supplied numeric (or string) expressions.
MthNotEqual		Returns a Boolean value indicating the result of the comparison.
MthNRandDraw		Draws a new random number from the normal distribution.

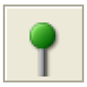
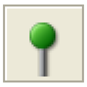
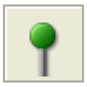
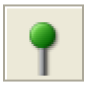
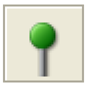
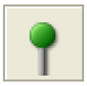
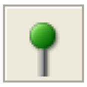
Command	Icon	Action Performed
MthNRandInit		Initializes the normal distribution.
MthPow		Returns the value of a base expression taken to a specified power.
MthURandInit		Initializes the uniform distribution.
MthURandDraw		Draws a new random number from the uniform distribution.
MthTan		Returns the tangent of an angle.
MthSin		Returns the sine of an angle.
MthShiftRight		The bitwise shift-right operator shifts its first operand to the right (>>) by the number of positions specified by the second operand.
MthShiftLeft		The bitwise shift-left operator shifts its first operand to the right (<<) by the number of positions specified by the second operand.
MthRound		Returns a number rounded to a specified number of decimal places.
MthR01Init		Initializes the random number generator.
MthR01Draw		Draws a new random number.

11.12.14 Object Library

The object library provides functions for handling automation objects.

The "**Object**" Prefix has been removed, in order to provide a better overview.

Command	Icon	Action Performed
CreateObject		Creates a reference to an automation object.
ReleaseObject		Releases the reference to an automation object.
IsNull		Returns true if the object is null (i.e. the object is not bound to an automation object); otherwise false.
PropertyGet		Obtains the value of the specified property of an automation object.
PropertySet		Sets the value of the specified property of an automation object.
Invoke0		Provides access to those functions exposed by an automation object that take no arguments.
Invoke1		Provides access to those functions exposed by an automation object that take one argument.
Invoke2		Provides access to those functions exposed by an automation object that take two arguments.
Invoke3		Provides access to those functions exposed by an automation object that take three arguments.
Invoke4		Provides access to those functions exposed by an automation object that take four arguments.
Invoke5		Provides access to those functions exposed by an automation object that take five arguments.
SetResultObject		Specifies an object to be used to store the result of the next call to one of the invoke functions.
ResetResultObject		Resets the result object previously set by a call to the SetResultObject.
SetNumberOfParameters		Sets the number of parameters of the collection holding the parameters to be passed in the next call to the Invoke function to a function exposed by an automation object. A maximum of 20 parameters is supported.
GetNumberOfParameters		Gets the number of parameters of the collection holding the parameters to be passed in the next call to the Invoke function to a function exposed by an automation object.

Command	Icon	Action Performed
AddParameter		Adds a new parameter to the collection holding the parameters to be passed in the next call to the Invoke function to a function exposed by an automation object. A maximum of 20 parameters is supported.
SetParameterAt		Sets the value of a parameter for a given index; number of parameters not allowed to grow.
GetParameterAt		Gets the value of a parameter for a given index.
RemoveAllParameters		Removes all parameters from the collection holding the parameters to be passed in the next call to the Invoke function to a function exposed by an automation object.
Invoke		Access to the functions exposed by an automation object.
EnumNext		Retrieves the next item in the enumeration sequence.
EnumReset		Resets the enumeration sequence to the beginning.

11.12.15 Report Library

The HSL report library provides functions to create report files for labware out of the sample tracking database within a method run (e.g.: a comma separated value result file (.csv) or an AT barcode file).

The "**HSLReport**" Prefix has been removed for a better overview.

Command	Icon	Action Performed
AddToReportList		Adds labware within the given sequence to an internal report list.
AddToReportListFromLabware		Adds labware within the given sequence to an internal report list.
Create ATBarcode File		Executes the Hamilton AT file filter program HxATFilter in quiet mode, to create the AT barcode file.

NOTE

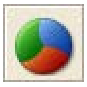
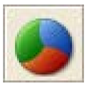
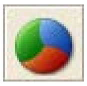
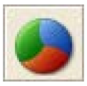
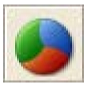
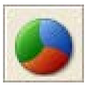
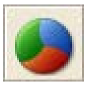
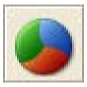
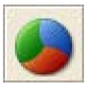
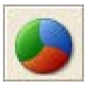
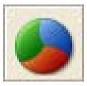
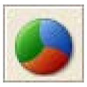
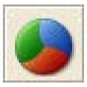
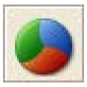
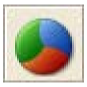
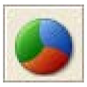
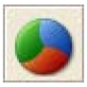
In creating an AT-Barcode file for vials containing more than one barcode (for example sample + reagent were dispensed into this well, both came from containers with a barcode), the last barcode in the pipetting order will be written into the AT-Barcode file.

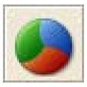
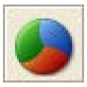
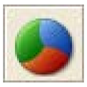
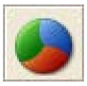
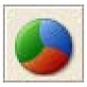
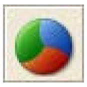
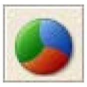
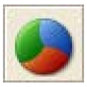
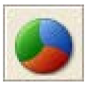
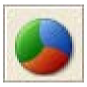
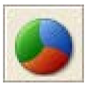
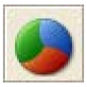
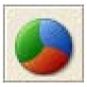
To prevent overwriting, a sample barcode of a reagent barcode (reagent dispensed after the sample) has to be deleted. To do this, set an empty string as a barcode for the container using the SetLabwareBarcode command from the HSLLabwareState library between loading and pipetting.

11.12.16 Sequence Library

The sequence library is very helpful for advanced sequence creation and handling.

Command	Icon	Action Performed
SeqAdd		Adds the labware item with the name labwareId and the positionId at the end of the sequence.
SeqAddEx		Adds the labware item with the name labwareId and the positionId at the end of the sequence. For Multi-Deck Layouts. There is an additional parameter called device context.
SeqCopySequence		Overwrites the state of the target sequence object with the state of another sequence.
SeqEdit		Provides a way to edit a sequence graphically.
SeqEdit2		Provides a way to edit a sequence graphically.
SeqEqualSequences		Determines whether the specified sequences are equal.

Command	Icon	Action Performed
SeqGetLabwareId		Returns the labware identifier of the item at the current position.
SeqGetLabwareIds		Retrieves the unique labware names of a sequence.
SeqGetMax		Returns the maximum number of positions/labware, which is allowed to be processed per step.
SeqGetName		Retrieves the name of the sequence.
SeqGetNext		Sets the current position on the next position in iteration order.
SeqGetPositionCountForCurrLabware		Returns the position count remaining for the current labware.
SeqGetPositionId		Returns the position identifier of the item at the current position.
SeqGetProperty		Gets at the specified sequence position the value associated with the specified property.
SeqGetTotal		Returns the total number of positions in this sequence.
SeqGetUsedPositions		Returns the number of positions/labware in a sequence, which has been processed by the last step.
SeqIncrement		Increments the current position by increment positions/labware in iteration order.
SeqInsertAt		Inserts given labware item after position pos.
SeqLookupPosition		Looks up a given labware position.
SeqMPH48SortLike96		Sorts a single or multi-(384) plate sequence in such a way that a 96-position pipetting pattern is preserved when pipetting with an MPH 48.
SeqOperatorAssignSeq		Assigns another sequence to the target sequence.
SeqOperatorDec		The decrement operator decrements the current position of the specified sequence object to current - used.
SeqOperatorInc		The increment operator increments the current position of the specified sequence object to current - used.

Command	Icon	Action Performed
SeqReadCurrentPosition		Reads an existing record in the database table containing the persistent current positions of sequences.
SeqReadFromFile		Initializes the sequence instance data from a configuration file.
SeqRemoveAll		Removes all labware items from the given sequence.
SeqRemoveAt		Removes the labware item at a given position.
SeqRemoveAllProperties		Removes at the specified sequence position all properties.
SeqRemoveProperty		Removes at the specified sequence position the specified property.
SeqResetSequenceIndexes		Resets all indexes (current position, end position and max positions) of the given sequence.
SeqSetMax		Sets the maximum number of positions/labware, which is allowed to be processed per step.
SeqSetProperty		Sets at the specified sequence position the specified property to the specified value.
SeqSetPropertyRange		Sets at the specified sequence positions the specified property to the specified value.
SeqSetUsedPositions		Sets the number of positions/labware in a sequence, which has been processed by the last single step.
SeqWriteCurrentPosition		Appends a new record or updates an existing record in the database table containing the persistent current positions of sequences.
SeqWriteToFile		Writes the sequence instance data to a configuration file.

11.12.17 String Library

The string library provides several functions to concatenate, fill, search and modify strings.

Command	Icon	Action Performed
StrAsciiToStr		Converts the given ASCII code (integer) to a character (string).
StrConcat12		Returns the concatenation of the arguments. If an argument is not of a string type, it is first converted to a string and then concated.
StrConcat2		Returns the concatenation of the arguments.
StrConcat4		Returns the concatenation of the arguments.
StrConcat8		Returns the concatenation of the arguments.
StrEvaluateExpr		Evaluates an expression. All variables referenced inside the expression must have global scope.
StrFillLeft		Fills leading characters to the string.
StrFillRight		Fills trailing characters to the string.
StrFind		Searches the string for the first match of a sub-string.
StrFStr		Converts the floating-point number into the corresponding character string.
StrFStrEx		Converts the floating-point number into the corresponding character string.
StrFVal		Converts the sequence of digits, contained in the character string str, into the corresponding integer.
StrGetLength		Returns the number of characters in a string object.
StrGetType		Retrieves the type of the value of a variable.
StrHexIStr		Converts the integer number into the corresponding hexadecimal character string.
StrIsDigit		Determines if the specified character (string) is a digit.
StrIStr		Converts the integer number into the corresponding character string.

Command	Icon	Action Performed
StrIVal		Converts the sequence of digits, contained in the character string str, into the corresponding integer. The sequence of digits is interpreted decimal. If the sequence begins with 0x, it is interpreted hexadecimal.
StrLeft		Extracts the first (that is, left-most) count characters from the string object and returns a copy of the extracted sub-string. If count exceeds the string length, then the entire string is extracted.
StrMakeLower		Converts the string object to a lowercase string.
StrMakeUpper		Converts the string object to an uppercase string.
StrMid		Extracts a sub-string of length count characters from the string object, starting at position first (zero-based). The function returns a copy of the extracted sub-string.
StrReplace		Replaces indicated sub-strings with another sub-string.
StrReverseFind		Searches a string object for the last match of a sub-string.
StrRight		Extracts the last (that is, right-most) count characters from the string object and returns a copy of the extracted sub-string. If count exceeds the string length, then the entire string is extracted.
StrSpanExcluding		Can be used to search the string for the first occurrence of any character in the specified set subStr. StrSpanExcluding extracts and returns all characters preceding the first occurrence of a character from subStr (in other words, the character from subStr and all characters following it in the string, are not returned). If no character from subStr is found in the string, then StrSpanExcluding returns the entire string.
StrStrToAscii		Converts the given character (string) to an ASCII code (integer).
StrTrimLeft		Trims leading whitespace characters from the string (removes newline, space, tab, and user-defined characters).
StrTrimRight		Trims trailing whitespace characters from the string (removes newline, space, tab, and user-defined characters).

11.12.18 Time Library

The time library allows getting time and date information from the computer.

Command	Icon	Action Performed
TimGetTime		Returns the current time (string).
TimGetDate		Returns the current date (string).
TimGetFormattedTime		Returns a string that contains the formatted time.
TimGetFormattedDate		Returns a string that contains the formatted date.

11.12.19 Synchronize Library

The synchronize library is used to handle critical sections in methods.

The “**SynLib**” Prefix has been removed on some commands, in order to provide a better overview.

Command	Icon	Action Performed
SynInitializeCriticalSection		This function initializes a critical section object.
SynEnterCriticalSection		Waits for ownership of the specified critical section object. The function returns when the calling thread is granted ownership.
SynLeaveCriticalSection		Releases ownership of the specified critical section object.
Connect		Opens a connection to another computer in a network. Every computer gets its own connection.
CloseConnection		Closes the connection with the specific SyncHandle.
SendData		Sends the string in the variable DataToSend to the connection with the specific SyncHandle.
GetReceivedData		Reads data out of the local buffer from the specific SyncHandle.
GetAllReceivedData		Reads all data out of the local buffer from the specific SyncHandle.
ResetBuffer		Resets and clears the incoming data buffer from the specific SyncHandle.

11.12.20 Tip Counting Library

The tip counting library provides all functions to store the available number of tips for following methods. This is useful if a method does not use up all tips on the deck. After a new start of a method, the instrument will continue using the next available tip.

The "**TipCount**" Prefix has been removed, in order to provide a better overview.

Command	Icon	Action Performed
Write2		Writes the indexes from the given sequence to the element counter database in the system directory.
Read2		Reads the indexes from the element counter database in the System directory into the given sequence.
Edit2		Reads the indexes from the element counter database into the given sequence and prompts the user to edit the sequence graphically.

11.12.21 Trace Library

The trace library allows using functions to write into the trace window.

Command	Icon	Action Performed
TrcFormatTrace		Function to trace 2 formatted strings.
TrcFormatTrace4		Function to trace 4 formatted strings.
TrcFormatTrace8		Function to trace 8 formatted strings.
TrcFormatTrace12		Function to trace 12 formatted strings.
TrcGetMethodFileName		Retrieves the path and name of the topmost HSL source file that includes the current HSL source file.
TrcInputBox		Displays the input request prompt in a dialog box and returns with the value of the specified type entered by the user.
TrcMessageBox		Displays a message in a message box and returns with a value, which identifies the button selected by the user.
TrcTrace		Trace function with two arguments.
TrcTrace4		Trace function with four arguments.
TrcTrace8		Trace function with eight arguments.

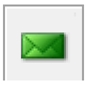
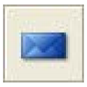
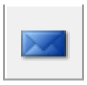
Command	Icon	Action Performed
TrcTrace12		Trace function with twelve arguments.
TrcTraceSequence		Traces the attributes of the given sequence.

11.12.22 Utility Library

The utility library is a collection of several useful commands to perform requests and more.

The “**Util**” Prefix has been removed, in order to provide a better overview.

Command	Icon	Action Performed
Abs		Returns a value of the same type that is passed to it specifying the absolute value of a number.
AddCheckSum		Computes the checksum of the specified file and writes the checksum value to the end of the file.
AsyncShell	c:\>	Runs an executable (exe, com, bat). This function runs other programs asynchronously.
GetBarcodeJoker		Returns the value for the barcode joker mapped to the given key (string).
GetEmailAddressOfSender		Get configured sender address for e-mails to be transmitted.
GetHWnd		Returns the application’s main window handle.
GetIVDSystem		Retrieves the IVD system installed flag from the system registry (integer).
GetLanguage		Retrieves the Phoenix language.
GetSMTPServerHostName		Request the configured SMTP server host name.
GetSimulationMode		Returns the simulation mode (0 = simulation off, 1 = full simulation)
GetTimeScaleFactor		Returns the current time scale factor for activities/task durations (float; defaults to 1.0).
GetTimerViewName		Returns the view name of the given timer.
GetUniqueRunId		Returns the unique ID of the current run.
GetUserName		Retrieves the name of the current user. This is the name of the user currently logged onto the system.

Command	Icon	Action Performed
IsFloat		Returns true if the variable var is a float.
IsInteger		Returns true if the variable var is an integer.
IsNegative		Returns true if number is negative.
IsNumber		Returns true if the variable var is a numeric expression.
IsString		Returns true if the variable var is a string.
Lock		Lock() and Unlock() are functions that enclose a series of HSL statements so that a group of HSL statements can be executed without interruption.
Lookup		Returns true if the array of variables contains the specified value, and if true, the parameter contains the index in the array of the found element.
Max		Returns the greater of two expressions (numeric or string).
Min		Returns the lower of two expressions (numeric or string).
Pause		Suspends the program execution at the next position where the Lock() and Unlock() functions match exactly.
RegisterAbortHandler		Registers abortHandler as a custom HSL function called before a method will be aborted. One or more abort handlers can be registered.
SendEMail		Send an e-mail through an SMTP server.
SetEMailAdressOfSender		Overwrite configured sender address for e-mails to be transmitted.
SetSMTPServerHostName		Overwrite configured SMTP server host name.
SetTimeScaleFactor		Sets the current time scale factor (float; defaults to 1.0). If the simulation mode is switched on, the time scale factor is used to scale task dependencies and activity durations
SetTimerViewName		Sets the view name of the given timer.
Sort		This sort algorithm acts on so called associative arrays (hash), which are built by two arrays. One array contains the keys, the other the corresponding values. This function sorts the keys.

Command	Icon	Action Performed
SyncShell		Runs an executable (exe, com, bat). This function runs other programs synchronously.
Unlock		Lock() and Unlock() are functions that enclose a series of HSL statements so that a group of HSL statements can be executed without interruption.
UnregisterAbortHandler		Unregisters abortHandler as a custom HSL function called before a method will be aborted
VerifyCheckSum		Verifies the checksum value of the specified file.

11.12.23 Utility Library 2

The HSL utility library 2 is an extension library to the core HSL utility library.

The “**Util2:**”, “**Util2: Debug**” and “**Util2: Error**” Prefixes have all been removed, in order to provide a better overview.

Command	Icon	Action Performed
CheckValueType		Checks a variable to be of a given type.
CheckValueRange		Checks a variable to be in a given (open) range.
CheckValueRangeMinMax		Checks a variable to be in a given (closed) range.
CheckValueTypeAndRange		Checks a variable to be of a given type and to be in a given (open) range.
CheckValueTypeAndRange MinMax		Checks a variable to be of a given type and to be in a given (closed) range.
VarArrCheckIndex		Generates a runtime error with detailed description if the index is invalid for the given variable array.
SeqArrCheckIndex		Generates a runtime error with detailed description if the index is invalid for the given sequence array.
VarArrGetAt		Returns a copy of the array element at the given index and raises a runtime error with a detailed description if the specified index is invalid for the given variable array.
SeqArrGetAt		Returns a copy of the array element at the given index and raises a runtime error with a detailed description if the specified index is invalid for the given sequence array.
ToString		Returns a string that represents the value of a given variable.

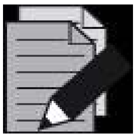
Command	Icon	Action Performed
RoundVolume		Rounds the given volume to one decimal place.
RoundVolumeUp		Rounds the given volume up to one decimal place.
RoundVolumeDown		Rounds the given volume down to one decimal place.
GetLabwarePosXYZ		Obtains the position of the given labware item from the Deck Layout using deck coordinates.
RaiseRuntimeError		Raises a runtime error.
RaiseRuntimeErrorInclPrevErrDesc		Raises a runtime error with an error description that includes the description of the previous error.
RaiseLast		Re-throws the current runtime error.
MakeHxResult		Returns an HxResult value given a major ID, a minor ID and an error code.
TraceSequence		Traces the name, the indexes and all labware and position IDs of the given sequence.
TraceSequenceAndData_1		Traces the given sequence and additional sequence data.
TraceSequenceAndData_2		Traces the given sequence and additional sequence data.
TraceSequencesAndData_1		Traces the given sequences and additional sequence data.
TraceSequencesAndData_2		Traces the given sequences and additional sequence data.
TraceArray		Traces the elements of the given array.
TraceArray_2		Traces the elements of the given arrays.
TraceArray_3		Traces the elements of the given arrays.
TraceArray_4		Traces the elements of the given arrays.
SetTraceArraySettings		Sets the current settings to trace multiple arrays.

11.12.24 HHS Library

The HHS library provides several functions to control the HHS, such as shaking and heating settings. The functions can be integrated into standard methods of the Microlab instrument.

To install the library, execute the file “**InstallHHSLibrary_Vx.x.exe**”. The file can be obtained from a local Hamilton Representative.

After confirming the installation of the addition, the heater shaker library will be installed automatically.

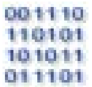


NOTE

The library requires the following Microsoft Package, which will be installed automatically during the setup procedure:

Microsoft Visual C++ 2005 Redistributable Package (x86).

Command	Icon	Action Performed
Create Star Device		Creates the device number which must be used as input parameter for each function of this library.
Create USB Device		Creates the device number which must be used as input parameter for each function of this library.
Terminate		The connection to the Microlab instrument and/or USB device is terminated. Note that this function does not stop the heating or shaking process of the Heater Shaker.
Start Shaker		This function starts the shaking process. If necessary, the Heater Shaker will be initialized. Before the shaking process is started, the plate is locked. Shaking has to be stopped by the “**Stop Shaker**” Command. Terminating the connection will not stop shaking. However, shaking is stopped upon method abort.
Start All Shaker		Start shaking on all initialized Shakers. Shakers that have not been initialized are not addressed. The plates are locked before the shaking process.
Start Shaker Timed		Start shaking for an indicated time. If necessary, the Heater Shaker will be initialized. Before the shaking process is started, the plate is locked. After shaking, the plate lock has to be opened with the “**SetPlateLock**” Function.
Start All Shaker Timed		Start shaking on all initialized Heater Shakers for an indicated time. The plates are locked before the shaking process. After shaking, the plate lock has to be opened with the “**SetPlateLock**” Function.

Command	Icon	Action Performed
Wait For Shaker		Wait for the Heater Shaker to finish. The plate is unlocked after shaking has been stopped. This command is only used in combination with “**Start Shaker Timed**” or “**Start All Shaker Timed**”.
Stop Shaker		Stop shaking and unlock plate.
Stop All Shaker		Stop shaking on all Heater Shakers. The plates will be unlocked subsequently.
Set shaker parameter		Set shaking parameters, such as shaking direction, shaking speed and acceleration.
Get Shaker Parameter		Get shaking parameters, such as shaking direction, shaking speed and acceleration.
Start Temp Control		Start temperature control on the Heater Shaker (must be greater than ambient temperature plus 5°C). Temperature control has to be stopped by the “**Stop Temp Control**” Function or will be constantly on. Terminating the connection will not stop heating. However, heating is stopped upon method abort.
Wait for Temp Control		Wait until the Heater Shaker has reached the set temperature. This function will wait until the defined temperature is reached and is stable for 180 seconds. Only then, the method will continue.
Stop Temp Control		Stop temperature control of the Heater Shaker.
Get Temperature		Receive the current temperature of the Heater Shaker.
Set Temperature Parameter		Set parameters for temperature control. In most cases, the default settings can be used, and this function is not needed.
Get Temperature Parameter		Receive the parameters for temperature control.
Get Temperature State		Get the status of the temperature control. The temperature should be within a defined temperature range.
Send Firmware Command		Send a firmware command to the Heater Shaker.

Command	Icon	Action Performed
Set Plate Lock		Open or close the plate lock. The plate is always locked automatically before shaking is started, but this command is useful to position and fix the plate in the center of the flat bottom adapter before pipetting, or when using the commands “**Start Shaker Timed**” or “**Start All Shaker Timed**” as these commands do not open the plate lock after shaking.
Set Simulation		Set run mode to simulation for all functions in this library. In simulation mode, no signals are sent to the HHS.
Set USB Trace		Turn on/off tracing of communication to and from USB port.
Begin Monitoring		Start to monitor the performance of the HHS. This function monitors the temperature and speed in the background. The tolerated range of the temperature can be set with the function “**SetTempParameter**”. The tolerated range of speed is defined in this function. The status of the temperature can be requested in a defined interval and is then written to the trace file.
End Monitoring		Get the shaking speed of a HHS.
Get Shaker Speed		Get the shaking speed of a HHS.
Get Serial Number		Get the serial number a HHS.
Get Firmware Version		Get the firmware version of a HHS.

11.12.25 HSL VacuumBrandPump / Microlab STAR BVS Library

The Microlab STAR BVS Library allows for the controlling of up to four BVS / CVS units simultaneously. There are two categories of functions provided:

High Level Functions

The high level functions control the whole vacuum process including tracking of the vacuum action to the Hamilton Database.

The use of HighLevelFunctions is recommended.

High level functions have the prefix **HSLStarBVSLib.**

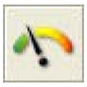
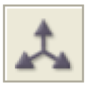
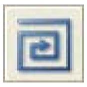
NOTE

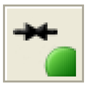
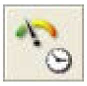
The BVS Library can also be used for CVS.

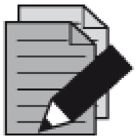
11.12.26 Low Level Functions

These functions reflect the command set of the "**BVS pump controller**" directly. These commands are old and are no longer recommended for use.

Low level functions have the prefix **HSLVacuuBrandPump**.

Command	Icon	Action Performed
BVSAbort		This function is used to stop all pump units and shut down their connections in an abort handler.
BVSGetAmbientPressure		Returns the ambient pressure measured with the specified pump unit.
BVSGetSimulationMode		Returns whether simulation mode is set for the specified BVS / CVS or not.
BVSInitialize		Initializes the connection to the specified BVS / CVS.
BVSSetSimulationMode		Sets the specified BVS / CVS to simulation mode.
BVSTerminate		Closes the connection to the specified BVS / CVS.
BVSTrack		Tracks a BVS / CVS volume move to the database.
BVSVacuum		Runs the vacuum process on the specified BVS / CVS.
BVSVacuumTrack		Runs the vacuum process on the specified BVS / CVS. The volume move is tracked to the Database.

Command	Icon	Action Performed
Initialize		By using this function, the communication port will be initialized. The pumping unit will be requested for errors to ensure the communication works.
OpenAirAdmittanceValve		Allows opening the air admittance valve.
ReqActualPressure		Requests the actual pressure in the system and the measured value will be returned.
StartPressureControl		Prepares the pumping control unit for a pressure-controlled execution of the pump and starts the pump.
StopPumpImmediately		A running pump started with the function StartPressureControl() can be stopped immediately at any time calling this function.
Terminate		Releases system resources occupied by the function Initialize().
WaitForPumpStopped		This function allows synchronization of pumping action and other actions which can be executed after the pump was started with a call of the function StartPressureControl().



NOTE

Do not mix high-level functions and low-level functions in the same method. Make sure to specify the library name.

11.12.27 Vector Database Tracking Library

The HSL database tracking library allows accessing and manipulating tracking data in the database.

The prefixes “**VectorDB_Deck:**”, “**VectorDB_Labware:**”, “**VectorDB_LabwareType:**”, “**VectorDB_Run:**”, “**VectorDB_Experiment:**” and “**VectorDB_TrackAction:**” have all been removed, in order to provide a better overview.

Command	Icon	Action Performed
AssignLabwareToJob		Assigns the given job to the given labware.
AssignLoadedLabware ToJobs		Assigns labware loaded on a given deck to corresponding 'Unprocessed' jobs.
GetDeckID		Retrieves the deck ID given an instrument name.
GetDeckID2		Retrieves the deck ID given an instrument.

Command	Icon	Action Performed
GetAllLabwareOnDeck		Retrieves all loaded labware given a deck ID.
IsLabwareLoaded		Returns the ID of the labware if the labware is loaded at the given position on the given deck.
IsLabwareLoadedByLabwareIdPositionId		Returns the ID of the labware if the labware is loaded at the given position on the given deck.
IsBarcodeLoaded		Returns the ID of the labware with the given barcode if the barcode is loaded on the given deck.
GetElementID		Retrieves the ElementID given a labware access name.
GetElementIDByDeckIDLabwareId		Retrieves the ElementID for the given labware and deck.
GetElementIDByDeckIDLabwareId		Retrieves the ElementID for the given labware and deck.
GetLabwareLoadingTime		Retrieves the labware loading time given a labware access name.
GetLabwareLoadingTimeByElementID		Retrieves the labware loading time on given deck for given labware.
GetLabware		Retrieves labware details information given a labware access name.
GetLabwareByElementID		Retrieves labware details information given an element ID.
GetLabwareBarcode		Retrieves the barcode given a labware access name.
GetLabwareBarcodeByElementID		Retrieves the barcode given an element ID.
GetLabwareVolume		Retrieves the volume given a labware access name.
GetLabwareVolumeByElementID		Retrieves the volume given an element ID.

11.12.28 Vector Database Worklist Management Library

The HSL vector database worklist management library allows accessing and manipulating worklist data in the vector database.

The “**VectorDB**” Prefix has been removed, in order to provide a better overview.

Command	Icon	Action Performed
AddJob		Adds a new job to the vector database.
AddJobAdditionalData		Adds additional data for the specified job to the vector database.
RemoveJob		Removes the specified job from the vector database.
RemoveJobAdditionalData		Removes additional data for the specified job from the vector database.
RemoveAllJobs		Removes all jobs (including additional data) that match the given state from the vector database.
GetJobs		Returns information about all jobs in the vector database that match the given job state.
GetJobIDs		Returns all job IDs that match the given filter in the given sort order.
GetJob		Returns information about the specified job from the vector database.
GetJobAdditionalData		Returns information about the additional data for the specified job from the vector database.
GetJobsSortedByPriority		Returns information about all jobs in the vector database that match the given criteria.
GetJobsForLabware		Returns information about all jobs in the vector database that match the given criteria.
GetJobsForLabware SortedByPriority		Returns information about all jobs in the vector database that match the given criteria.
GetJobsForBarcode		Returns information about all jobs in the vector database that match the given barcode criteria.
GetJobsForBarcode SortedByPriority		Returns information about all jobs in the vector database that match the given priority criteria.
GetJobsForElementID		Returns information about all jobs in the vector database that match the given element criteria.
GetJobsForElementID SortedByPriority		Returns information about all jobs in the vector database that match the given criteria.
GetJobState		Gets the state of the specified job from the vector database.

Command	Icon	Action Performed
SetJobState		Sets the state of the specified job in the vector database.
RemoveJobsForLabware		Removes information about all jobs that match the given criteria.
RemoveJobsForBarcode		Removes information about all jobs that match the given criteria.
RemoveJobsForElementID		Removes information about all jobs that match the given criteria.

11.12.29 Data Handling Steps

Data handling steps are standard procedures for data and file handling.

The data handling steps have to be activated by clicking on "**Method → Instrument and Smart Steps…**"

Command	Icon	Action Performed
GenerateMappingFile		Creates mapped report file(s) for labware within a method run.
SetLabwareBarcode		Set the barcode for a labware within the sample tracking database.
RemoveLabware		Removes labware from the access of sample tracking.
ImportWorklist		Imports worklist data from files into the vector database.
UpdateJobStatus		Set the job status of a sample contained in the worklist data to 'Processed'.

11.13 Customized Error Handling

For various error situations, a defined walk-away error handling which uses predefined default settings can be set. Power Steps, Smart Steps, Easy Steps and Single Steps offer this option.

Three different levels of error handling are determined:

Fully-manual:

This behavior is the standard error handling.

1. An error dialog will be prompted.
2. The user has to select or indicate a recovery action that will be executed.
3. After doing so, the recovery action dialog has to be closed by the user.

Semi-automated:

1. An error dialog will be prompted.
2. The user has to select or indicate a recovery action that will be executed.
3. The recovery action dialog will appear for a limited time. If the user does not specify an error recovery during the given timeframe, the dialog box closes automatically and will execute the default error handling.

Fully-automated (walk-away):

1. No error dialog opens on the screen.
2. The first defined recovery action will be executed immediately after an occurrence of an error.
3. If the first recovery action is unsuccessful, the second defined recovery action will be executed.

For every instrument-specific easy/single step of the method, an individual error recovery can be defined. The following configurations are possible:

- Appearance of the error recovery dialogs (which buttons are available)
- Default procedure
- Which error is flagged in the trace file
- A timeout (the timeframe of the dialog when it will automatically close down) after which the default recovery action will be executed

To Customize the Settings:

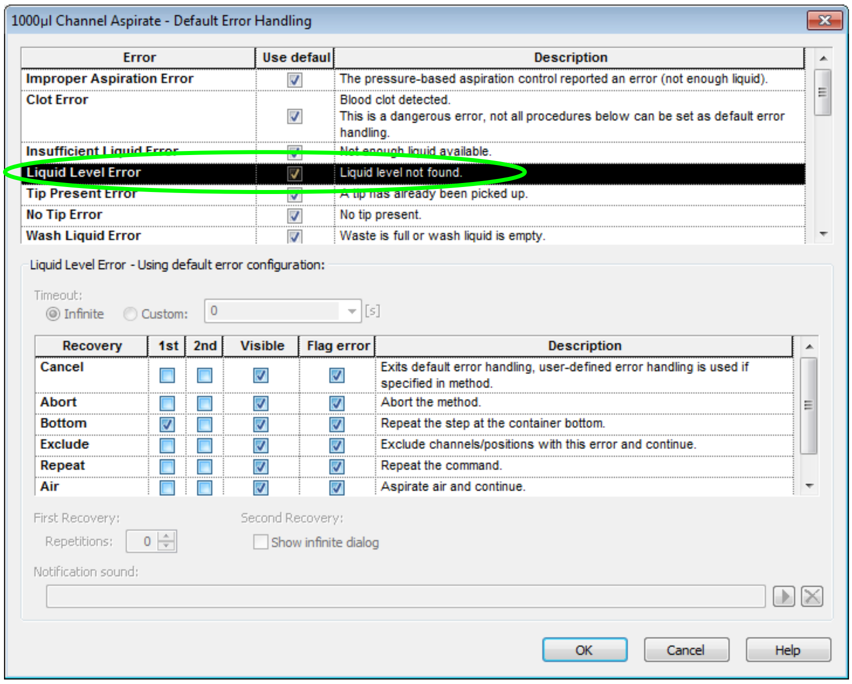
Disable the "**Use Default**" and the "**Timeout: Infinite**" Checkboxes. Only then the other settings become editable. A brief error description is given, followed by the available recovery options. Only one default procedure can be selected.

Among the choices are:

Cancel	Quits the current step and starts the user defined error handling if specified. If no user defined error handling is present, the method aborts.
Abort	Aborts the method
Bottom	Repeats the step at the bottom of the container
Exclude	Excludes all pipetting channels with this error and continues
Repeat	Repeats the command
Air	Aspirates air and continues

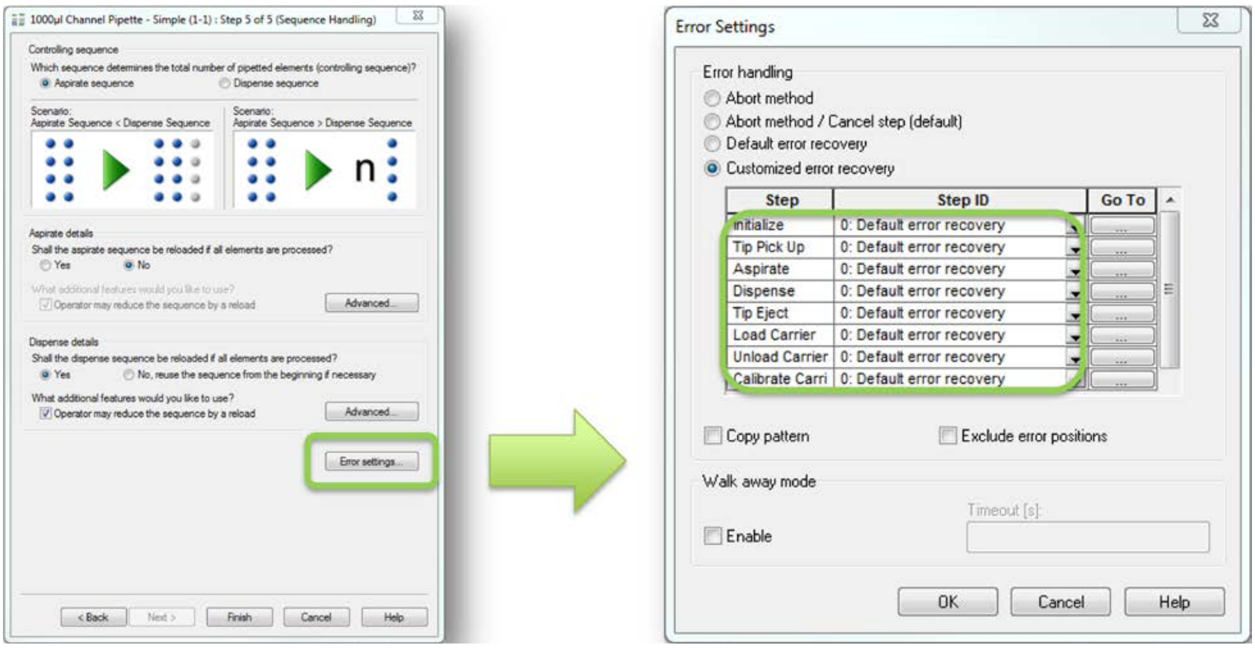
- Use the "**Visible**" Flag to add the appropriate button to the error recovery dialog box.
- The "**Set error flag**" will mark an erroneous container in the database.
- It is also possible to enable walk-away handling of errors (semi- and fully-automated mode):
- Disable the "**Timeout: Infinite**" Checkbox. Enter a timeout into the input field. The run time error dialog then pops-up, waits for the specified timeout and closes to continue with the default error recovery chosen for this error. If the timeout is set to 0, no dialog will open: the selected error recovery will be executed automatically.
- If the user clicks on the error during the timeout, the walk-away will be stopped, and the user has to select a recovery and continue manually.
- For a list of all errors and their recovery options, refer to the Online Help and the error settings dialog.

Because of this, every instrument-specific Smart Step, Easy Step or Single Step has an **[Error Settings…]** Button. The following is an example of the “**1000µl Channel Aspirate**” (Single Step):



- Like all the other types of errors found in the list, “**Liquid Level Error**” is activated with the default settings.

Error Settings:

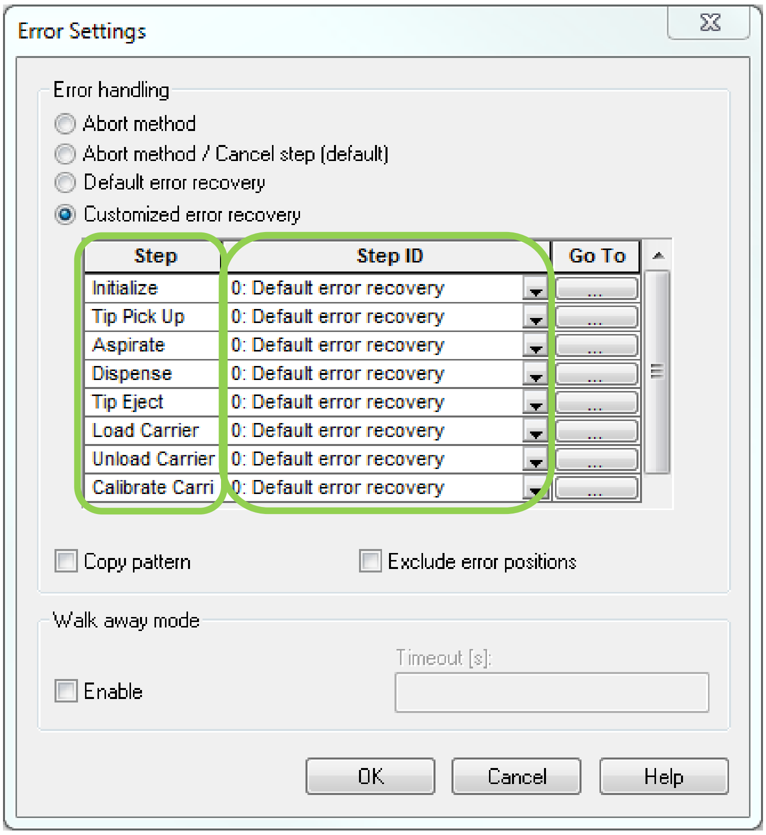


The Column *Step* shows all the individual parts of a Smart Step (what is executed inside a Smart Step)

The Column *Step ID* allows the the programmer to select:

- *2 - Abort Method*
- *1 - Abort / Cancel Step**
- *0 - Default error recovery*

or to **pass an ID** for a customized behavior in the case of an error.

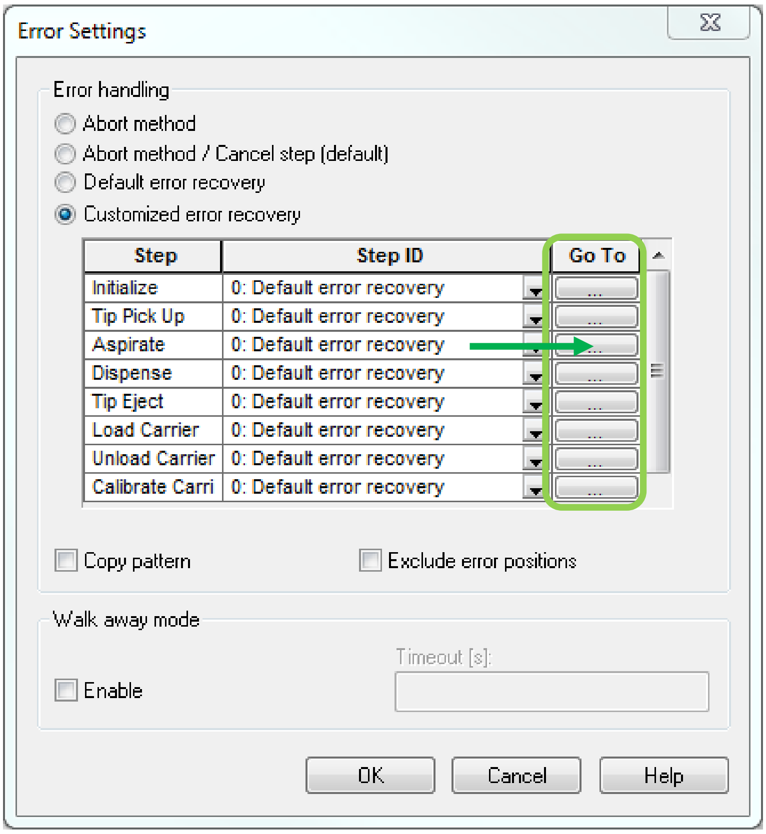


*if an *'Error handling by the user'* will be used:

Use the ***Go To*** column to customize the error handling:

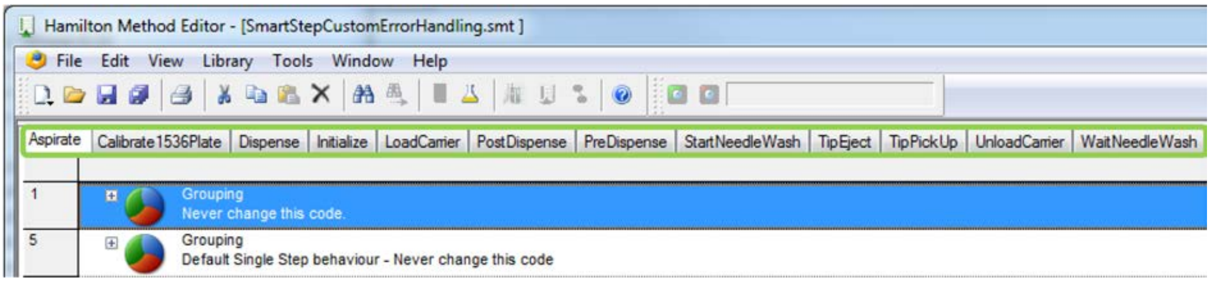
To keep it simple for the programmer, the ***Go To*** leads towards the sub-method library, containing the customizeable error handling.

Select the part of the step where you want to customize the error handling, and click on the ***Go To*** Tab.



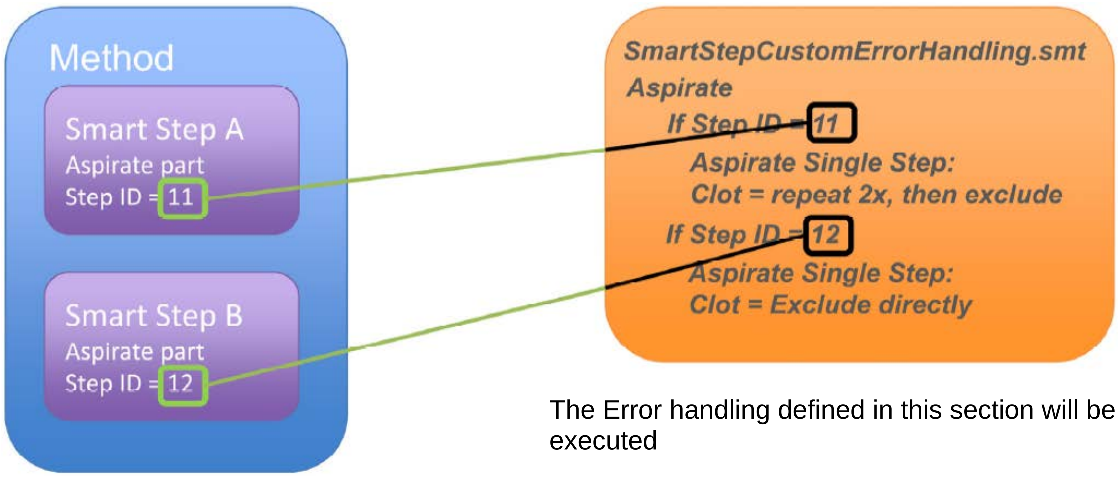
Clicking on ***Go To*** and a Sub-method Library opens. This Sub-method Library is the interface for the programmer to define customized error handling in Smart Steps.

For every section of the Smart Step, a group with the equivalent single step is shown. Each of these groups has an ID (Code).



This Single Step can now be modified to control the behavior in case of an error, in the same way as on a directly-programmed Single Step.

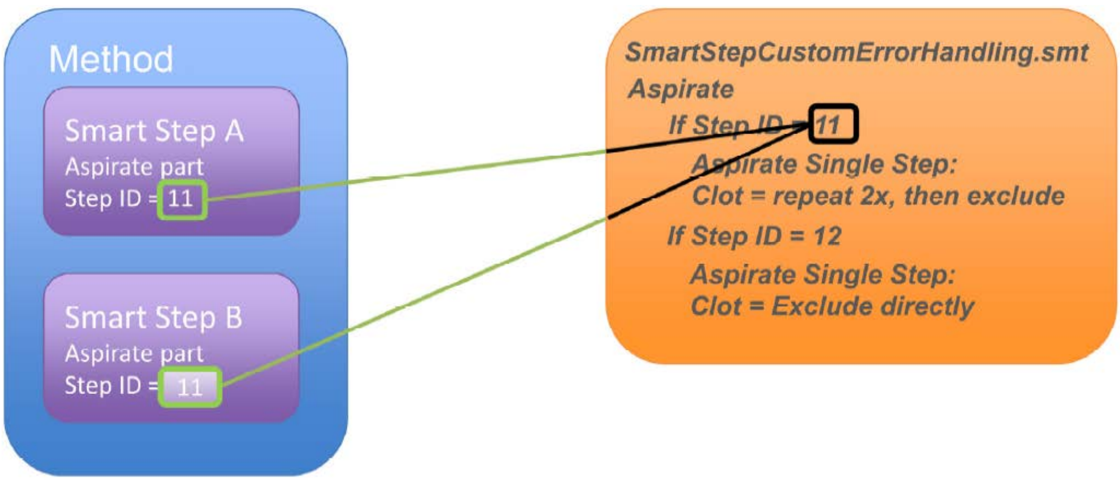
Schematic Explanation



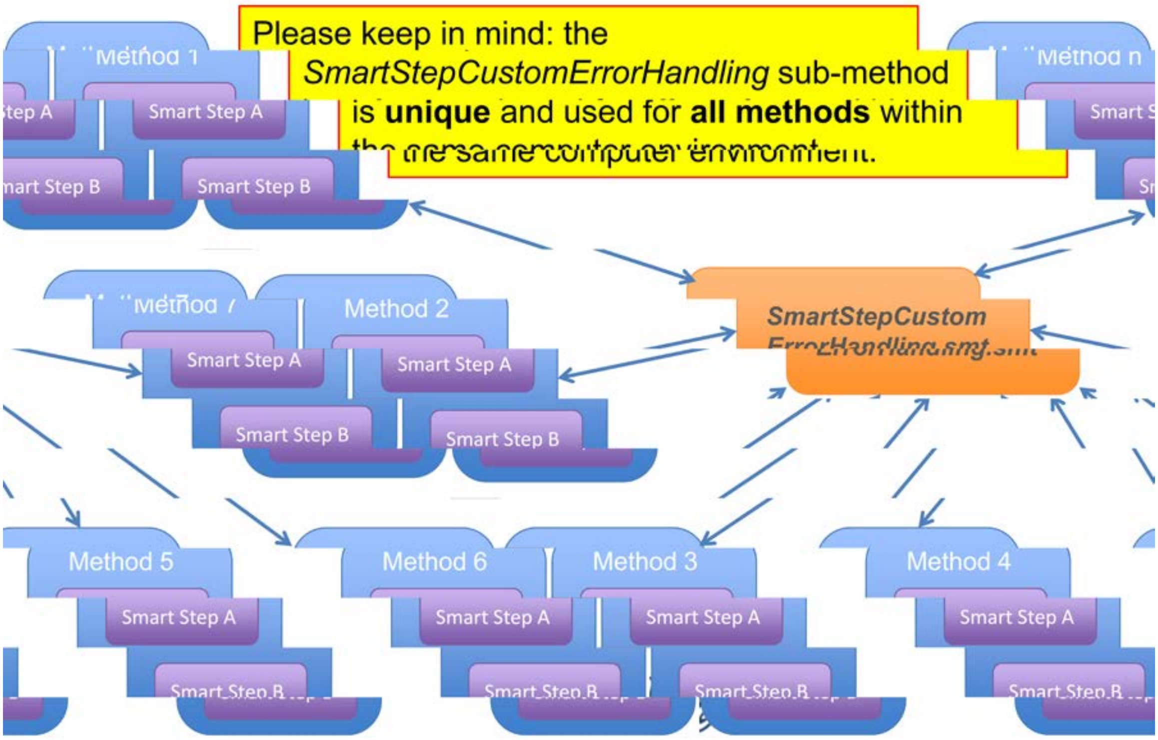
The Error handling defined in this section will be executed

Smart Step A has Step ID 11 defined for an Aspirate Error

Same is true for Smart Step B with Step ID 12.



Of course, both Smart Steps could use the same error Step ID.



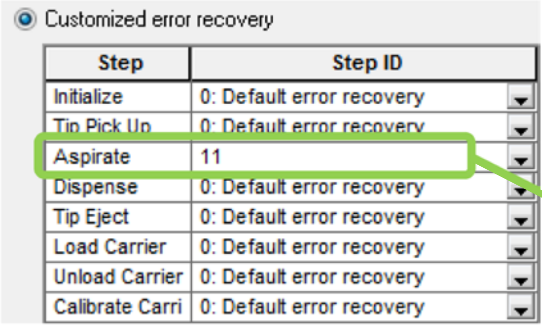
Import/Export behavior

To make sure the *SmartStepCustomErrorHandling* Submethod is exported, the flag ‘Export original Hamilton files’ must be checked:

During import, check the flag ‘Import original Hamilton files’.

ATTENTION

Please be aware that the imported Sub-method Library will overwrite the existing one! If you have added customized code blocks in your existing library, please copy this code before importing and add it after the import.



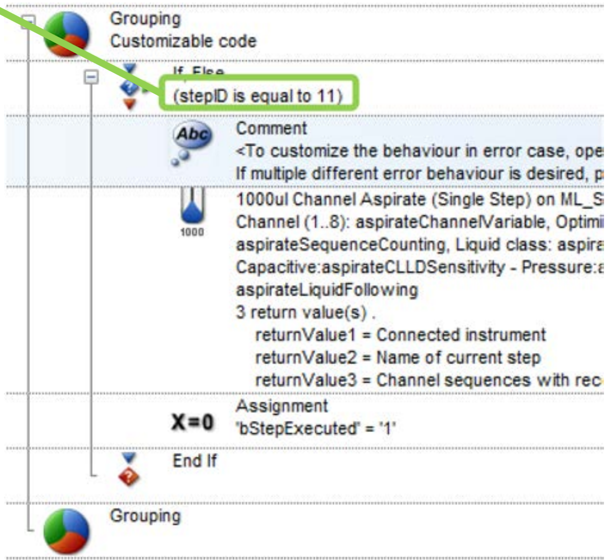
*Any other positive integer number refers to a group in the *SmartStepCustomErr orHandling.smt*

-2: ABORT

-1: Abort/Cancel (jump to *Error handling by the user*)

0: Default (use the default given in the acc. step)

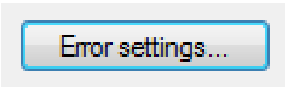
1-n: customized error handling code block* (example here: 11)



As seen, the Single Step in the Submethod Library acts as a ‘container’ for the error behavior. To customize it, just make the desired changes in the error handling of this step.

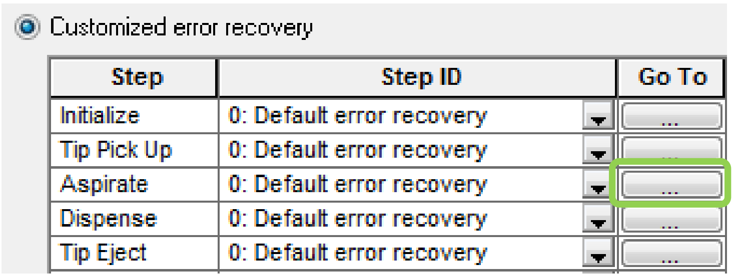
To do so, the following steps have to be performed:

- In the Smart Step, click on ***Error settings***

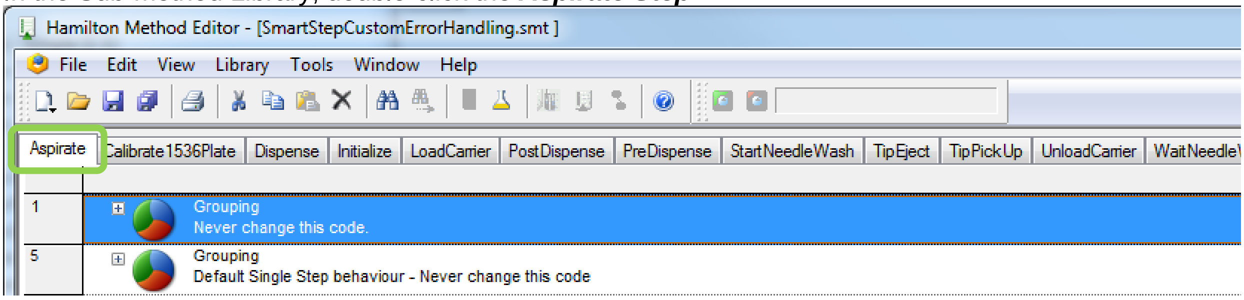


- Activate the ***Customized error recovery*** radio button

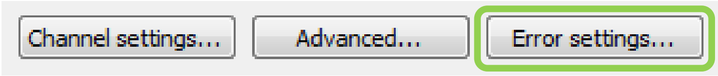
- In the line ***Aspirate***, click on ***Go To***



- In the Sub-method Library, double-click the ***Aspirate Step***

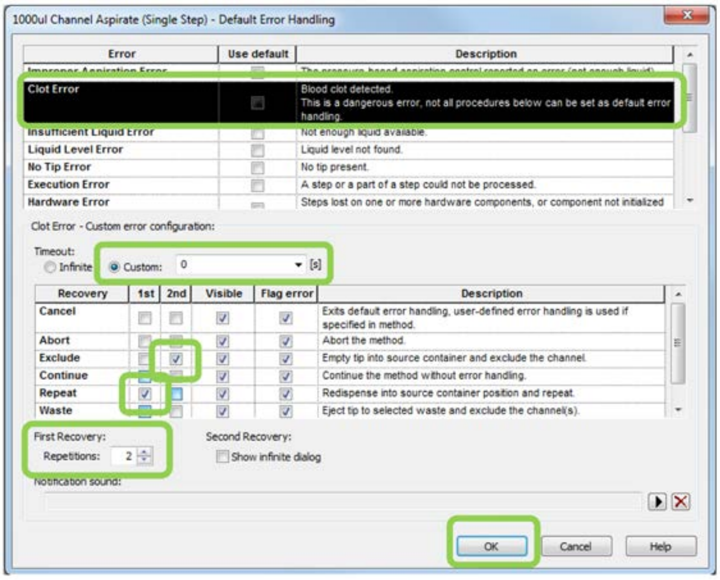


- In the ***Aspirate Step***, click the ***Error Settings Tab***

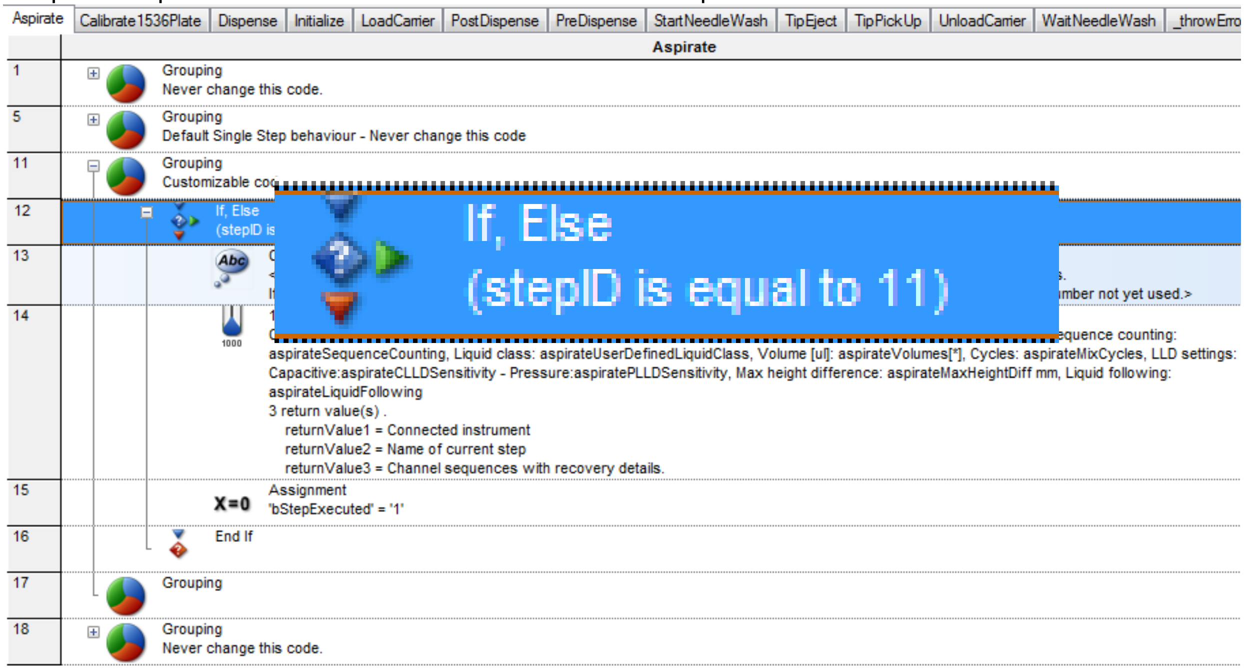


The Error Handling Panel opens:

- Select the ***Clot Error*** line
- In Timeout, switch to ***Custom*** (0)
- Set the ***1st*** recovery to ***Repeat***
- Set the ***2nd*** recovery to ***Exclude***
- Define the ***Repetitions*** for the First Recovery
- Close this dialog with ***OK***
- Close the Single Step with ***OK***

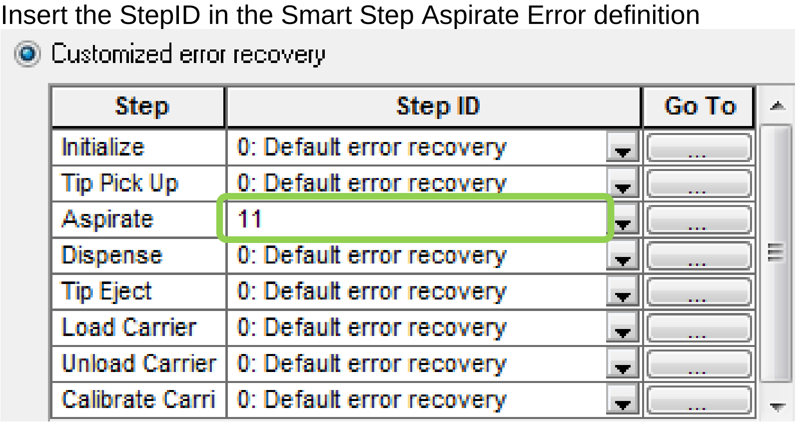


- Keep the StepID in mind – this is the 'link' for the Smart Step



- Close the Sub-method Library

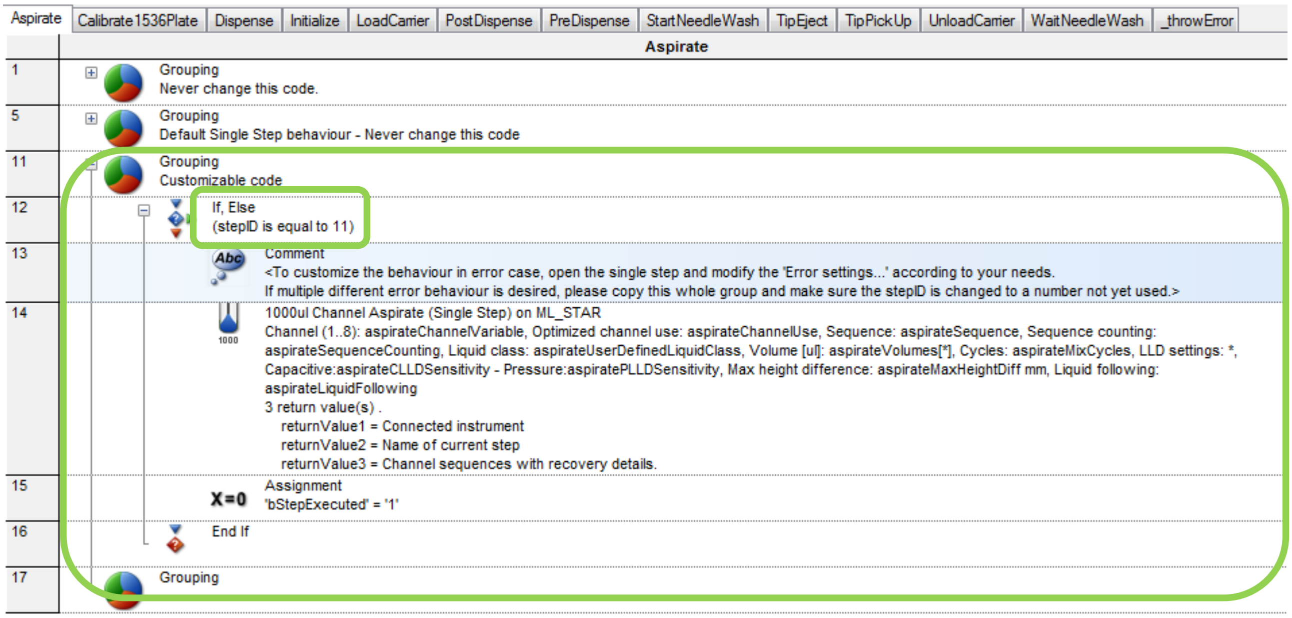
- Insert the StepID in the Smart Step Aspirate Error definition



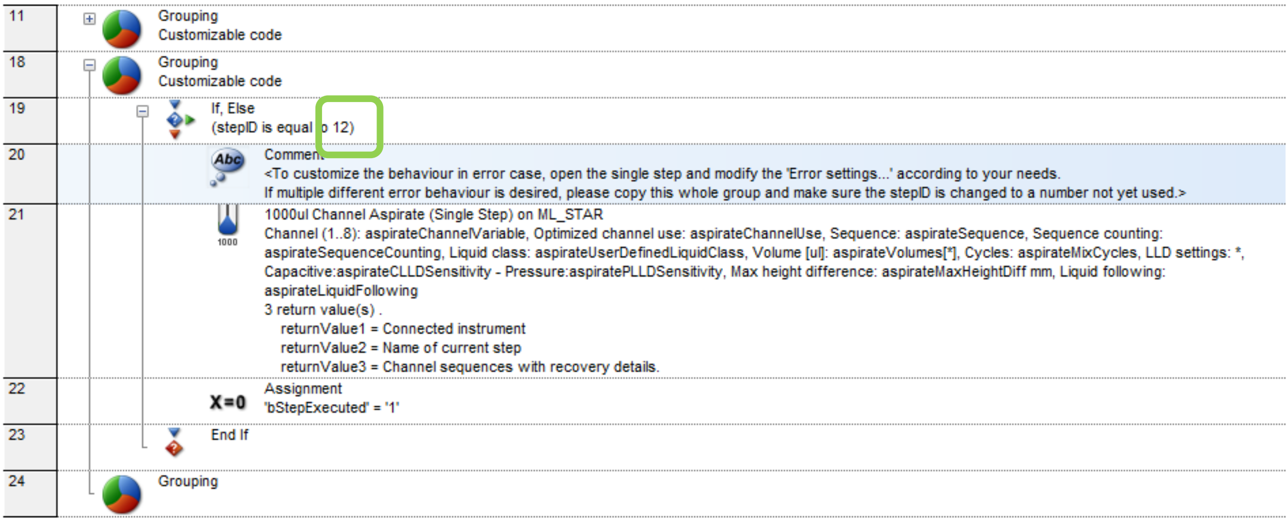
- Confirm with ***OK***, then close the Smart Step with ***Finish***

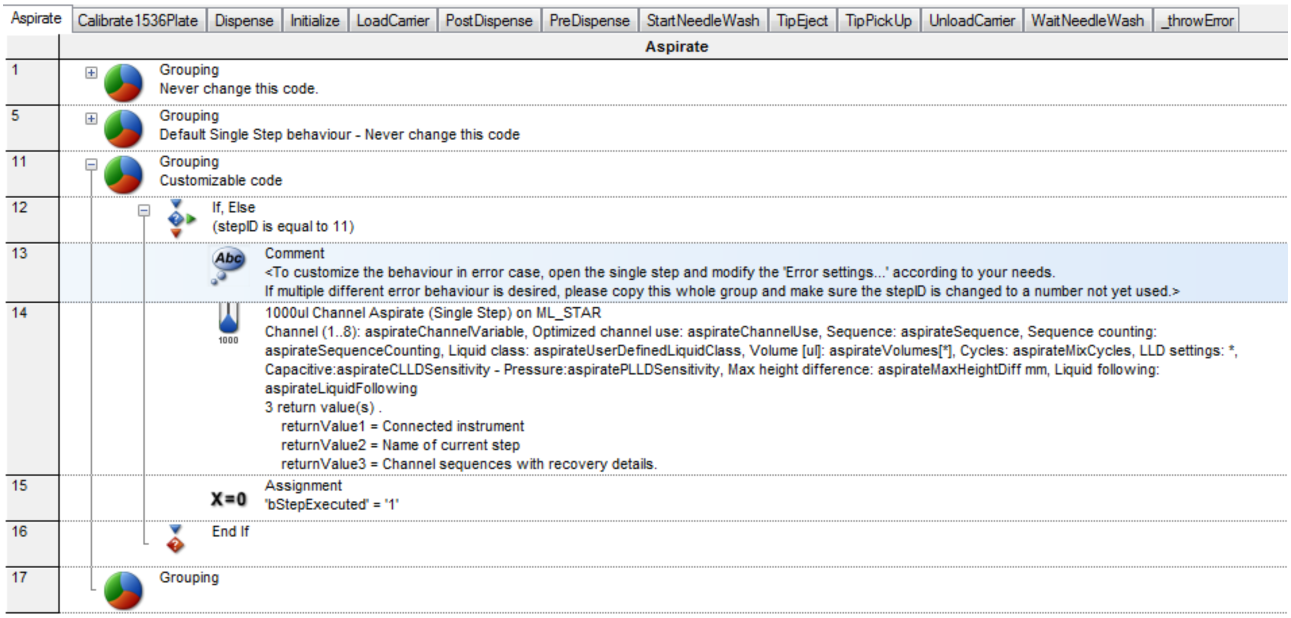
Adding more Error Blocks with unique Step IDs

If additional error options are needed, simply copy a ***Customizable code*** group and change the ***stepID*** to a not-yet-used integer number.



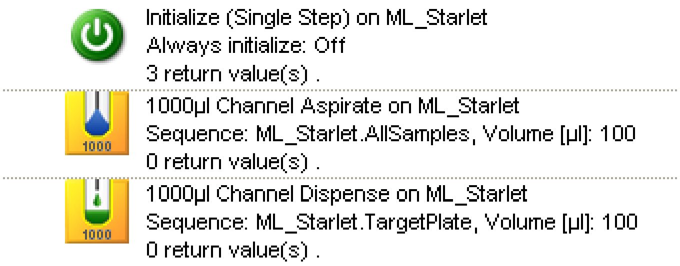
If additional error options are needed, simply copy a ***Customizable code*** group and change the ***stepID*** to a not-yet-used integer number.

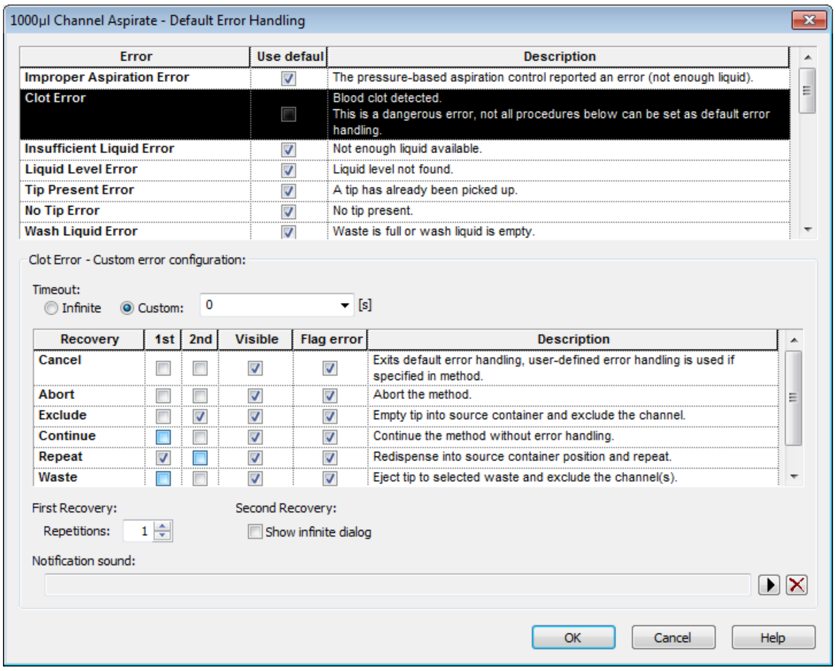




11.13.1 Example 1: Error Settings with Easy / Single Steps

This example shows how to configure an Easy Step Aspirate to repeat the aspiration, in the case of a clot error. If the second aspiration fails as well, then the erroneous pipetting channel will be excluded.





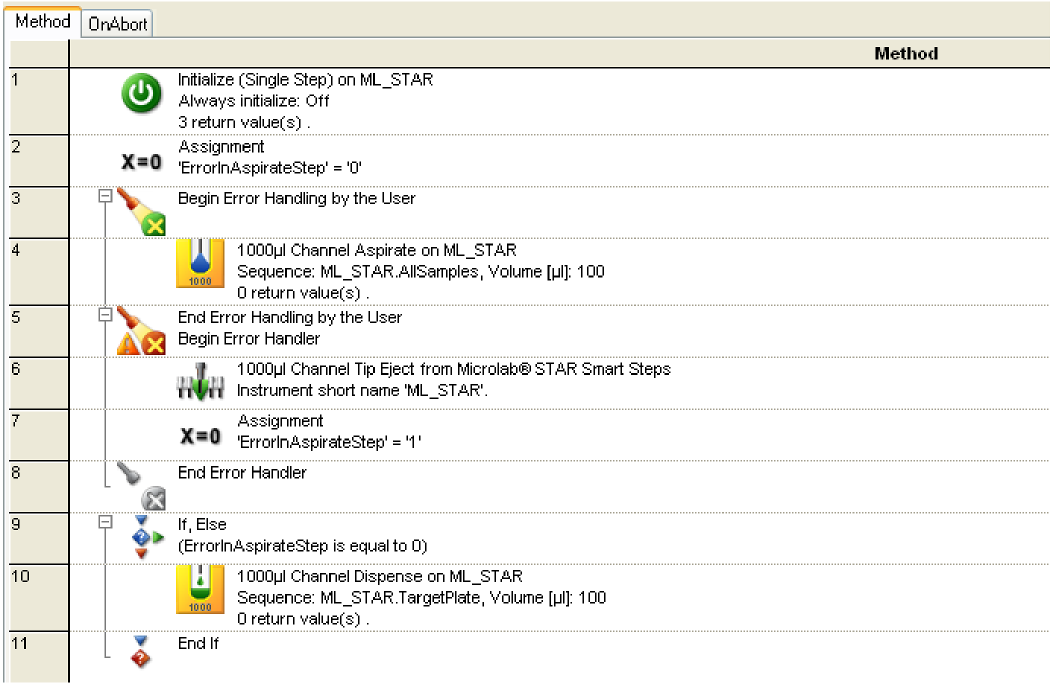
As seen in the image above, the checkboxes “**Use Default**” and the “**Timeout: Infinite**” are not activated.

Subsequently, the first recoveries are activated. In this case, a REPEAT error is selected as the first error handling. It first tries one time to aspirate the sample probe. In the case of a clot, the aspirated liquid is dispensed with half speed. Afterwards a second attempt at aspiration occurs. After a failed repeated attempt, this pipetting channel is excluded (second recovery = EXCLUDE).

Based on this principle, the desired error handling can be set for every listed error.

11.13.2 Example 2: Error Handling by the User

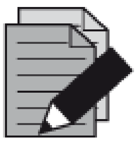
If none of the pre-defined possibilities in the error setting matches the user’s needs, an individual error handling can be programmed through the step “**Error Handling by the User**”.



The steps to be observed are programmed in between “**Begin Error Handling by the User**” and “**End Error Handling by the User**”.

If any error occurs in such a step, the method proceeds to the steps between “**Begin Error Handler**” and “**End Error Handler**”. This is the user defined error handling.

As a result, every desired error handling can be programmed.



NOTE

*The error settings of the specific error in the corresponding step have to be set to CANCEL, in order to be able to use the “**Error Handling by the User**”.*

To make use of the "**Error Handling by the User**", the settings listed below must be considered. The example on the previous page shows a "**1000µl Channel Aspirate**" Step where the clot handling is controlled through the "**Error Handling by the User**":

1. Uncheck the "**Use default**" Checkbox
2. Uncheck the "**Infinite**" Checkbox
3. Set the **[Cancel]** Radio Button

12 Demo Methods for the Microlab Instrument

12.1 Overview

Most of the instrument's day-to-day operations will be driven by methods, so it is important that these are correctly defined and are at the same time easy to operate.

The demo methods described in this section are applicable to Microlab STAR, STAR^{V}, and VANTAGE instruments.

Storage of Methods

Programmed methods are stored in the methods sub-directory on the hard disk. They can be opened, edited and saved, either in the graphical Method Editor or in the text-like HSL Method Editor.

Purpose of this Section

This section is a step-by-step approach on how to program a number of simple methods which are normally used in laboratories. Following these steps will help in becoming familiar with the layout and how the software works. Modifying the suggested methods to suit particular requirements or programming new methods based on what is suggested can then be done.

NOTE

Due to the current installation, a reduced selection of commands might be available. To make all commands or the commands of interest available, enable them in the Configuration Editor. Refer to *Section 3.4 Step Selection* *and / or* *Section 3.5 Microlab Instrument Settings*.

NOTE

Sequence definitions have a significant influence on method programming.

ATTENTION

Ensure that all methods are tested. It is recommended to run a simulation first, followed by a water run and (if successful), a run with real liquids.

The programmer is responsible for the validation of the method.

Note that the methods developed for one particular instrument and configuration do not automatically fit to another instrument or configuration. Differences in hardware and labware make revisions of programmed methods necessary.

Make sure to validate the method to be used for the instrument and configuration, particularly when importing methods from one computer or instrument to another.

The Demo Methods

The examples in the following pages will describe how to create your own layouts, methods, and sequences.

The demonstration examples can help in understanding the details which are not specifically described in the document.

12.2 Method for Sample Preparation using the Action Editor

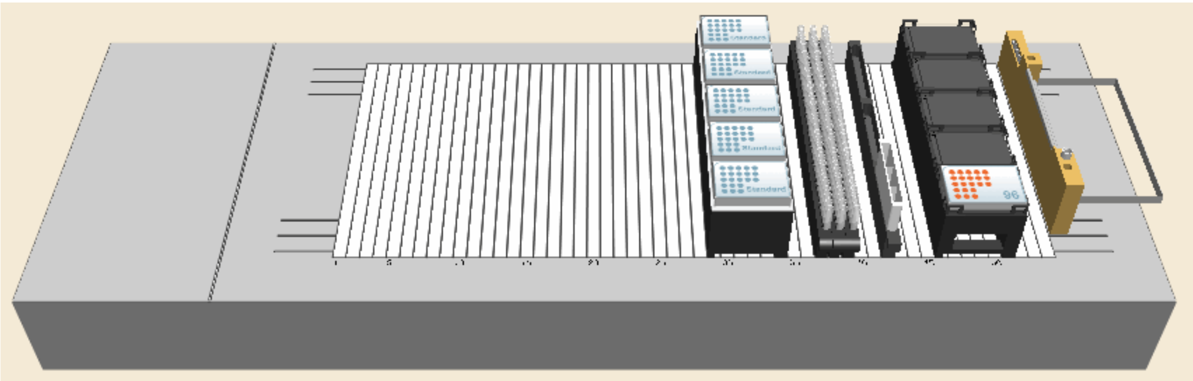
The action editor is a helpful tool for graphical method creation. It should be activated if the scheduling software is NOT installed. If scheduling software is installed, the same icon will start the activity editor.

This demo will:

- Aspirate buffer from a reagent trough
- Aliquot this over a Microplate, starting at A1
- Aspirate samples from three tube racks
- Dispense them into the same target plate and mix

The transferred volume will be 50µl of buffer and 100µl of the sample. The method is named "**MixSampleAndBuffer**". The method uses new CO-RE tips for each well. Start with an empty target plate.

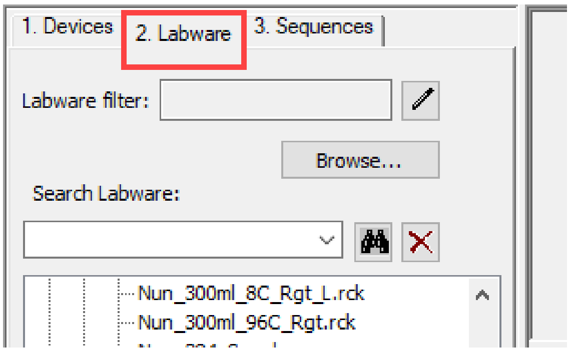
First, create a new method. This will also add a Deck Layout with the same name ("**MixSampleAndBuffer**"**.lay**). The Deck Layout for this method is shown in the picture below.



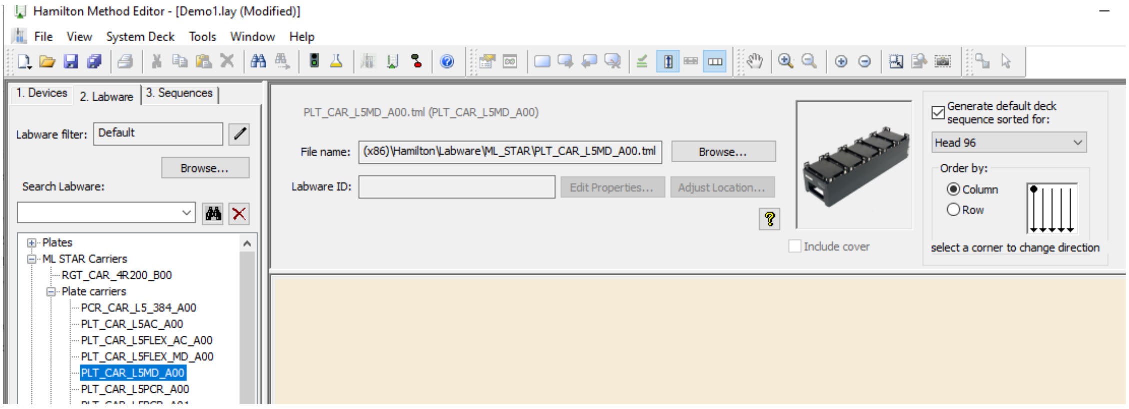
Creating the Deck Layout:

1. Start the HAMILTON VENUS application from the desktop shortcut, navigate to System Tools, and click on the "**Hamilton Method Editor**" Shortcut.
2. Select "**File → New → Method**" to create a new method. A window will open to be able to save the new method.
3. Enter the filename (MixSampleAndBuffer) and click **[Save]**. A new method window and system deck window are opened in the Method Editor. To activate, click on the system deck window.
4. Click the "**ML_STAR**" Tab in the lower-left corner.
5. Make sure the "**1000µl Channels**" Stamp Tool is selected to define the sorting of the default sequences when adding labware.

6. Click the on the “**Labware**” Tab.

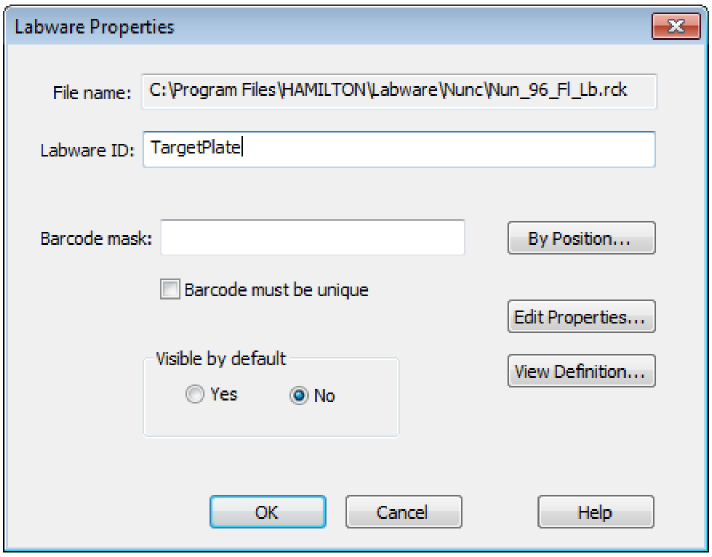


7. In the tree list right above the tabs, expand “**Microlab STAR Carriers → Plate Carriers**” and select “**PLT_CAR_L5MD_A00**”. This is a carrier for microplates.

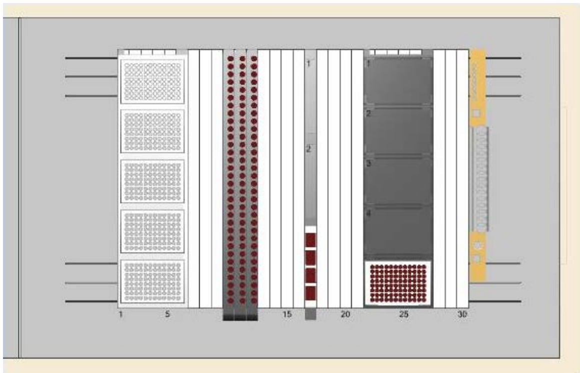


→ An image of the selected carrier is shown in the right-hand box.

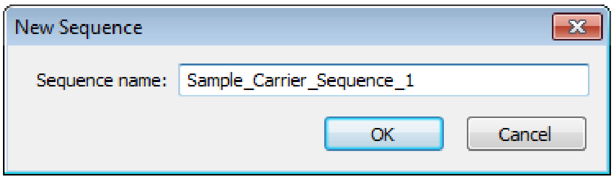
8. “Drag-and-Drop” the Carrier from the box onto the deck.
9. Repeat the steps above to add a Tip Carrier with 300µl Tips (without filter). Click “**Microlab STAR Carriers → Tip Carriers 96**” and “Drag-and-Drop” the TIP_CAR480_BC_ST_A00 onto the deck.
10. Following the previous steps, add three Sample Carriers to the deck. Click “**Microlab STAR Carriers → Sample Carriers → 32 positions → SMP_CAR_32_12x95_A00**” and drag it onto the deck. Repeat this step until there are 3 sample Carriers on the deck.
11. Same as the steps above, add a Reagent Carrier preloaded with 3 reagent troughs to the deck. Click “**Microlab STAR Carriers → Reagent Carriers → RGT_CAR_3R_A01**” and drag it onto the deck.
12. Place the target Plate on the Plate Carrier. Click “**Plates → 96 position plates → Nunc_96_Fl_Lb (low border)**” and drag the plate to the carrier position 5.
13. Right-click on the plate and select “**Properties**” in the Context Menu.
14. Change the LabwareID of the plate to “**TargetPlate**”.



15. Delete two of the three Reagent Troughs (since they are not needed). Select a Trough and click the Delete key on the keyboard or right-click the Trough and select "**Delete**" from the Context Menu.
16. Click the "**Sequences**" Tab to start editing the sequences. The upper part of the window changes to sequence editing.
17. Click the **[Clear selected]** Button to make sure that nothing is selected.
18. Click and hold the left mouse button at one corner of the Sample Carriers and move over all three Sample Carriers.



19. The positions appear in dark brown.

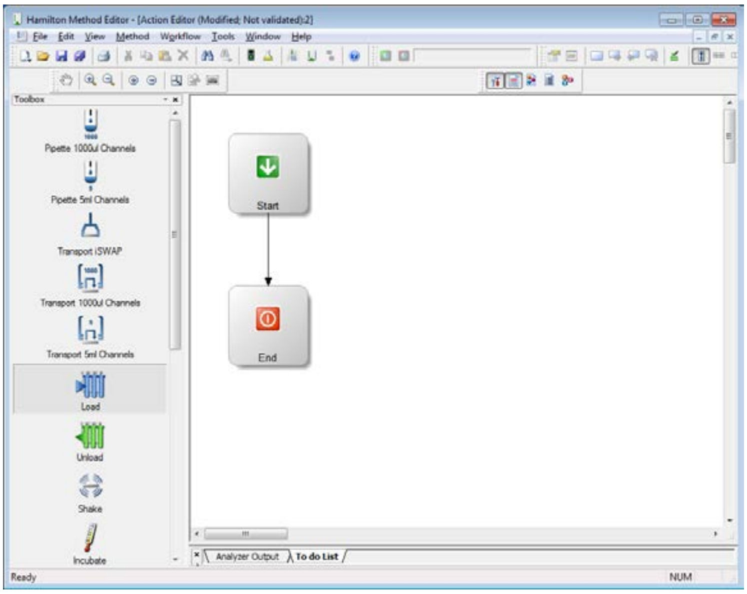


20. Click the **[Save as]** Button and enter the sequences name as shown in the image.
21. Save by clicking **[OK]**.

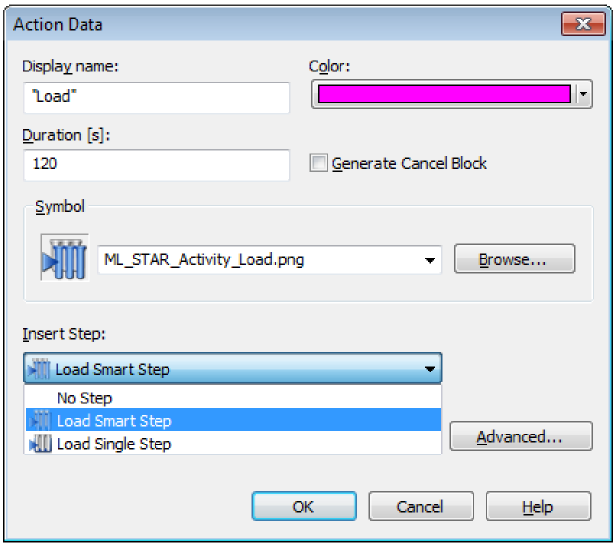
Creating the Method

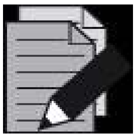
1. Open the Action Editor by clicking on the icon in the toolbar:

2. The Action Editor window should appear as presented below.



3. Click the “**Load**” action and drag it in the right section of the window. The Action Dialog appears as follows:
4. Leave the default values for the “**Display name**”, “**Color**”, “**Duration**” and “**Symbol**” selection. In the “**Insert Step**” Field, the loading step can be included.
5. This dialog allows for the creation of a method which will run on an instrument. Actions without steps can only be used for calculations.
6. Select the “**Load**” Smart Step from the “**Insert Step**” List and Click **[OK]**.





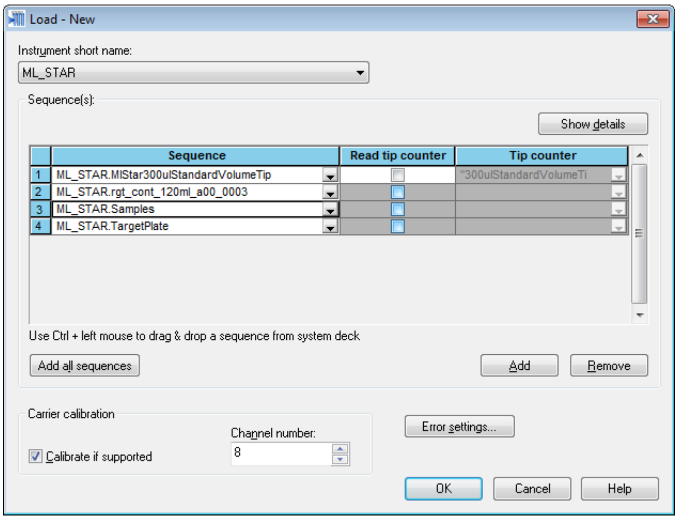
NOTE

Inserting steps to an action offers the option to create running methods for instruments.

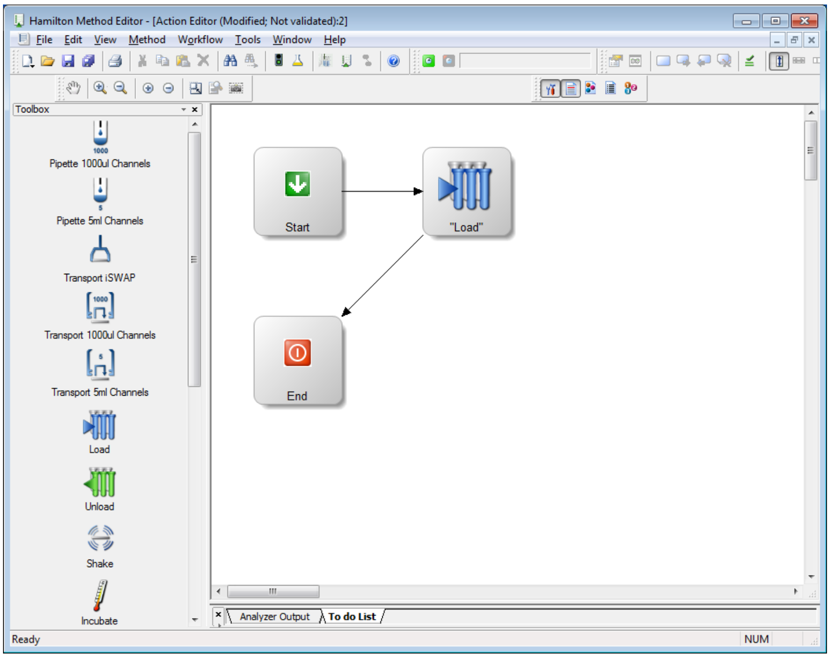
*Inserting "**No Step**" to an action is used to create a 'dummy' method for throughput calculations.*

Steps can also be inserted later.

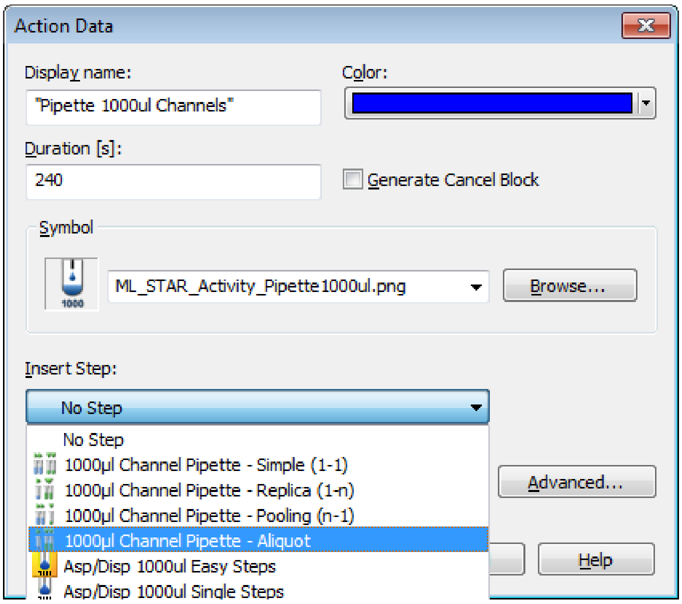
7. The action dialog will be closed, and the "**Load**" Smart Step Dialog will open as follows:



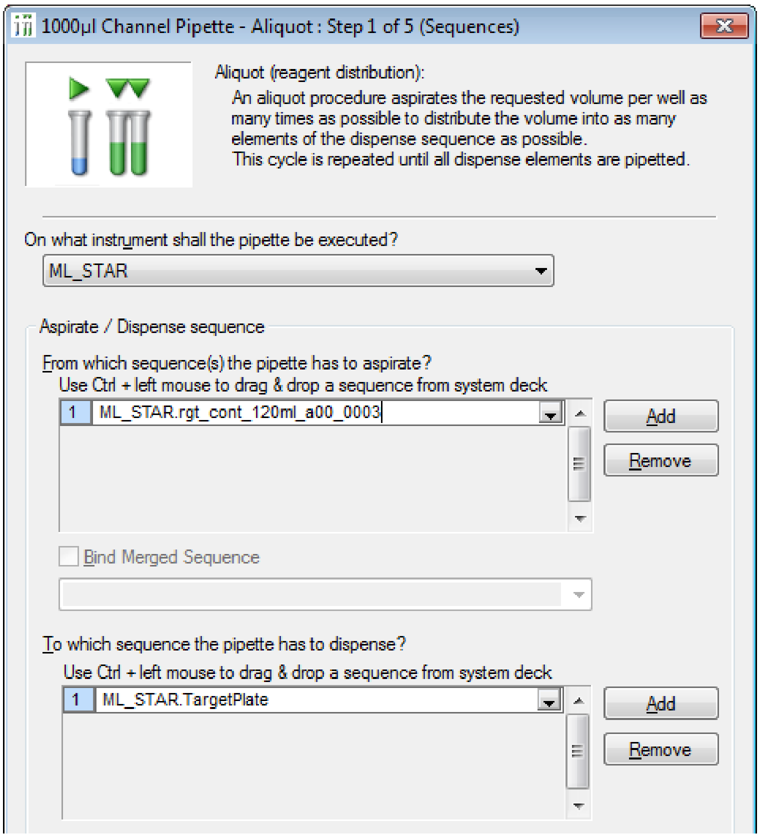
8. Add the sequences to load by clicking the **[Add]** Button and selecting the required sequences.
9. Click **[OK]** to close the "**Load**" Dialog
10. The "**Load**" Icon is now visible in the window as shown below.



11. Drag the action “**Pipette 1000µl Channels**” to the right window. The Action Data Dialog will open.
12. Leave the default values for the “**Display name**”, “**Color**”, “**Duration**” and “**Symbol**” selection. To execute the pipetting, select the “**1000µl Channel Pipette – Aliquot**” for the liquid transfer from the buffer trough onto the Target Plate.

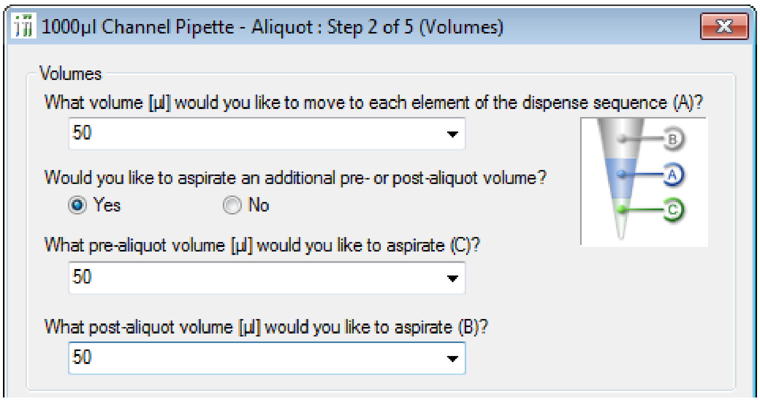


13. After clicking **[OK]**, the “**Smart Step Pipette**” Dialog opens.

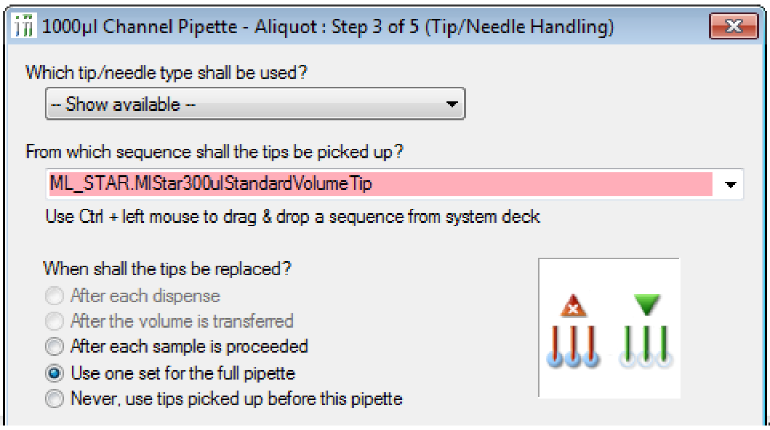


14. Fill out the sequences as shown above (by using the sequence list) and click **[Next >]** to continue.

15. In the next step, specify the pipetting, pre- and post-aliquot volumes and click **[Next >]**.

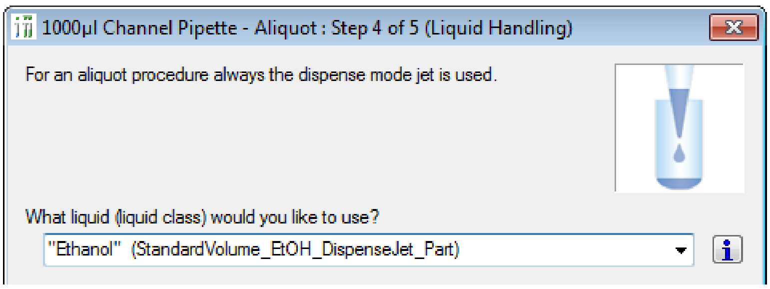


16. Here, a tip is pre-selected. Since only one kind of tip is on the deck, this tip type will be selected automatically. Make sure the "**Use one set for the full pipette**" Radio Button is activated.



17. Click **[Next >]** to continue.

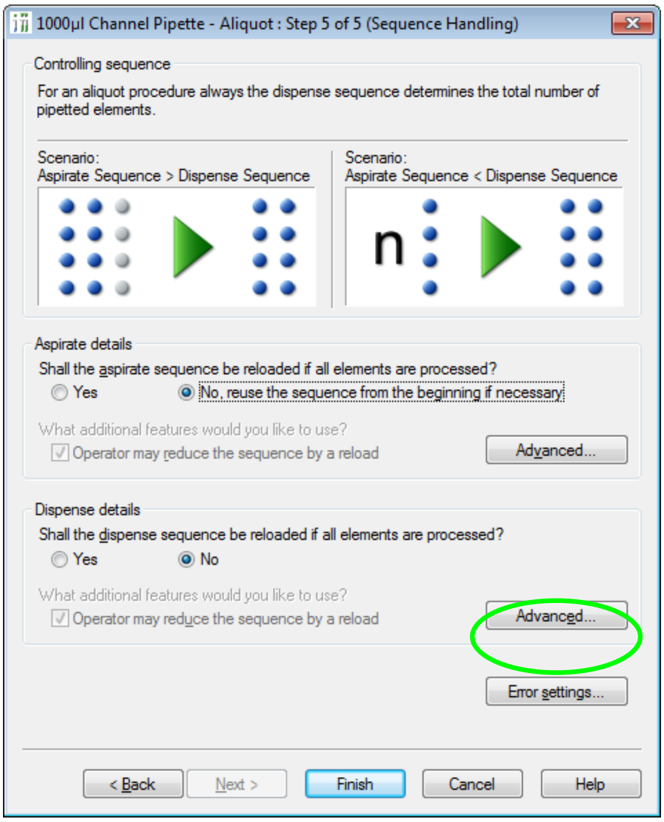
18. Select the liquid class that will be used for pipetting.



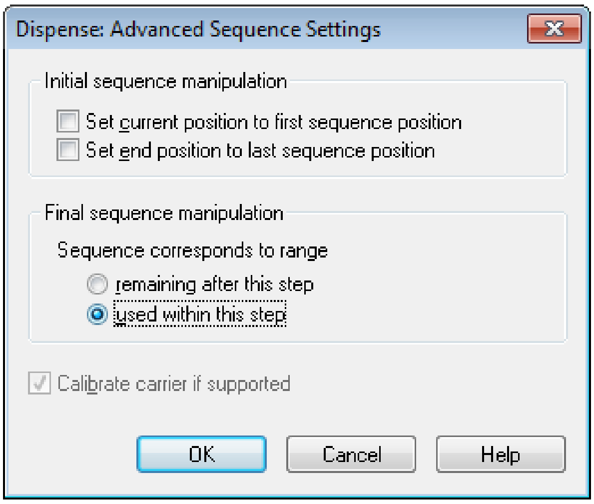
19. Click **[Next >]** to continue.

    In this step of the "**1000µl Channel Pipette – Aliquot**" Dialog, make sure that the next pipette step will find available sequence positions on the target plate. Otherwise, the entire TargetPlate sequence is used up by the aliquot and no more positions are available for the samples.

20. To do so, select the **[Advanced]** Button in the "**Dispense details**" Section as highlighted below.

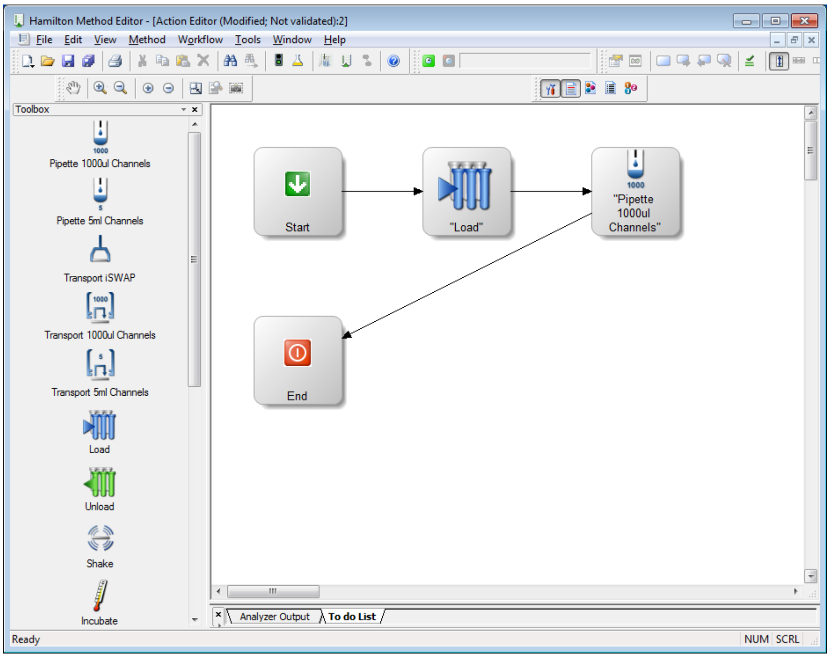


21. On the next screen, activate the **[Used within this step]** Radio Button. This will set the current position of the Target Plate back to 1 (which is Well A1), and pipetted the samples into the buffer.



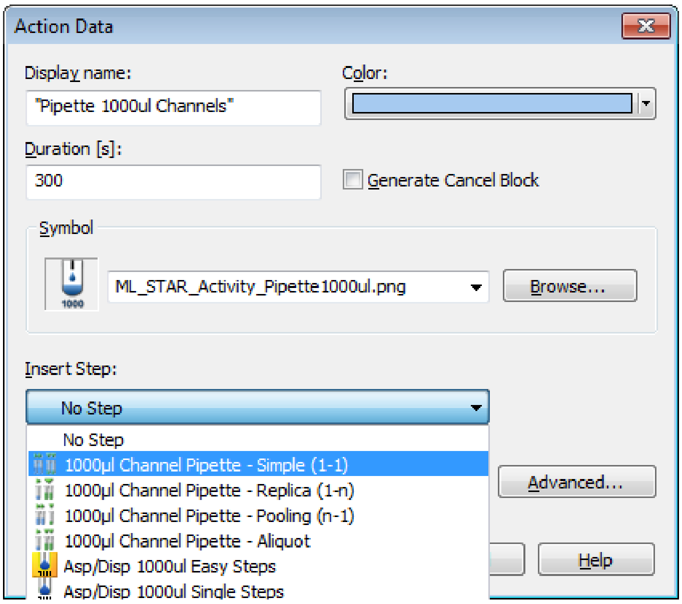
22. Click **[OK]** to close the "**Dispense: Advanced Sequence Settings**" Dialog.
23. Click **[Finish]** to end the "**1000µl Channel Pipette**" Wizard.

The pipetting from Buffer-to-Target Plate step is now visible in the window, as presented below.



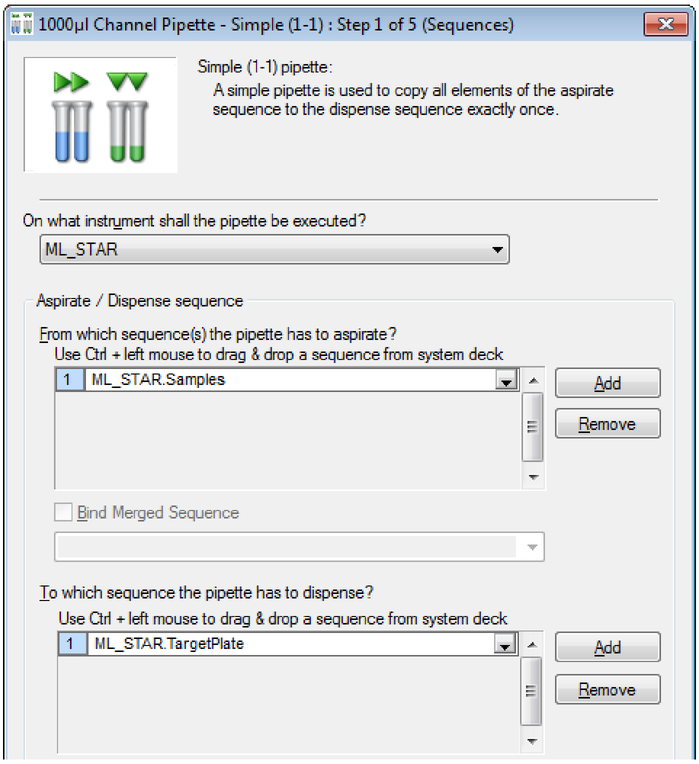
Pipetting 1:1

Add another "**Pipette 1000µl Channels**" Action to the right section of the window.

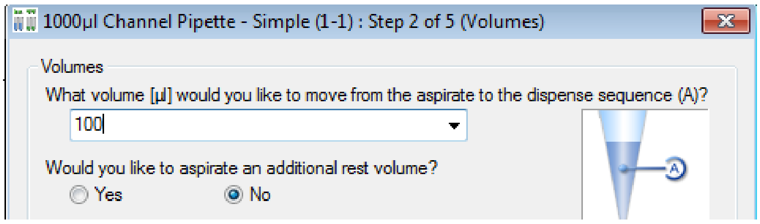


1. This time, change the display name to "**Pipette Samples**", change the color to light blue, set the "**Duration**" to 300 seconds and insert a "**1000µl Channel Pipette – Simple (1-1)**" Step.
2. Click **[OK]** to continue.
3. The "**1000µl Channel Pipette – Simple (1-1)**" Dialog will open.

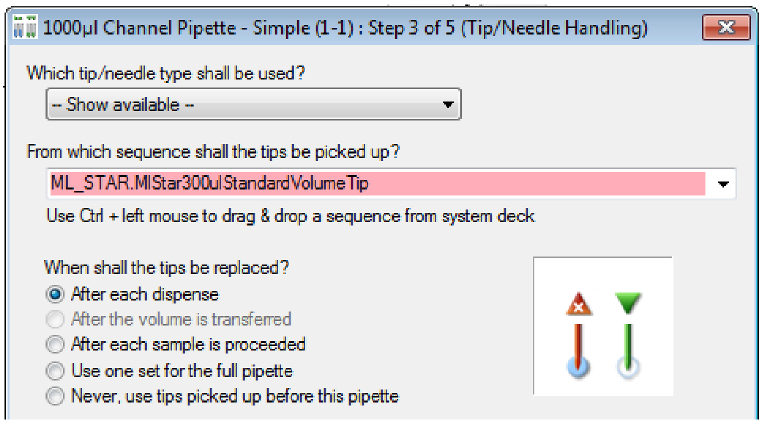
4. Insert the aspirate- and dispense sequences as shown. Click **[Next >]** to continue.



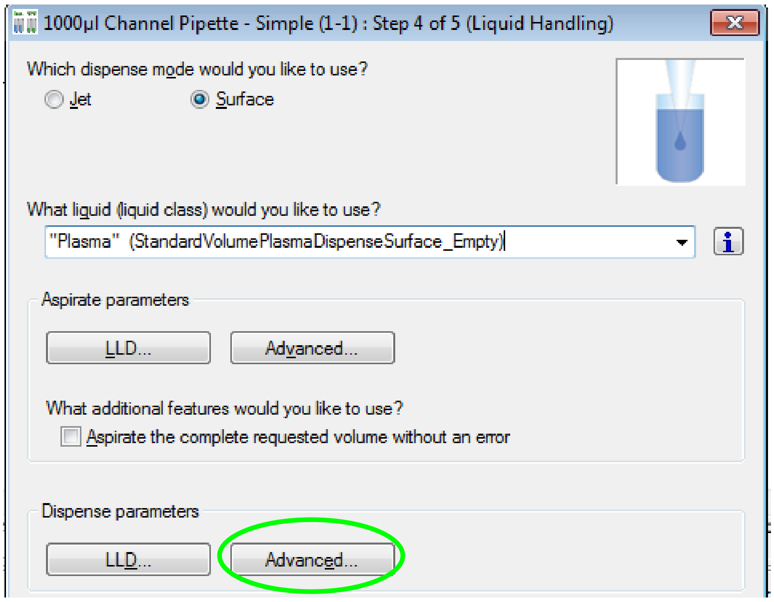
5. Set the volume in the next window to 100µl and Click **[Next >]** to continue.



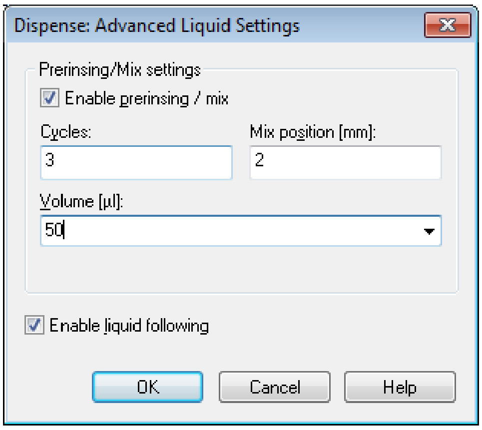
6. Next, make sure that the **[After each dispense]** Radio Button is activated.
7. Click **[Next >]** to continue.



8. Next, specify the dispense mode "**Surface**" and the liquid class "**Plasma**".

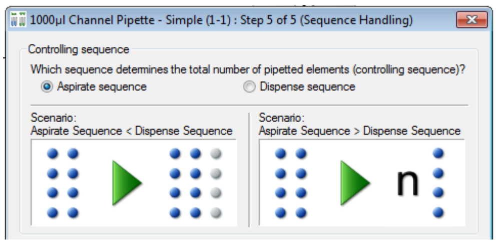


9. Under "**Dispense parameters**", click the **[Advanced…]** Button and enter the following values found on the next page.



10. Click **[OK]** to close this dialog and click **[Next >]** to continue.

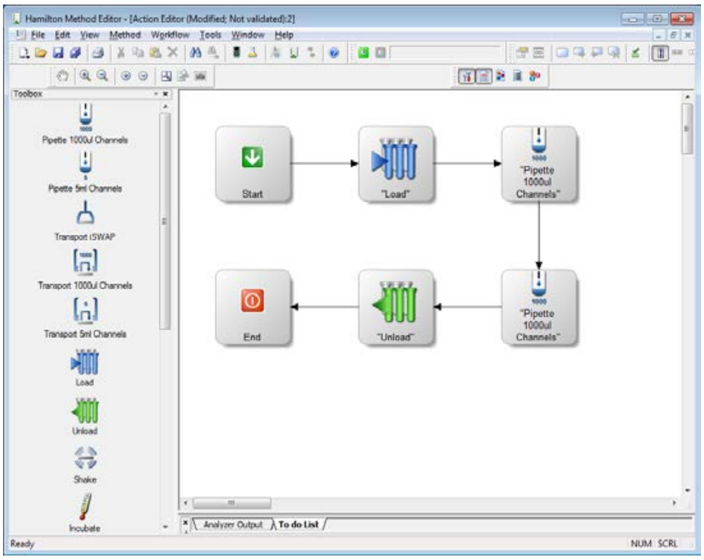
11. On the last step, activate the radio button to be used for the controlling sequence.

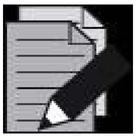


12. Complete by clicking **[Finish]**.

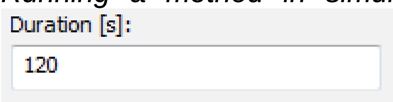
13. Add the “**Unload**” Action. Follow the previous steps in adding the action.

    The final method should look like the image below.

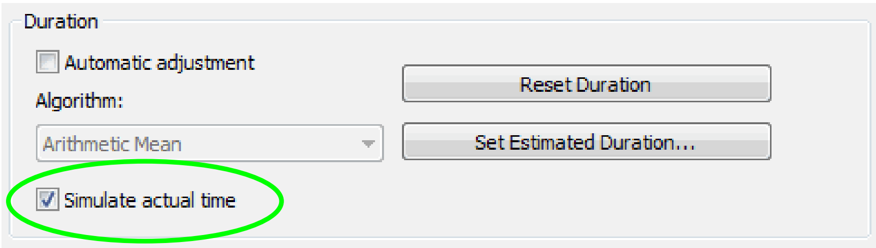




NOTE



*Running a method in simulator mode will consume the specified action duration. Here, the setting will generate a timer of 120 seconds. This can be switched off in the “**Advanced setting of the action**”.*



*Uncheck the “**Simulate actual time**” Box to NOT consume the specified duration.*

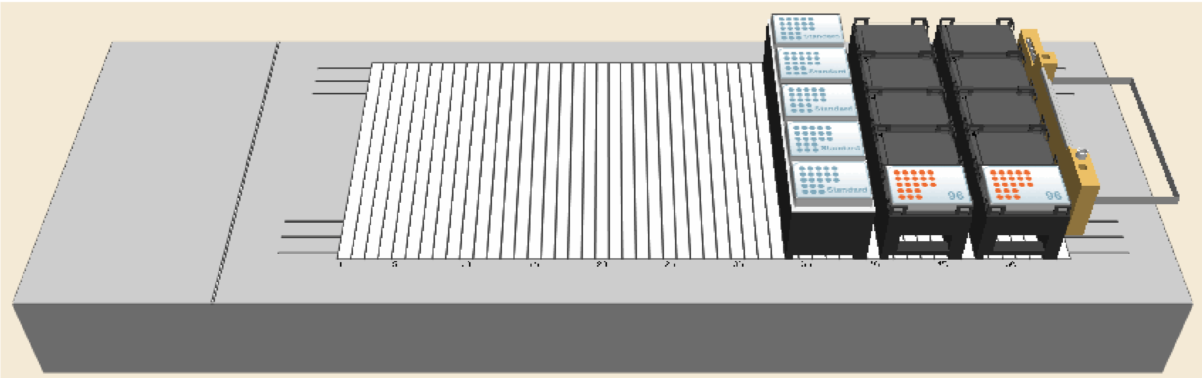
12.3 Method to Copy from Plate to Plate using Smart Steps

The method described in the following "**copies**" wells and does the following:

- Aspirates liquids from wells on a plate
- Dispenses liquid to the corresponding wells on another plate (A1→A1…H12→H12)

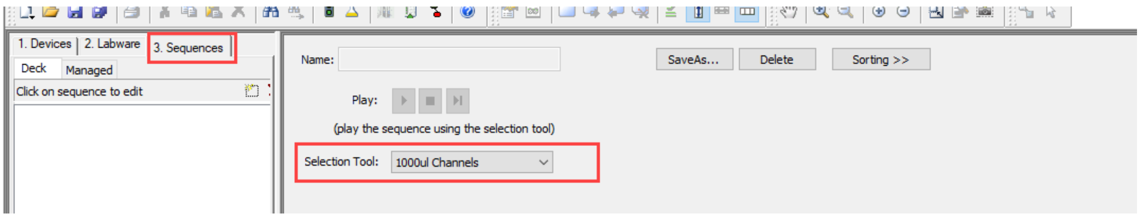
The transferred volume is 20µl. The method is named "**OnePlateToPlatePipette**". The method uses new CO-RE tips for every well. We will start with an empty Target Plate.

First, an appropriate Deck Layout must be created and saved as "**OnePlateToPlatePipette.lay**". The Deck Layout for this method is shown in the following picture.



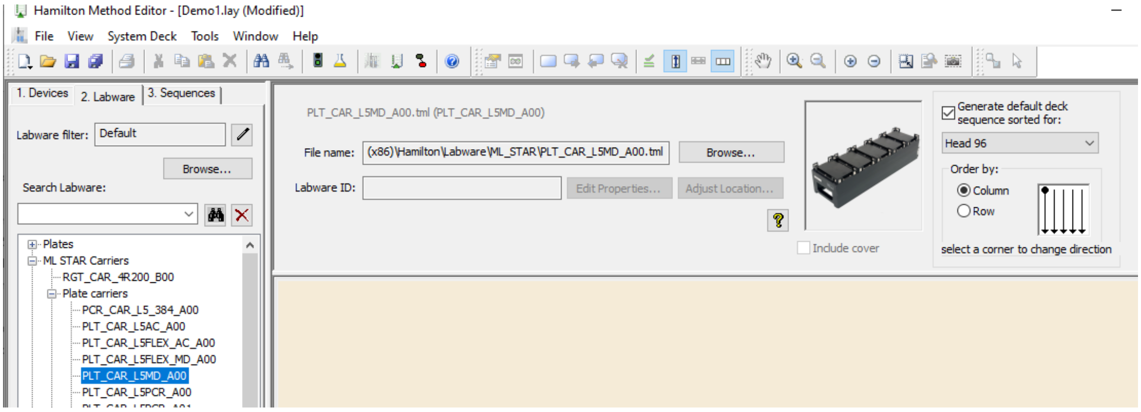
Creating the Deck Layout:

1. Start the HAMILTON VENUS application from the desktop shortcut, navigate to System Tools and click on the "**Hamilton Method Editor**" Shortcut.
2. Select "**File → New → Method**" to create a new method. A window opens to save the new method.
3. Enter the same filename and click **[Save]**. A new method window and System Deck Window are opened in the Method Editor. To activate, click the System Deck Window.
4. Make sure the Stamp Tool "**1000µl Channels**" is selected.



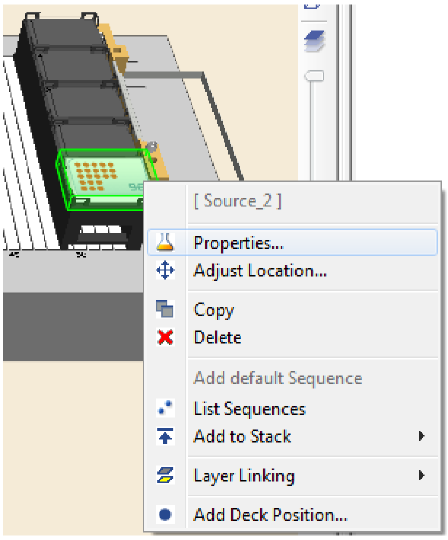
5. Click on the "**Labware**" Tab.

6. In the list above the tabs, click “**Microlab STAR Carriers → Plate Carriers**” and select “**PLT_CAR_L5MD_A00**”. This is a carrier for Micro Plates. An image of the selected Carrier will then be shown in the right-hand box.



7. “Drag-and-Drop” the carrier from the box onto the deck.

8. Right-click on the Carrier. From the **Context Menu**, select “**Properties**”. In the “**Properties**” Dialog, assign a LabwareID to the Carrier, e.g. “**SourceCarrier**”.



9. Repeat the steps above to create a second Carrier with the name (LabwareID) “**TargetCarrier**”.

10. Same as the previous steps, add a Tip Carrier with 300µl tips (without filter).

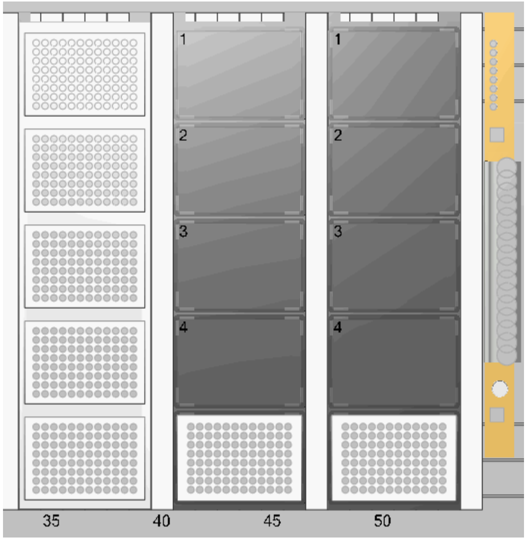
11. Click “**Microlab STAR Carriers → Tip Carriers 96**”. “Drag-and-Drop” the TIP_CAR480_BC_ST_A00 onto the deck. Change the LabwareID of the Carrier to “**TipCarrier**”.

12. Following the previous steps, add a plate from the “**Plates → 96 position plates**” to the “**SourceCarrier**” and the “**TargetCarrier**”. Open the “**Properties**” with a right-click on the plate and rename the plates to “**SourcePlate**” and “**TargetPlate**”.

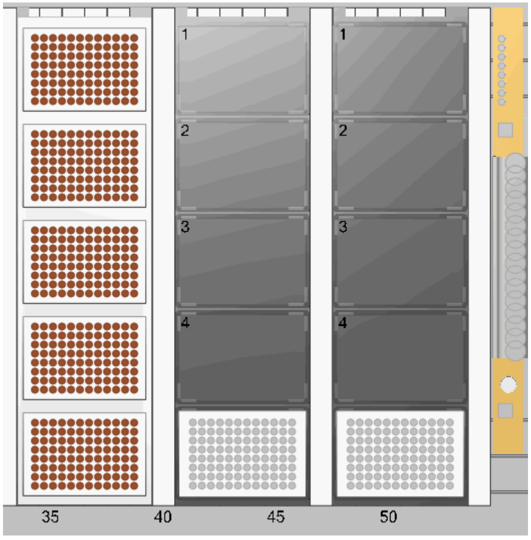
13. Select “**File → Save**” to save the Deck Layout.

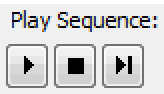
Creating the Sequence

1. Click the "**Sequences**" Tab to start editing the sequences.
2. The window will be switched to sequence editing.
3. Use the magnifying glass to zoom to the region of interest – the two plates and the Tip Rack.



4. Click the Tip Rack entry in the "**System Managed Sequences**" List.

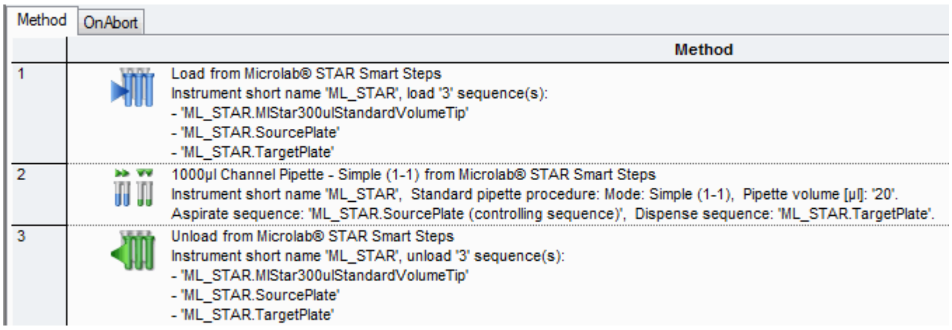


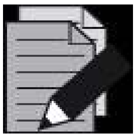
5. The positions of the selected sequences are highlighted in the System Deck View.
6. If the **[Play]** Button is clicked 

, observe that one column of dots at a time changes its color for a short time, beginning at the left column, running from left to right. Simply click the **[Stop]** Button in the toolbar to stop the play.
7. The order in which the tips will be processed is optimal for an 8 pipetting channel Microlab instrument. Note that the default name is given to the sequence automatically. To change or define a sequence name, select the sequence, and click the **[Save as]** or **[F2]** Button.
8. Now, click the source plate entry in the "**Deck Sequences**" List. Observe that in playing the process order, the single wells will also be processed column-by-column, from left to right. Stop the play.

Creating the Method

1. Open the Steps View of the Method Editor through this icon: 
2. Writing a method can easily be performed by dragging step icons from the toolbox and dropping them into the method window on the right section of the window. The resulting method will look like the picture shown below.



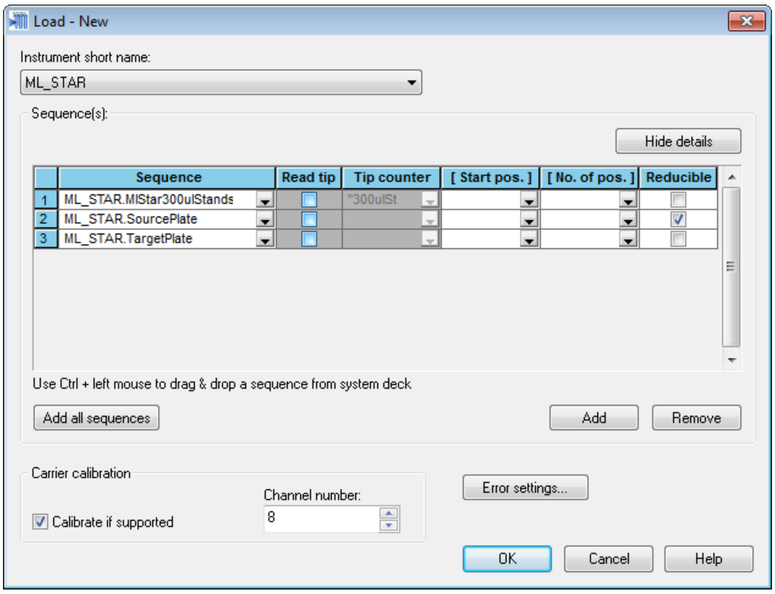


NOTE

For safety reasons, explicit loading (specifying loading commands within the method) is recommended. If no loading commands are specified, no checking of the carrier positions is performed, and the user must ensure that all carriers are positioned manually on the correct tracks.

The system will be initialized automatically when using Smart Steps.

3. "Drag-and-Drop" the "**Load**" Smart Step into the method window.



4. Click on **[Add all sequences]** to make sure that all Carriers are going to be loaded onto the instrument deck.

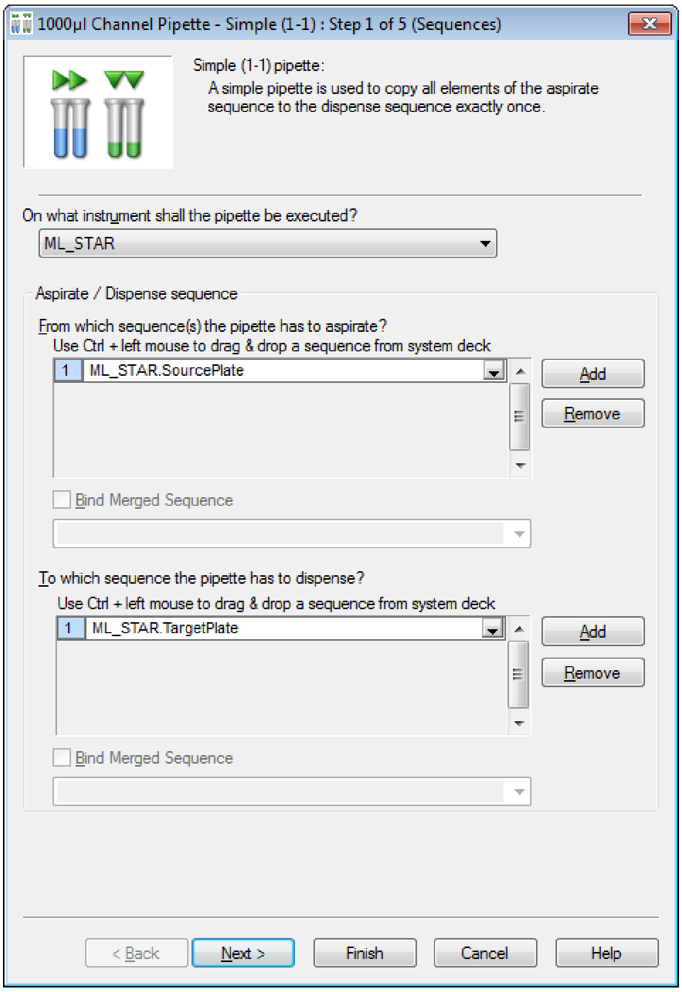
5. Click on **[Show Details]** and leave only the "**Reducible**" Checkbox for the source sequence enabled, if it is only during run time that the decision on which (or how many) of the wells of the Source Plate is to be transferred to the Target Plate will be made.

	Sequence	Read tip	Tip counter	[Start pos.]	[No. of pos.]	Reducible
1	ML_STAR.MlStar300ulStands	☐	"300ulSt			☐
2	ML_STAR.SourcePlate	☐				☑
3	ML_STAR.TargetPlate	☐				☐

6. For an instrument with the Autoload option, this command loads the Carriers automatically onto the instrument deck during run time. For a manual load instrument, this command requests the user to load the Carriers for run time.
7. Click **[OK]** to continue.

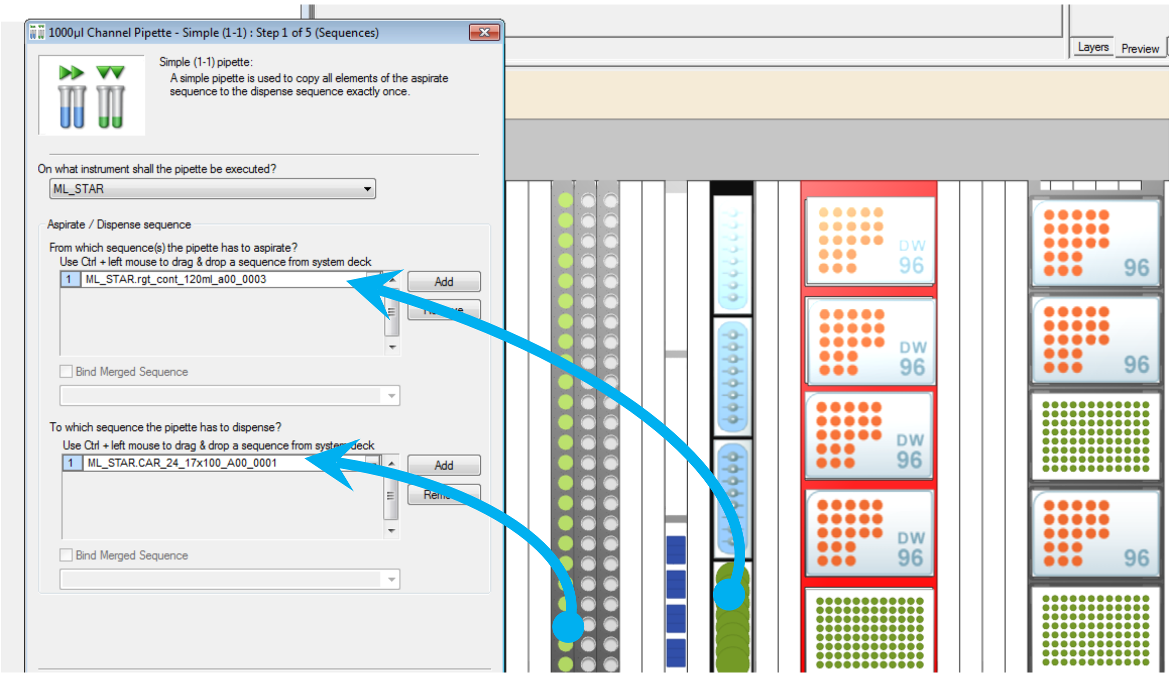
Program the Liquid Handling Details

1. Drag the Smart Step "**1000µl Channel Pipette – Simple**" to the line below the loading step (a simple transfer from plate-to-plate).
2. Activate the drop-down list and select the instrument used for pipetting. In this example, the "**Microlab STAR**" is selected.
3. Select the source and target sequence.



or

4. “Drag-and-Drop” the sequences from the Deck Layout into the input fields.



5. Click **[Next >]** to continue.

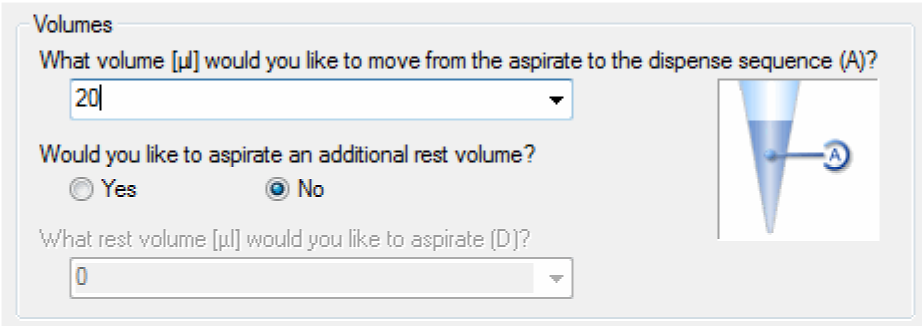
NOTE

*The “**Aspirate**” and “**Dispense**” Sequences can be selected in the Graphic Deck Layout View and then “Dragged and Dropped” onto the list fields in the “**Step Wizard**” Dialog (Use Ctrl + left mouse click to “Drag-and-Drop” from deck to step).*

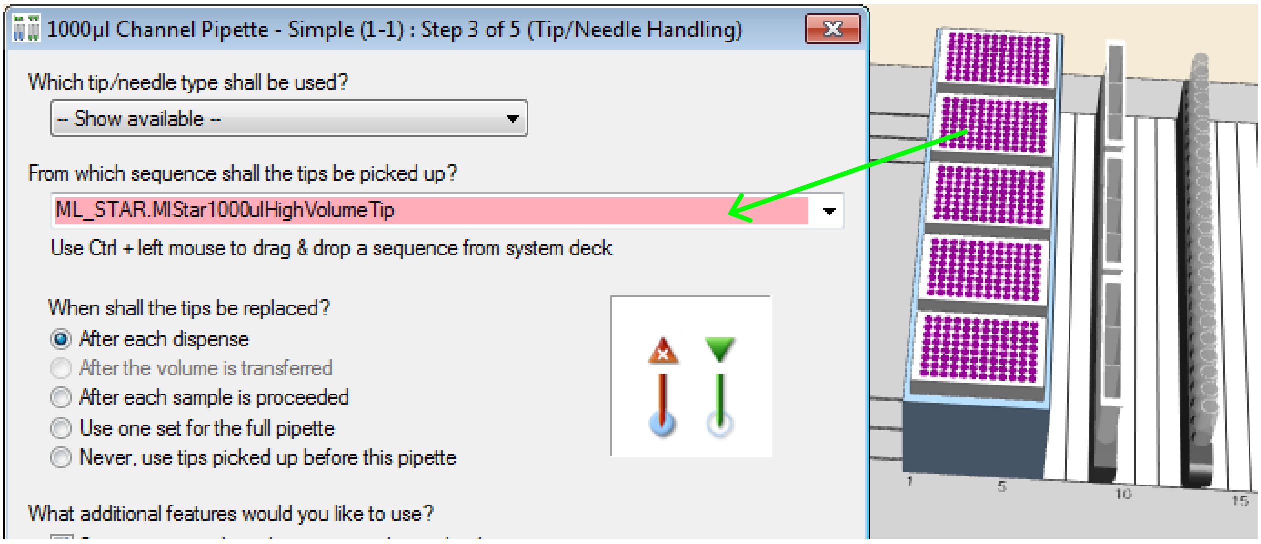
The two windows (Deck Layout View and Step Wizard) are interactive: when a sequence in the step wizard is clicked, this sequence is highlighted in the corresponding color. All aspirate / pick-up sequences are shown in blue, while all dispense/eject sequences are shown in green.

Having a sequence grid in the step (e.g. loading step), the inserted sequences will not only be highlighted but will also be flashing.

6. Here, the volume to be transferred is specified – 20µl; no additional volume should be aspirated. In this field, it is also possible to use a variable or an array.

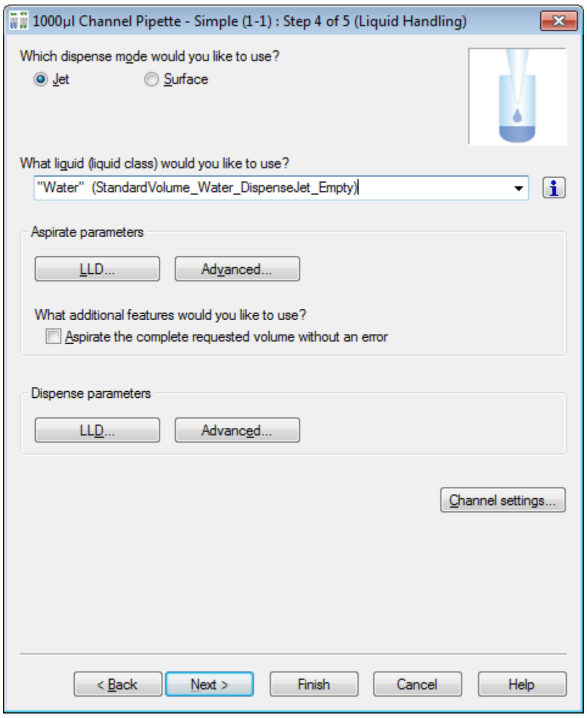


7. In this step, select a tip sequence (e.g. "**ML_STAR.MlStar300ulStandardVolumeTip**") from the drop-down field or "Drag-and-Drop" it from the Deck Layout. The tip handling chosen here is to take new tips for each sample.



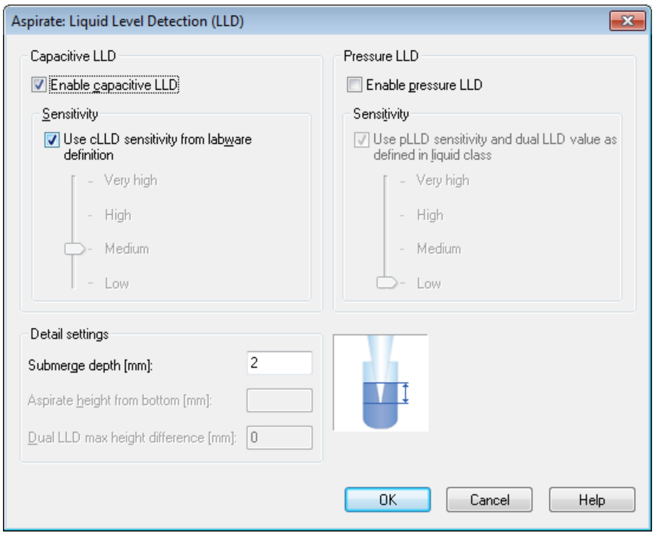
8. Click **[Next >]** to continue.

9. In this example, the dispense mode is set to "**Jet**" because initially, the target plate is assumed to be empty. The liquid class used in this example is water.

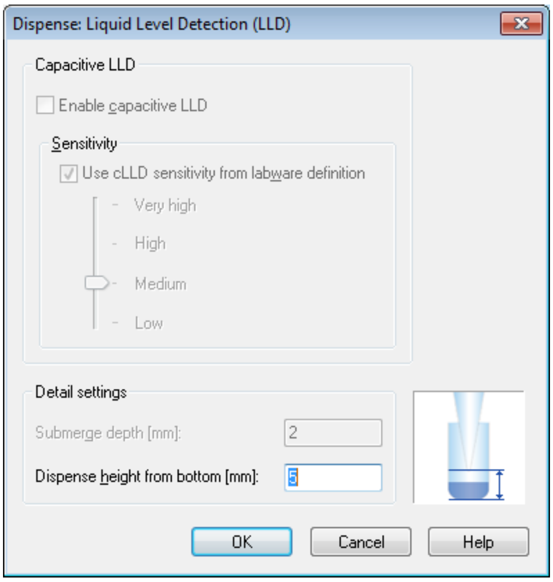


10. Click "**Advanced**" to go to the LLD Settings.

11. On aspiration, capacitive-based LLD may be used.

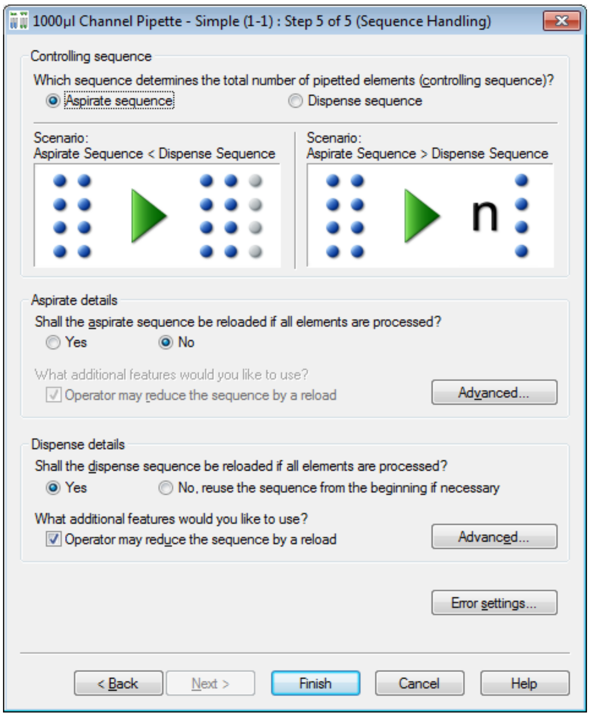


12. On dispense, set the dispense height from bottom to 5 mm.



13. Click **[OK]**.

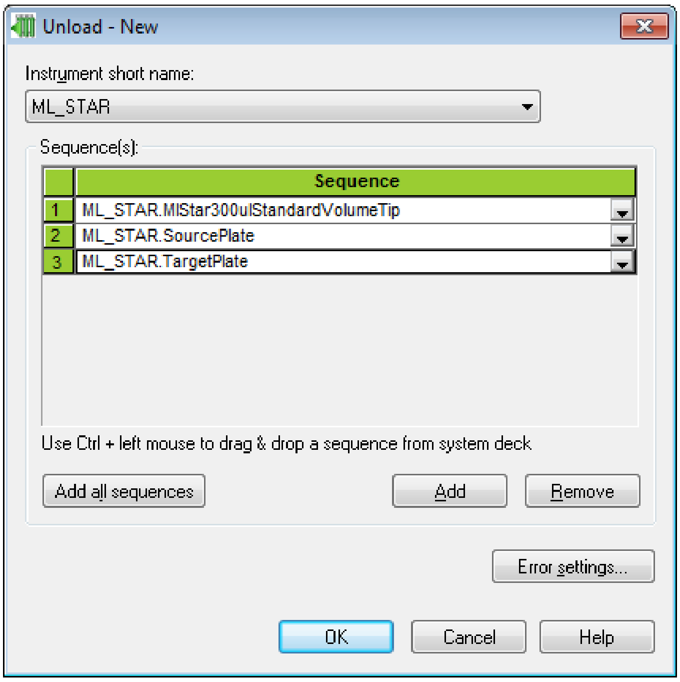
14. On the next step shown below, select the aspiration sequence as the controlling sequence.
15. Even if both sequences have the same length initially, it is reasonable to choose the aspiration sequence as the controlling one. Remember that on loading this sequence, the sequence may be reduced to fewer than 96 positions. Then, only the current number of wells is transferred to the target plate.



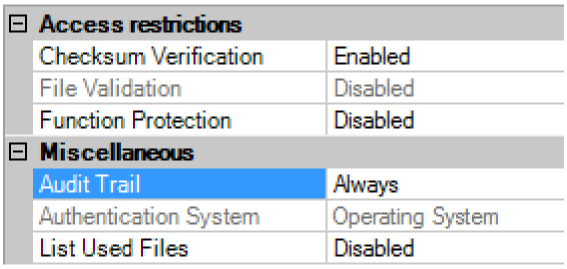
16. In the Aspirate details, set the reloading of the aspirate sequence to "**No**".
17. Accept the defaults for the settings under **[Advanced…]** and for the **[Error settings…]**.
18. Click **[Finish]** to close the "**Smart Step**" Dialog.

Unloading the Deck

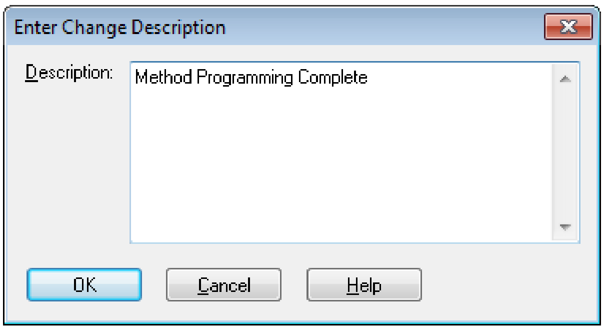
1. Finally, drag the "**Unload**" Smart Step to the line below the pipette step:



2. Click **[Add all sequences]** to add all sequences to the unload step and click **[OK]**.



3. Setting the Audit Trail (Always or Validation Only) will allow the possibility to enter change description.
4. Click "**File → Save**" in the Method Editor to store the method.



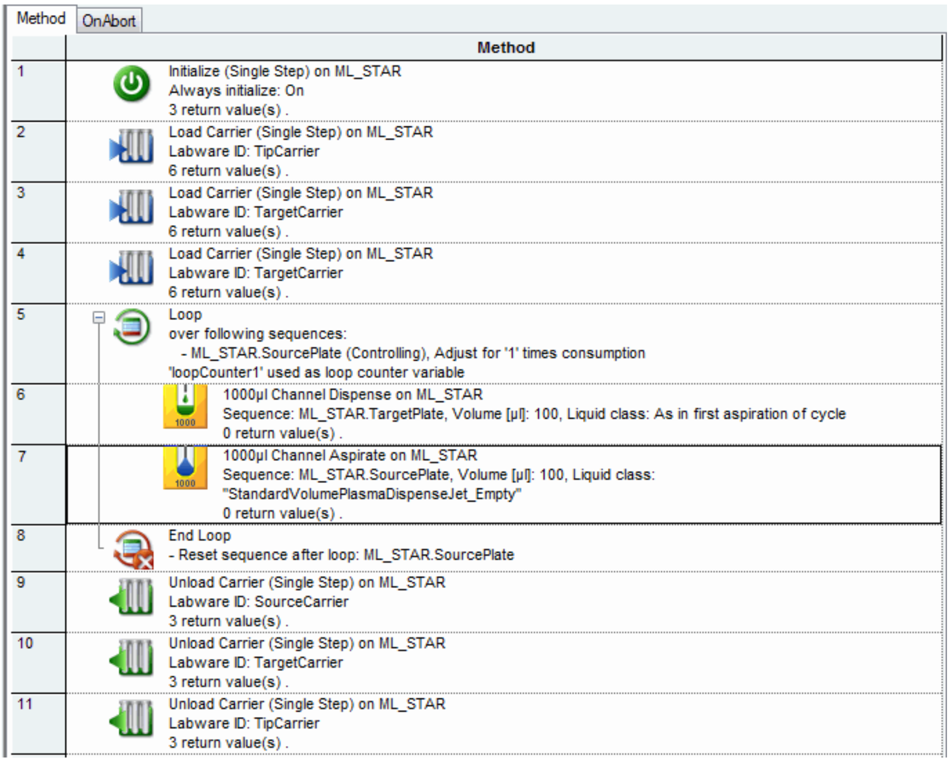
5. Enter the change description and click **[OK]**.

12.4 Method to Copy from Plate to Plate using Single and Easy Steps

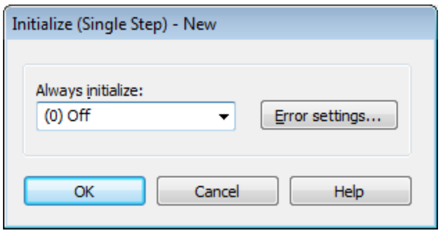
The method that will be discussed in this section does exactly the same as the method "**OnePlateToPlatePipette**" described in Section 12.3 Method to Copy from Plate to Plate using Smart Steps. The only difference is that the method is now written using Easy Steps instead of Smart Steps. Here, no sample reduction is possible.

This method uses the same Deck Layout as the method used with the Smart Steps.

1. Open this system Deck Layout and create the easy step version using the following guide:
2. Create a new method called "**OnePlateToPlateEasySteps**".
3. The method can be written easily by "Dragging and Dropping" the icons from the toolbox into the method window. The resulting method looks like this:



4. The first step is an initialize step. In the "**ML_STAR**" toolbar, drag "**Initialize**" to the main window. A window will be displayed.

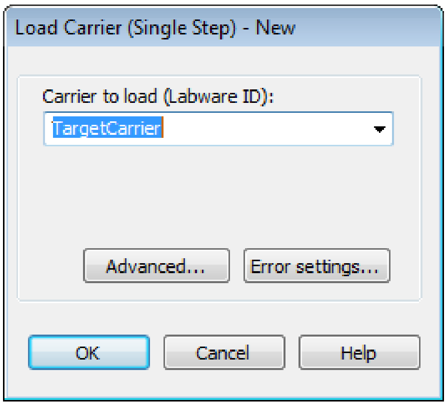


NOTE

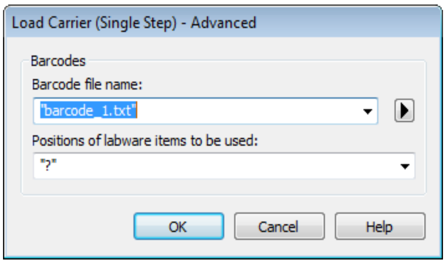
In using a single step in the beginning of a method, the initialize command has to be added prior to the single step.

When using Smart Steps and Easy Steps, the initialization will be executed automatically.

5. Drag "**Load Carrier**" into the next line of the method.

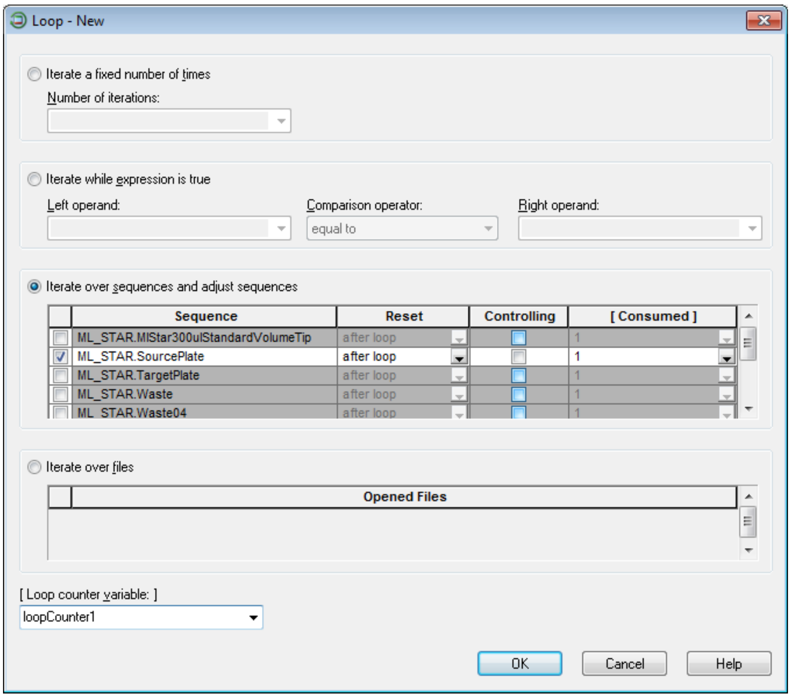


6. This command loads the carriers automatically onto the instrument deck during run time.
7. Specify the name of the carrier to be loaded.
8. Click **[Advanced…]** to display the path where the plate barcodes are stored, under the default file name "**barcode_1.txt**" (Note that the checkbox "**barcode trace**" within the Configuration Editor must be checked to generate the file). The positions on the deck are automatically retrieved from the Deck Layout during run time.



9. Click **[OK]** twice.
10. Repeat the "**Load Carrier**" Command for the Target Plate Carrier and the Source Carrier.
11. To copy the whole source plate and not just the first 8 wells to the Target Plate, the steps "**Tip pick-up**", "**Aspiration**", "**Dispense**" and "**Tip eject**" have to be performed 12 times (96 wells divided by 8 pipetting channels). This can be achieved using a loop command.
12. The loop statement consists of two lines, a "**begin loop**" and an "**end loop**" statement. The codes are inserted between these two statements will be looped.
13. Drag the "**Loop**" Command from the General Steps into the next line of the method window.

14. The “**Loop**” Dialog window will appear as follows:

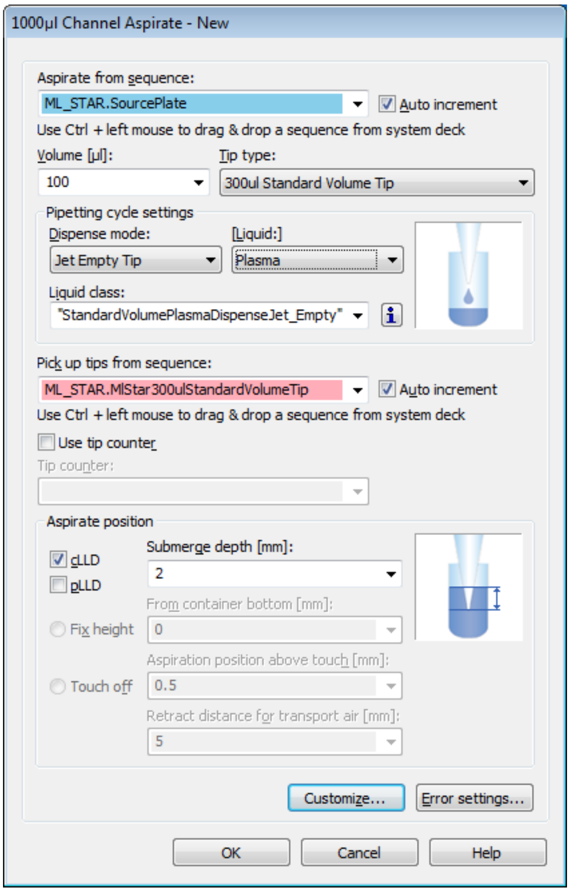


15. A loop can be performed, looping:
    - Over a fixed number of iterations
    - An expression (repeat which the statement in the expression is true)
    - A sequence
    - A file (until end-of-file is reached)
16. In this example, a looping over the “**SourcePlate**” Sequence is done. This means that the loop will continue until all sequence positions (the 96 wells) of the “**SourcePlate**” have been used. Only then the loop will stop.
17. Choose the default “**after loop**” for the “**Reset Sequence**” Option, to reset the sequence “**SourcePlate**” to the initial position (1) after the loop is done. If a pipette with the same sequence “**SourcePlate**” will be performed once more at a later time, the sequence will then start at the first well again.

NOTE

Keep in mind that in looping over a sequence, the sequence has to be incremented within the loop.

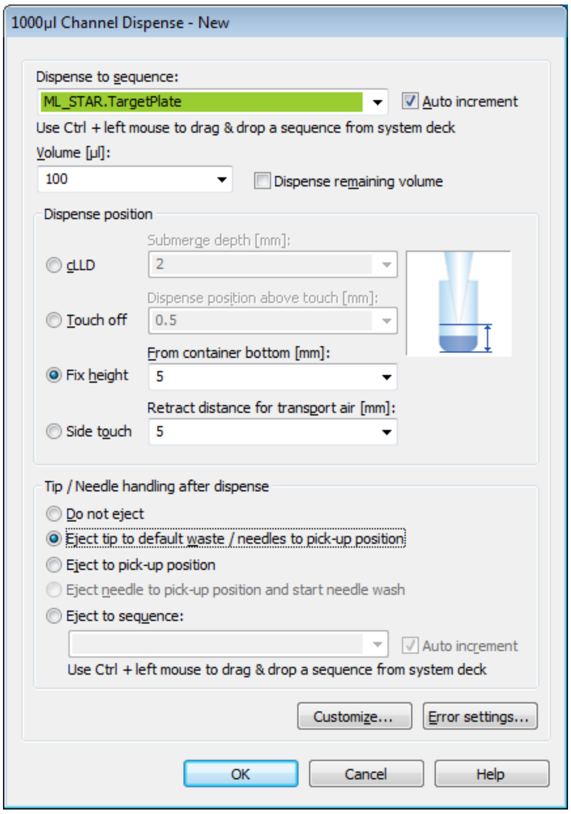
18. Now, drag the "**1000µl Channel Aspirate (Easy Step)**" to the line below the tip pick-up step. A dialog box, as shown below will appear. Specify the values indicated below the dialog box.



- Aspirate from Sequence: "**ML_STAR.SourcePlate**" (aspirate from source plate)
- Volume: 100µlTip Type: 300ul Standard Volume Tip
- Dispense Mode: Jet Empty Tip
- Liquid Class: StandardVolumePlasmaDispenseJet_Empty (For questions about these parameters, refer to Section 13.5 Pipetting)
- Submerge Depth: 2 mm

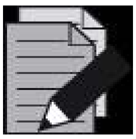
19. After completing all the fields, click **[OK]**.

20. Drag the “**1000µl Channel Dispense (Easy Step)**” to the position below the aspirate command. A dialog box, as shown below will appear. Specify the values indicated below the dialog box.



- Dispense to Sequence: “**ML_STAR.TargetPlate**” (Target Pplate)
- Sequence Counting: “**Auto increment**”
- Volume: 100µl
- Dispense Position: 5 mm, “**Fix height from container bottom**” (which corresponds to a height of 2 mm above the container bottom for dispensing)

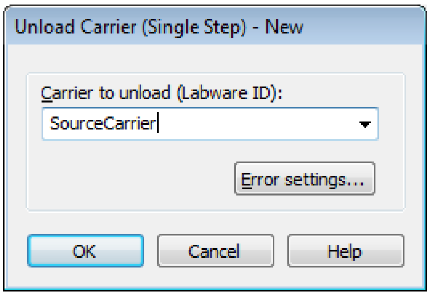
21. After completing all the fields, click **[OK]**. The loop is now complete.



NOTE

Using different liquid classes for aspiration and dispensing is possible, but not recommended.

22. Finally, the carrier must be unloaded outside the loop.
23. To unload the Carriers, drag the "**Unload**" Command (with the Carrier name as a parameter) into the method and drop it after the end loop.
24. The "**Unload**" Dialog appears:

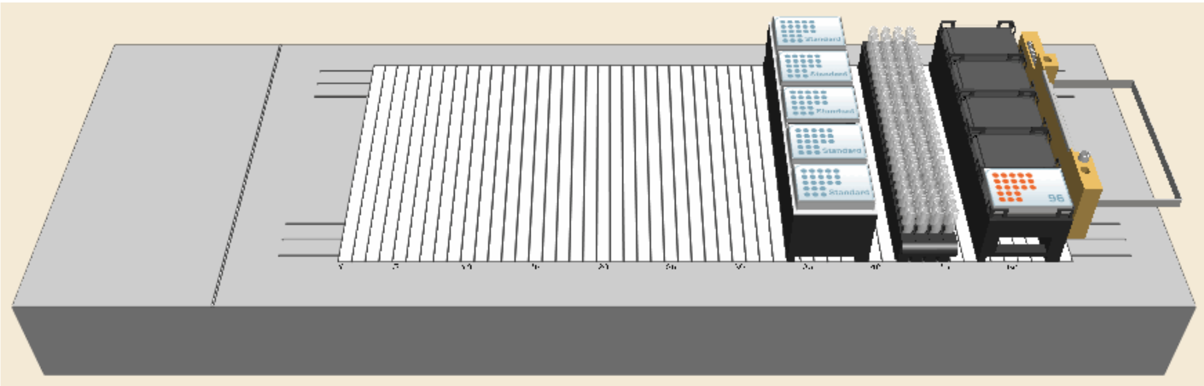


25. Click **[OK]**.
26. Insert "**Unload**" Commands for all three Carriers.
27. The method is now complete. Save the method and exit by selecting "**File → Exit**" in the Method Editor.

12.5 Method to Copy from Tubes to Plates using Smart Steps

This method copies tubes. It aspirates liquid from tubes in a Carrier and dispenses them into wells of a Micro plate. The maximum number of tubes to be processed is 96, corresponding to a maximum of four 24-tube Carriers (1T).

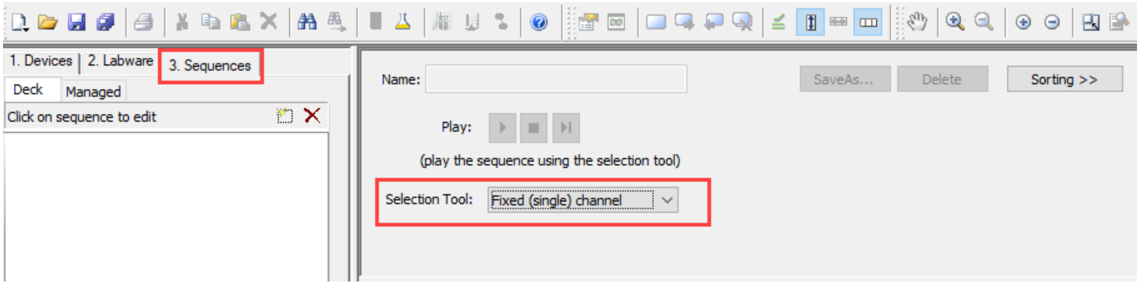
1. First, the appropriate Deck Layout must be created and saved. In this case, it will be "**TubesToPlatePipette.lay**". The Deck Layout for this method is shown in the following picture:



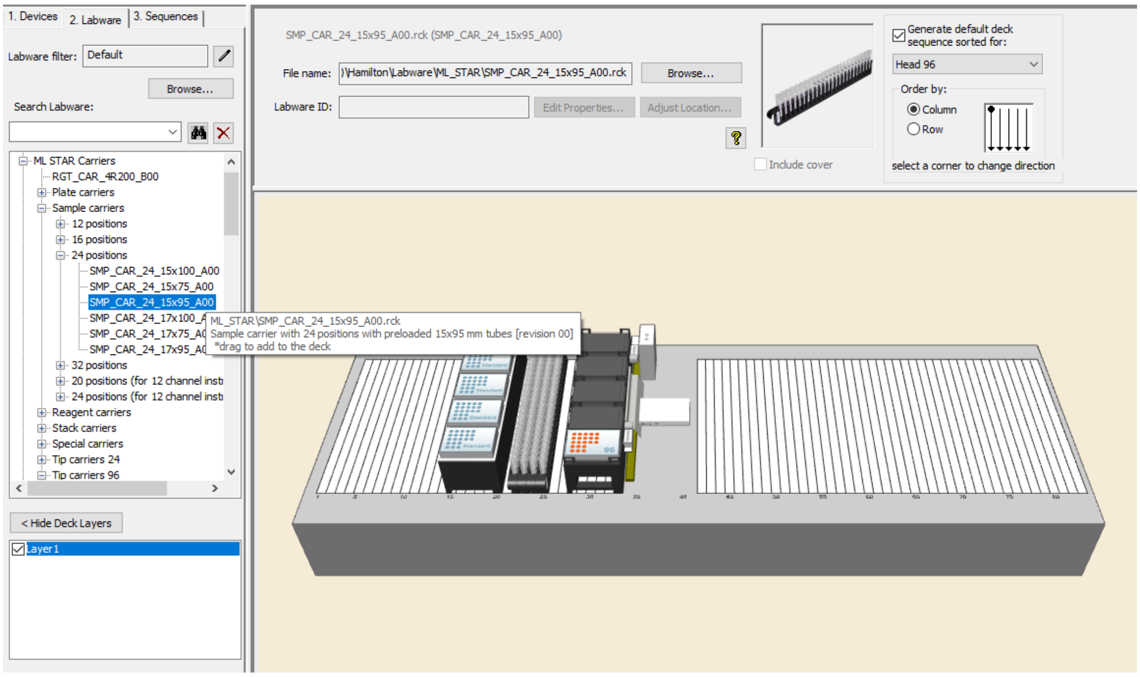
Creating the Deck Layout:

2. Start the Method Editor from the HAMILTON VENUS.
3. Select "**File → New → Method**" to create a new method. A window will open to be able to save the new method.
4. Enter the filename "**TubesToPlatePipette.med**" and click **[Save]**. A new method window and system deck window are opened in the Method Editor. To activate, click the System Deck window.

5. Make sure the Stamp Tool "**Fixed (single) channel**" is selected, as shown below.

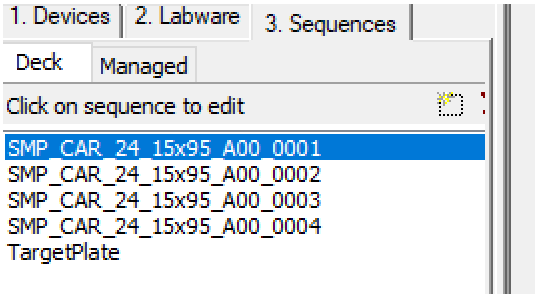


6. Click on the "**Labware**" Tab.
7. In the list above the tabs, expand the "**Microlab STAR Carriers**" by clicking the "+" right next to the entry. Do the same to the "**Sample carriers**". Select the "**24 positions**" Entry. A list of 24-position carriers is displayed.
8. Select the Carrier "**SMP_CAR_24_15x95_A00**" (rack holding 24 tubes with a 15 mm diameter and a height of 75 mm). An image of the selected Carrier will then be shown in the right-hand box.

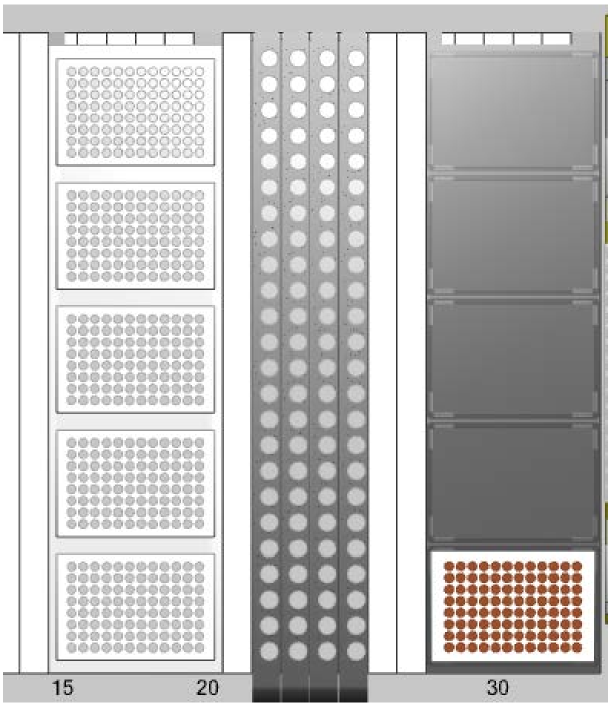


9. "Drag-and-Drop" the Carrier image onto the deck. Add the other three Tube Carriers onto the deck (adjacent to the first Carrier).
10. Following the steps above, add a Plate Carrier from the "**Microlab STAR Carriers → Plate Carriers**" list.
11. Select "**PLT_CAR_L5MD_A00**" **or** use the **[Browse]** Button to search for "**PLT_CAR_L5MD_A00.tml**" in the "**ML_STAR**" Labware sub-directory.
12. Same as the previous steps, add a target plate from the "**Plates → Nunc**" group.
13. Select "**Nunc 96 Fl Lb (low border)**" and "Drag-and-Drop" the plate onto the deck.
14. Change the LabwareID of the plate to "**TargetPlate**"
15. Following the same steps above, add a Tip Carrier with 300µl tips (without filter).
16. Click "**Microlab STAR Carriers → Carriers 96**" and "Drag-and-Drop" the TIP_CAR480_BC_ST_A00 onto the deck.

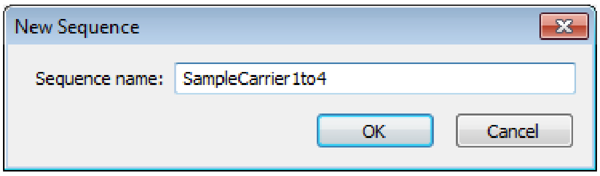
17. Select “**File → Save**” to save the Deck Layout.
18. Click the “**Sequences**” Tab to start editing the sequences.
19. At this point, the sequence window should look like the image presented below.



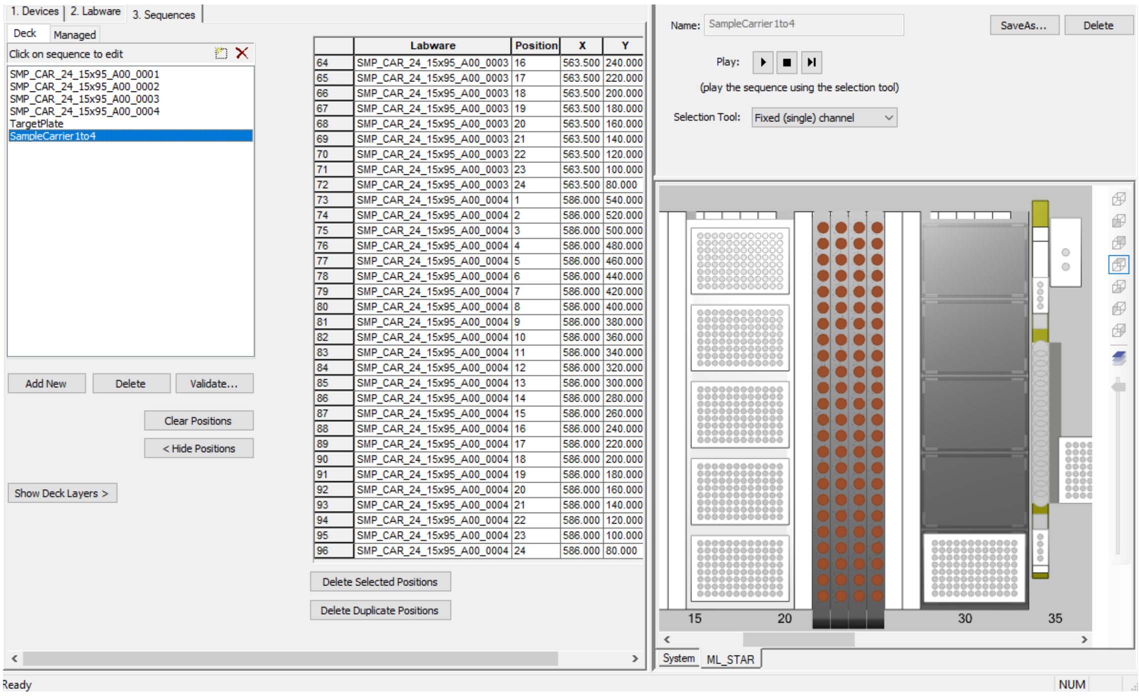
20. Zoom-in using the zoom in/out toolbar or the “**View**” Menu:



21. Click **[Clear Positions]** to ensure that no sequence or positions are selected before the next step .
22. Rubber band all four tube carriers by performing a left-click (do not release) followed by moving over all of the Sample Carriers. The 96 selected tubes of the four Carriers are then highlighted.
23. Give a name to create a new sequence from the selection.



24. Click the **[Show Positions]** Button to see all the selected positions in a table. The table (presented below) shows the positions of the sequences holding all tubes.



25. As long as no sorting option is selected, the sequence is sorted as follows:

“**SMP_CAR_24_15x95_A00_0001**”: Position 1 to 24

“**SMP_CAR_24_15x95_A00_0002**”: Position 1 to 24

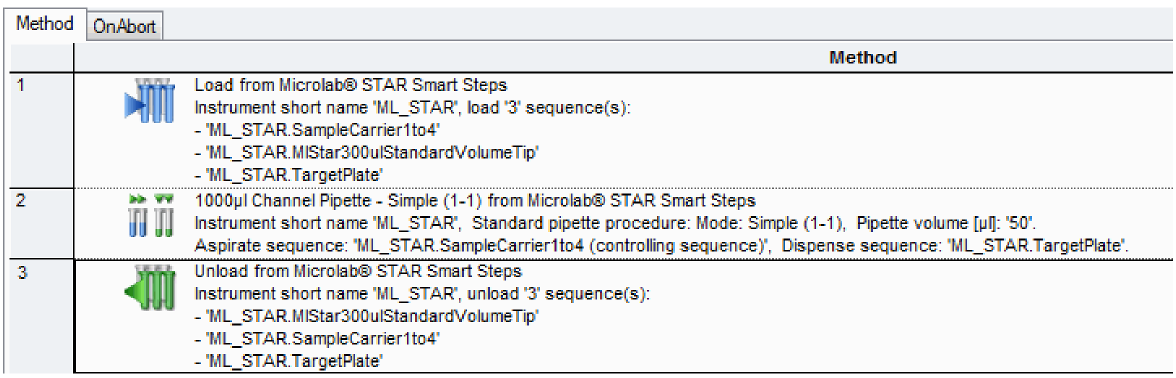
“**SMP_CAR_24_15x95_A00_0003**”: Position 1 to 24

“**SMP_CAR_24_15x95_A00_0004**”: Position 1 to 24

26. Select “**File → Save**” to save the Deck Layout. This is the end of the Deck Layout and sequences creation.

Creating the Method

1. Open the Steps View of the Method Editor. Add all the steps by “Dragging-and-Dropping” icons from the toolbox into the Method Window. The resulting method should appear as presented below.



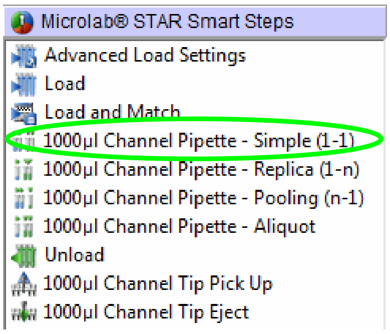
2. Add the Smart Step “**Load**” by “Dragging-and-Dropping” it from the toolbar to the Method Window. The “**Load**” Dialog appears:

3. Click on the drop-down found in the "**Sequence**" Field and select the sequence "**ML_STAR.SampleCarrier1to4**".
4. Click **[Show details]** in order to check the column "**Reducible**".
5. Following the previous steps, add the "**ML_STAR.MlStar300ulStandardVolumeTips**" and "**ML_STAR.TargetPlate**" Sequences.
6. To do so, use the **[Add]** Button from the "**Load**" Dialog. Finish with **[OK]**.

NOTE

*Be reminded that the sequences to be loaded can be selected through the graphical Deck Layout View and by using the "Drag-and-Drop" technique on the list fields in the "**Load**" Dialog (with Ctrl + left mouse "Drag-and-Drop"). The two windows (Deck Layout View and "**Load**" Dialog) are interactive: when a sequence in the "**Load**" Dialog is clicked, this sequence flashes with a specific color in the Deck Layout View.*

7. Drag the Smart Step "**1000 µL Channel Pipette – Simple (1-1)**" to the next line in the method:



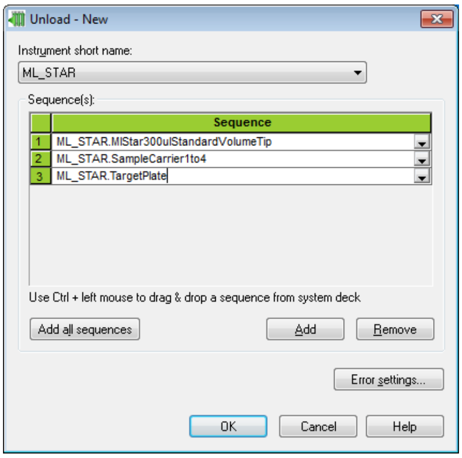
8. In the "**Smart Step**" Wizard screens, select the following:

Step 1:	- Aspiration sequence: "**ML_STAR.SampleCarrier1to4**" - Dispense sequence: "**ML_STAR.TargetPlate**"
Step 2:	- - Volume = 50µl, no additional residual volume
Step 3:	- Tip sequence: "**ML_STAR.MlStar300ulStandardVolumeTips**" - Tip handling: "**After each dispense**"
Step 4:	- Liquid Class: Water "**StandardVolume_DispenseJet**" - LLD on aspiration: settings are capacitive, sensitivity low, submerge depth 2 mm - LLD on dispense: fixed height 2 mm from bottom
Step 5:	• Controlling sequence: "**Aspiration sequence**" • Aspiration details: no reload • Dispense details: reload may be selected. This is of no relevance here, since the aspiration sequence is controlling and as long as, or

	even shorter (by reduction on run time) than the dispense sequence

9. Click **[Finish]** to continue.

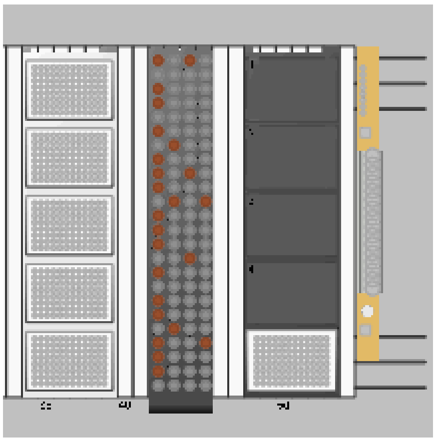
10. Drag the “**Unload**” Smart Step into the next line. Click on the list to add the three sequences into the list of sequences to unload. Finally, click **[OK]** to add the “**Unload**” Step into the method.



What happens when running this Method?

11. Within the loading step, the sample sequence was chosen to be reducible. This means that, at run time, the following dialog is shown, enabling reduction in the number of samples (from any position). The user may reduce the number of samples down to 24. The altered sequence is immediately shown in the Deck View. The method processes 24 tubes (in this case, from the first Carrier) to the plate and then stops.

	Labware positions	First	Last	Remove All	Removed	Remaining
1	SampleCarrier1to4	1	93		72	24
2	MlStar300ulStandardVolumeTip	1	480		0	480
3	TargetPlate	1	96		0	96



12. It is also possible to deselect particular tubes from the sequence by clicking on the wells or by using the rubber band. These positions will then turn to gray. A **[Reset]** Button is available to restore the original sequence.

12.6 Method for Hit Picking using Smart Steps

Assume that there is a source plate and a photometer reads the optical absorbance of the wells of the plate. Create a target plate with all the compounds in the source plate having an absorbance of A>1.0E.

The 'hit picking' method does exactly this. The photometric results are retrieved from a file, and a sequence of hits (A>1.0E) is created 'on the fly' according to the absorbencies that have been read. Pipetting then occurs according to the sequence.

For this method, a source containing the absorbencies of the 96 wells of the source micro plate is needed. The database can be any of the following:

- ASCII text file
- Microsoft Excel file
- Microsoft Access database
- Vector database

In this example, the database used is a Microsoft Excel file. The name of the Excel sheet being used is "**Absorbance**". The sheet contains three columns, as follows:

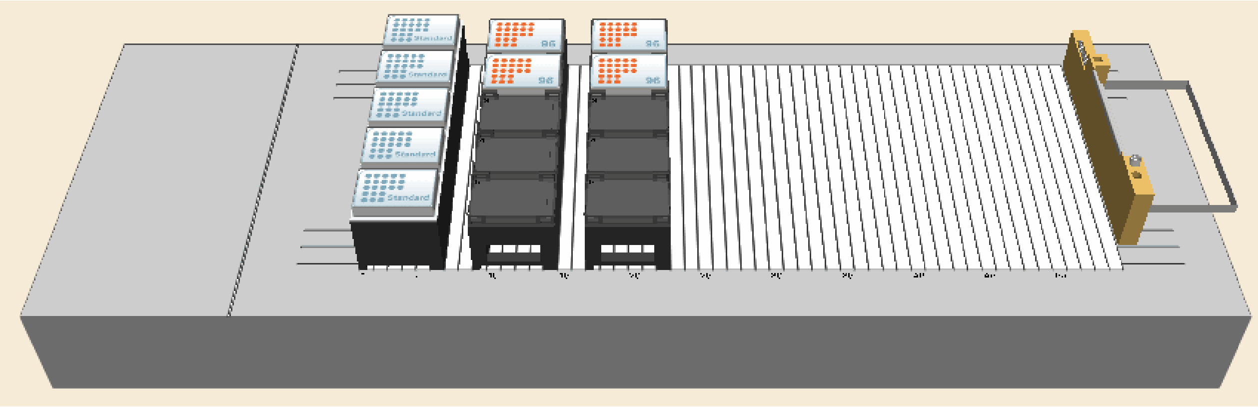
- "**LabID**" defining the plate name of the source plate ("**Source_1**" for the first plate and "**Source_2**" for the second plate)
- "**PosID**" defining the position in the micro plate alphanumerically (A1, A2, …, H12)
- absorbance or optical density "**OD**" in mE

The worklist then has 193 lines, 1 header line and the entries from 2 plates with 96 wells each.

	A	B	C
1	LabID	PosID	OD
2	Source_1	A1	310
3	Source_1	B1	233
4	Source_1	C1	564
5	Source_1	D1	265
6	Source_1	E1	32
7	Source_1	F1	697
8	Source_1	G1	310
9	Source_1	H1	233
10	Source_1	A2	564
11	Source_1	B2	265

The Deck Layout contains (see picture below):

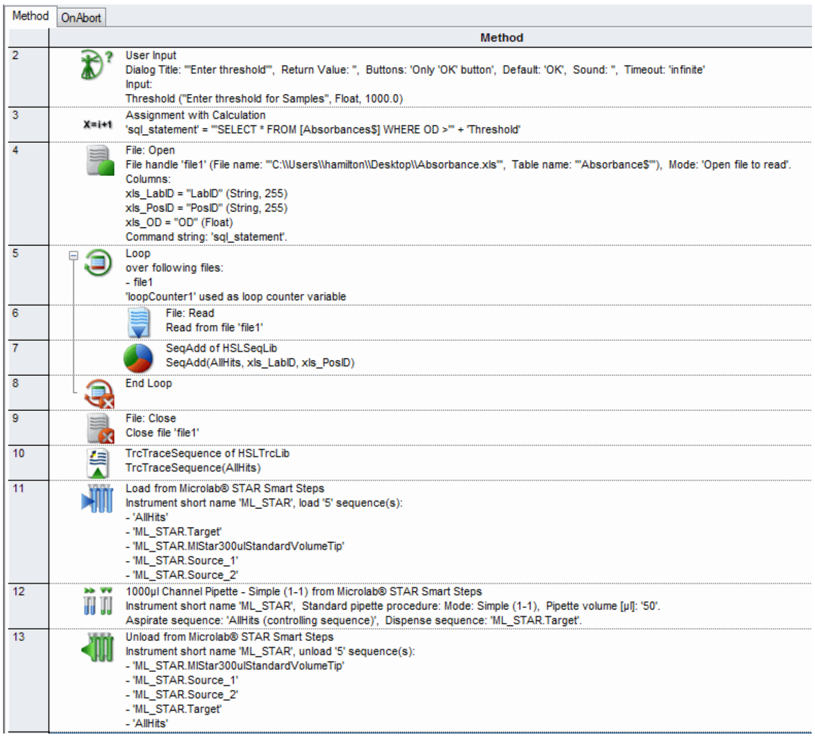
- One Tip Carrier "**TIP_CAR_480_ST_A00.tml**" preloaded with five standard volume tip racks
- One Plate Carrier "**PLT_CAR_L5MD_A00.tml**" with two Nunc plates "**Nun_96_Fl_L.rck**", which are the source plates (change the properties such that one plate is called "**Source_1**" and the other "**Source_2**")
- One Plate Carrier "**PLT_CAR_L5MD_A00.tml**" with two Nunc plates "**Nun_96_Fl_L.rck**", which are the target plates



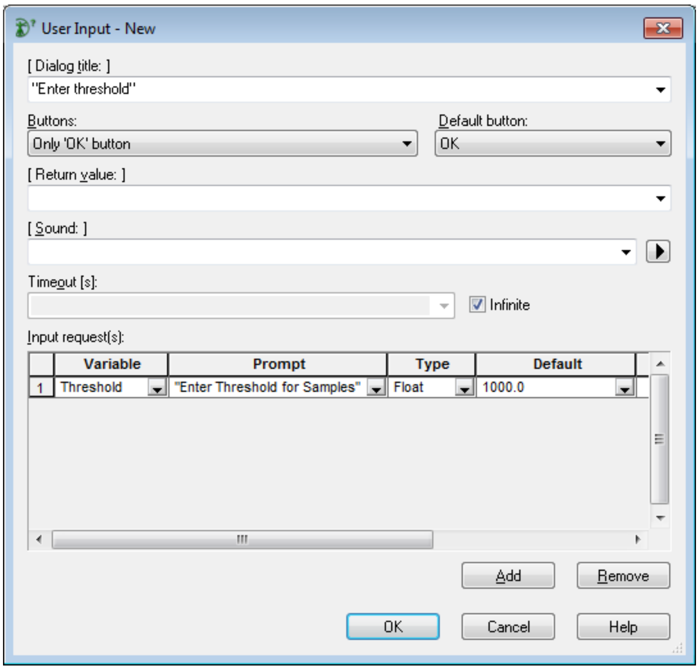
1. Create the “**Target**” Sequence by activating the “**Sequences**” Tab in the Deck Layout Editor.
2. Click on **[Clear selected]**.
3. Activate the “**Channels**” Stamp Tool and create a sequence spanning the two target plates and name it as “**Target**”.
4. Switch to the Method Editor window.
5. Include the two libraries “**HSLTrcLib.hsl**” and “**HSLSeqLib.hsl**” in the method.

Creating the Method

1. “Drag-and-Drop” the steps from the toolbox into the method, as shown below.

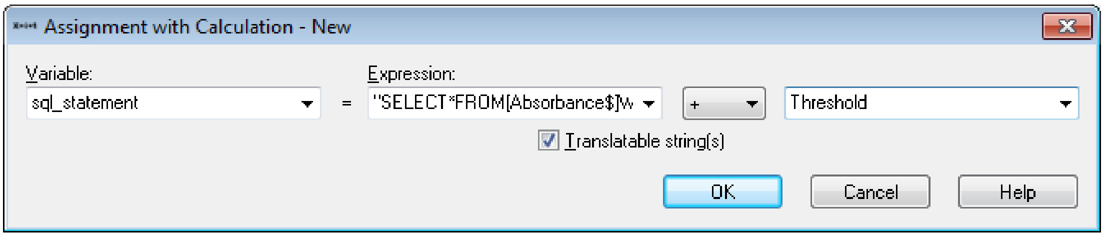


2. The first step is to request the user to input a threshold value for the absorption. This is saved as a variable called "**Threshold**".



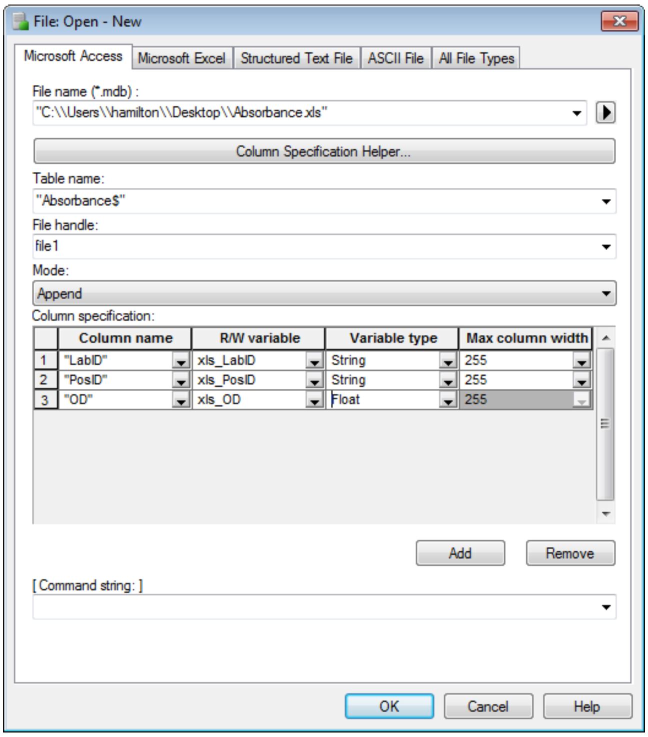
3. In the "**Prompt**" Field, type in the question to be asked to the user (do not forget to put the string in quotation marks, for this is a text).
4. Specify the "**Type**" (integer, float, string) of the input required. In this case, use a float type because this value will be used in an SQL statement.
5. In the next fields, a default value that is already displayed when the dialog opens can be specified. Having an integer or float value, also minimum and maximum values can be defined.
6. Click **[OK]** to finish the user input.
7. In the next step, create an SQL statement to only get the values with the specified OD or greater. The complete expression in the left field is:

   "**SELECT * FROM [Absorbance$] WHERE OD >**"



8. This will create a variable SQL_Statement which depends on the user input, e.g. if the user types in 500 as threshold, the statement to open the excel file will be "**SELECT * FROM [Absorbance$] WHERE OD > 500**"
9. Here, "**Absorbance$**" refers to the name of the Excel sheet which is going to be opened during the next steps.
10. Drag the "**Open File**" Icon into the method to open the file holding the absorbance information. The file format is also defined within this step as follows:
11. Select the tab for opening "**Microsoft Excel**" Files.

12. Under “**File name (*.xls)**”, enter the file name. When browsing for the file, the full absolute path is required with the filename. Note that the backslash must be typed twice. To use a relative path, delete it or simply enter the path enclosed in quotation marks.



13. The file type is an Excel file (.xls) where the sheet name (sheet1 if not defined otherwise, language-dependent) and the $ sign must be added within the quotation marks.

14. Define a file handle (here the default: “**file1**”) which is a name for the file used within the method. Later, data will be read from this file by referring to this file handle.

15. Select “**Open File to Read**” as the mode to read data from this file.

16. Now, define the file format. Here, a variable is assigned to each column of the file. Later, for each reading step, one record (one line) is read from the file, and the contents read are assigned to their corresponding variables automatically.

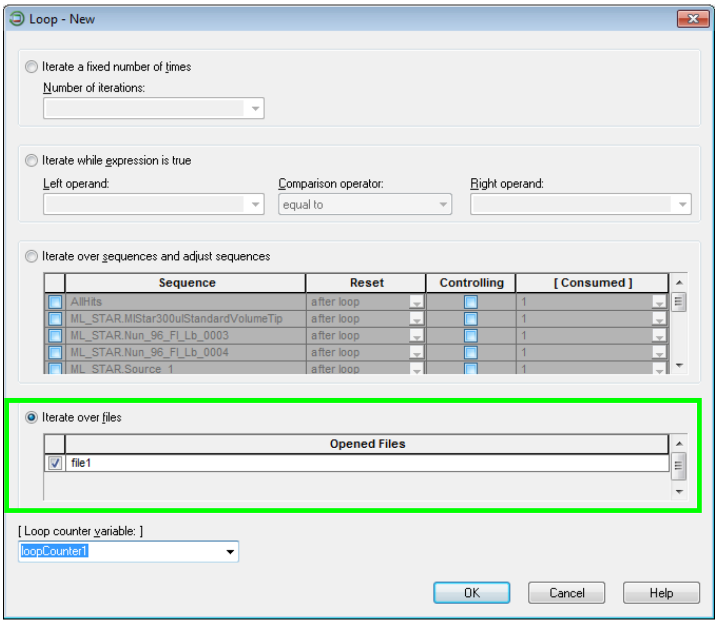
17. Now define the file structure. One line in the “**Column specification**” of the file opening dialog represents one column in the file. Click on the **[Add]** Button to add the next line to the dialog.

18. Enter the data as given in the screenshot above. Assign a header, a variable, and a variable type to each column.

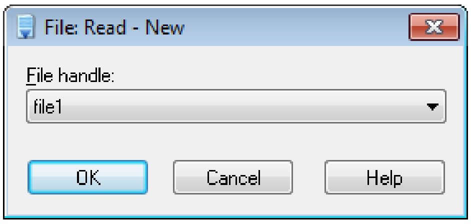
19. Select the variable “**sql_statement**” from the drop-down list as a command string.

20. Click **[OK]** to finish the definition.

21. Create a loop after opening but before closing the input file. This will loop over the file until the file has been read completely.

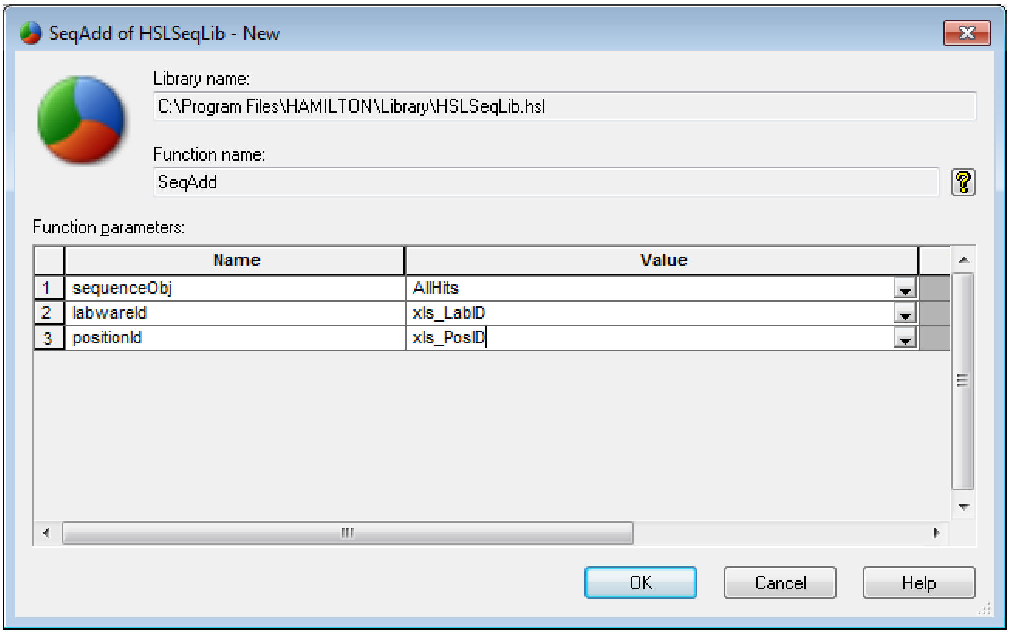


22. Within the loop, the first record of the file is read.

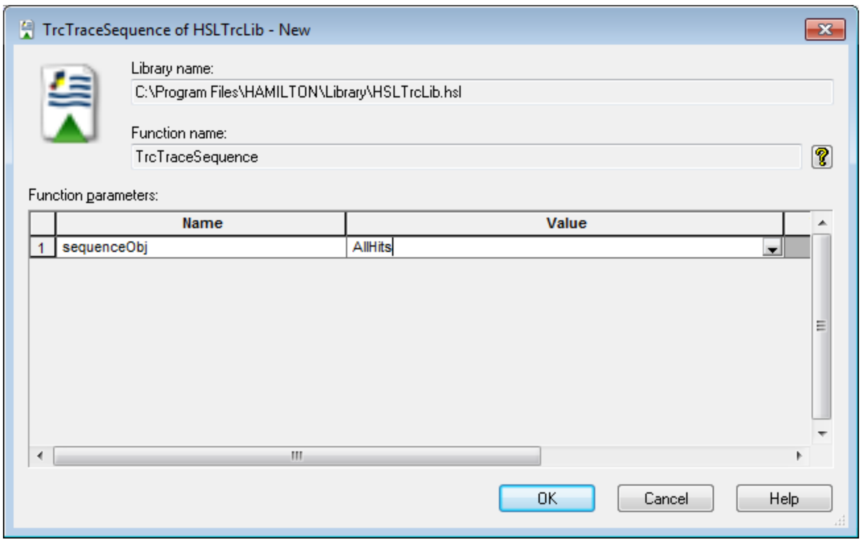


23. As the file is being read, the variables defined in "**Open File**" will be filled with the information found.

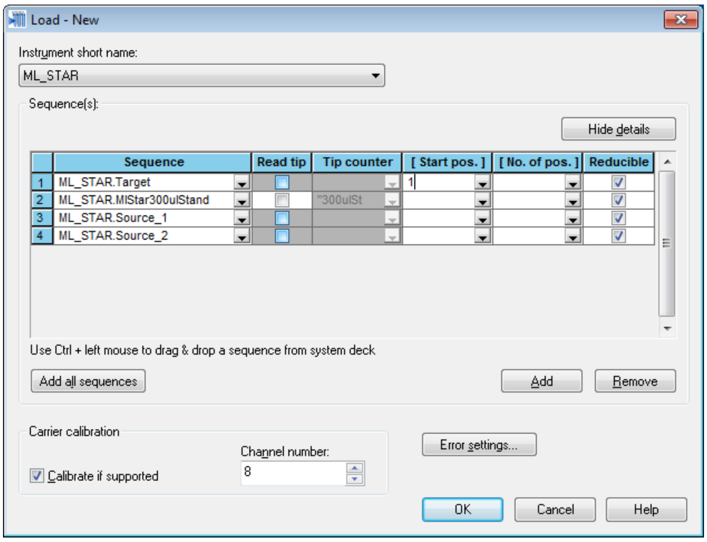
24. Use the “**SeqAdd**” Library Function. This command will add positions to a sequence when giving the LabID/PosID information (see image below).

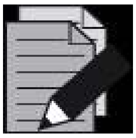


25. In the sequence “**Name**” Field, insert a sequence name. Since in this example, a completely new sequence is being created for all the hits “**AllHits**”, a name can directly be typed in. The system will then create a new sequence with the specified name.
26. Because the SQL selection string automatically skips all records with absorption values less than or equal to the value of the “**Threshold**” Variable, no other selection statements such as IF/ELSE are required.
27. In the “**LabwareId** “/ “**PositionId**”, fill-in the two variables from the open file.
28. After the loop, use the “**TrcTraceSequence**” Command to see which positions have been added to the “**AllHits**” Sequence. This command will list all sequence positions in the trace window.



29. Add a "**Load**" Smart Step. After the file has been analyzed and the hit sequence has been generated, the loading of the sequences (Carriers) onto the deck may start.



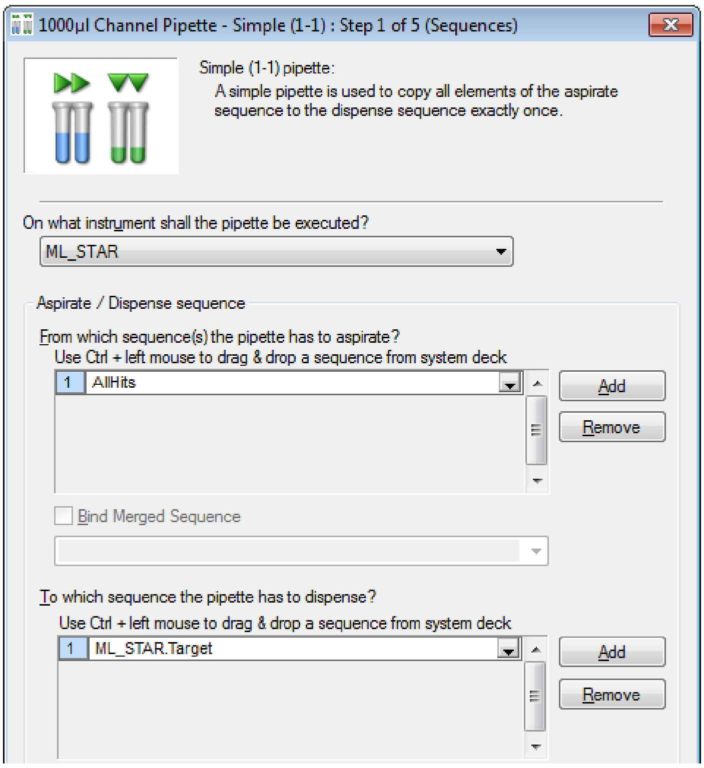


NOTE

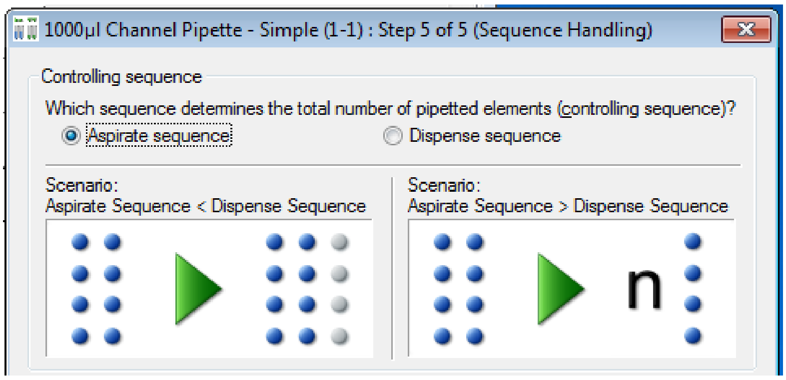
*The "**AllHits**" Sequence was created in the previous step and the current position of the sequence is undefined.*

To set, use the SetCurrentPosition command from the General Steps or in the load step, set the start Position to 1.

30. Add the "**1000µl Channel Pipette - Simple**" Smart Step into the method to start pipetting.



31. In this step of the wizard, make sure that the control sequence is properly defined (see image below). The “**AllHits**” Aspiration Sequence is the controlling sequence and therefore is not reloadable. The target sequence is not reloadable either.



32. The rest of the settings made are very similar to the previous example “**OnePlateToPipette**”.
33. To end with, add a step for unloading.

12.7 Method for Reformatting using the CO-RE 96 Probe Head

This method copies four 96-well plates into a 384-well plate using the CO-RE 96 Probe Head.

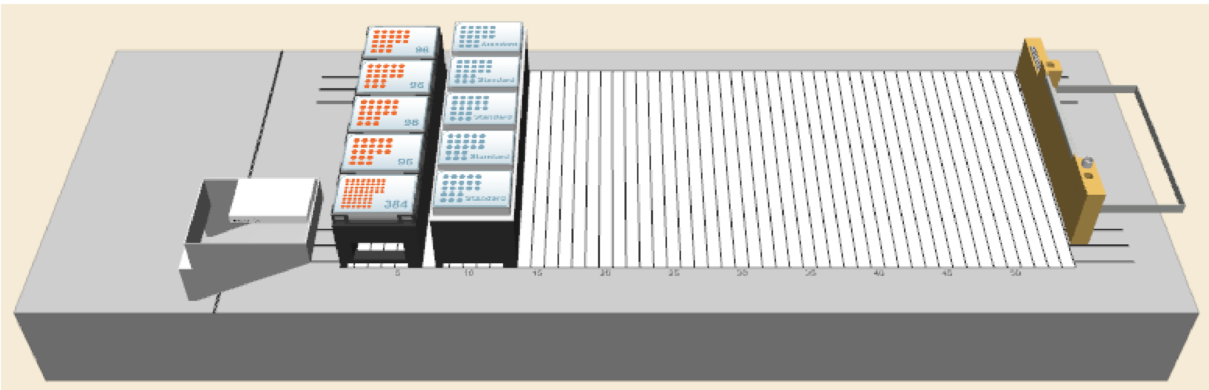
First, make sure that the CO-RE 96 Probe Head is activated in the Hamilton System Configuration Editor.

Now, it is possible to create a new method named “**Demo96To384.med**”.

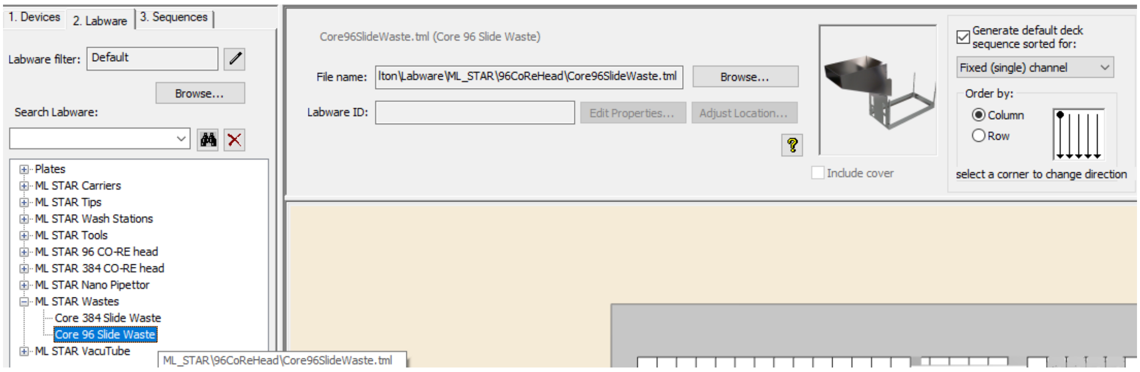
Creating the Deck Layout:

To create this Deck Layout, perform the following steps:

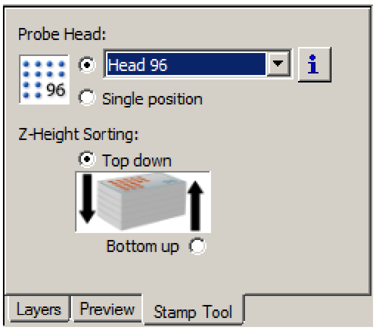
1. Start the Method Editor.
2. Select “**New → Method**” from the File Menu in the Method Editor. A prompt will then ask for a name to be assigned to the new method.
3. Enter a name, e.g. “**Demo96To384.med**” and click the **[Save]** Button. A new method window and a new “**System Deck**” Window are displayed, both empty.

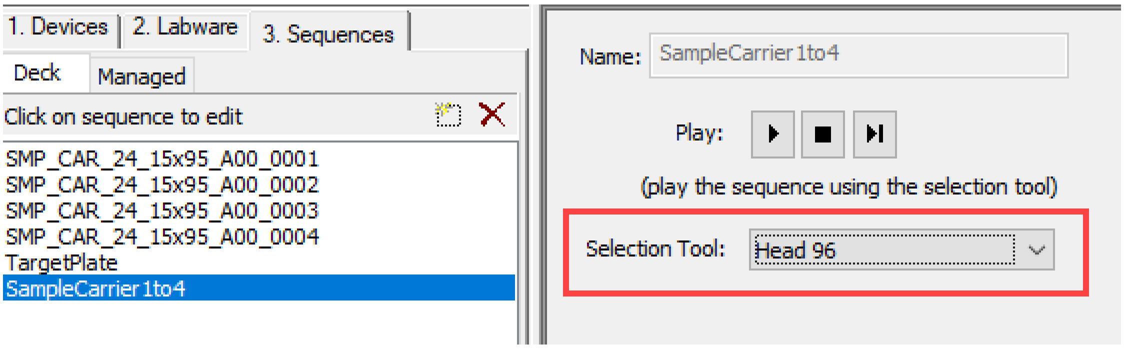


4. Click the “**Labware**” Tab above the System Deck View. A list of labware is displayed on the left-hand side.



5. Type “**Slide**” in the search labware field. Drag the “**Core96SlideWaste**” into the lower-left edge of the instrument deck. A frame will indicate the position where to release.
6. Type “**L5 MD**” in the “**Search Labware**” Field. “Drag-and-Drop” a “**PLT_CAR_L5MD_A00**” onto the deck.
7. Set the Stamp Tool to “**Head 96**”.





8. Type ‘Nunc’ in the “**Search Labware**” field. “Drag and Drop” four “**Nunc 96 Fl lb (low border)**” plates to position 1-4 of the Plate Carrier.
9. On the Plate Carrier’s position 5, place the plate “**Nunc_384_Sq.rck**”.
10. Type “**ST 48**” in the “**Search Labware**” field. Add the “**TIP_CAR_480_ST_A00**”.
11. Save the Deck Layout.

ATTENTION

Be aware that the CO-RE 96 Probe Head may require more space to pick up tips than the single pipetting channels.

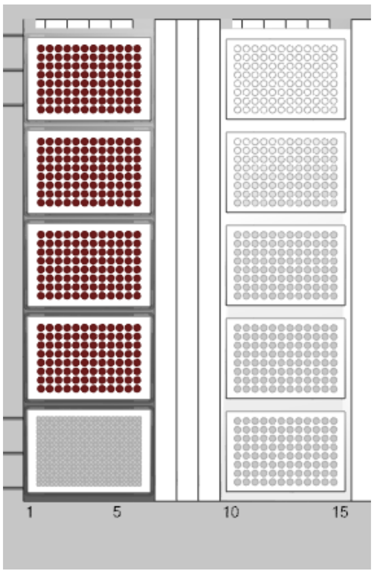
Creating the Sequences

Now, define the necessary sequence to be used in this method. To define a sequence, follow these:

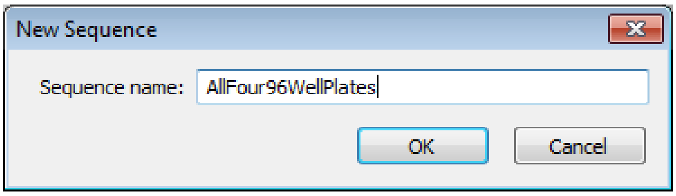
1. Click the "**Sequences**" Tab in the Method Editor. In the upper section, the "**Sequence Editor**" Screen is displayed.

2. Zoom-in by using the + zoom from zoom in/out toolbar , the "**View**" Menu in the Method Editor or by using the scroll wheel of the mouse.

3. Use the rubber band function over all four 96-well plates. All wells of the four selected plates are highlighted, as seen in the image below.



4. Save this sequence as "**AllFour96WellPlates**".

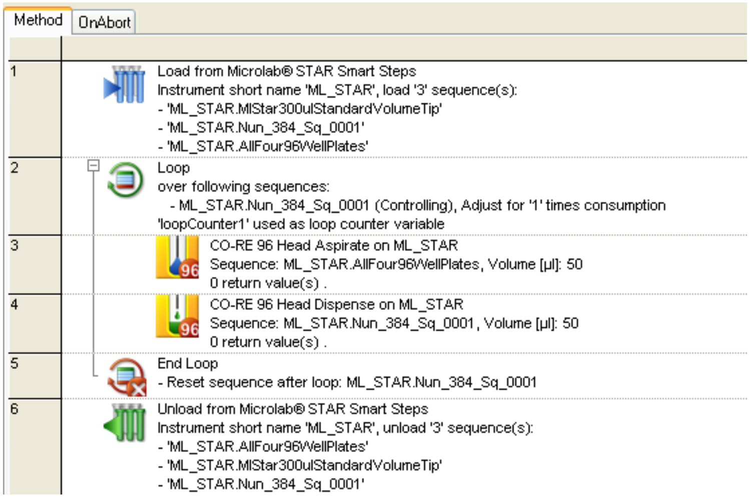


5. Save the new sequence by clicking **[OK]**. All the necessary sequences for the method have been created.

6. Click on the "**Method Editor**" Icon to activate the Method Editor.

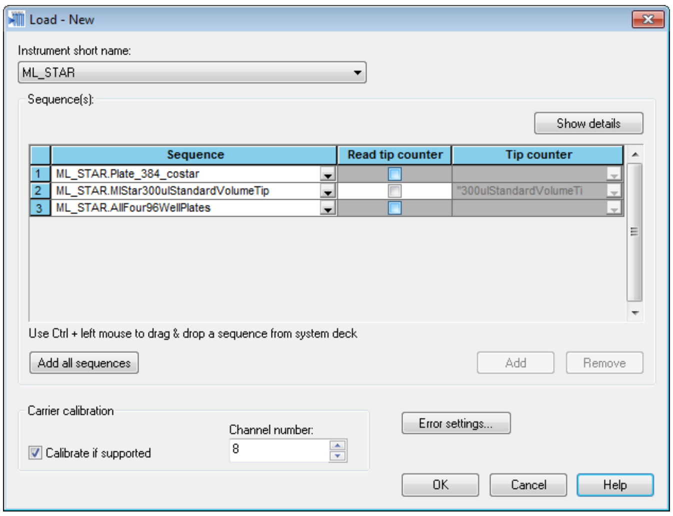
Creating the Method

The next step is to write the method in the Method Editor. Add all steps by dragging icons from the toolbox and dropping them into the method window. Finally, the method should look as displayed below:



In more detail, the necessary steps for the method are the following:

1. Add the "**Load**" Smart Step by dragging-and-dropping it from the toolbar into the method window: the "**Load**" Dialog appears as shown below.

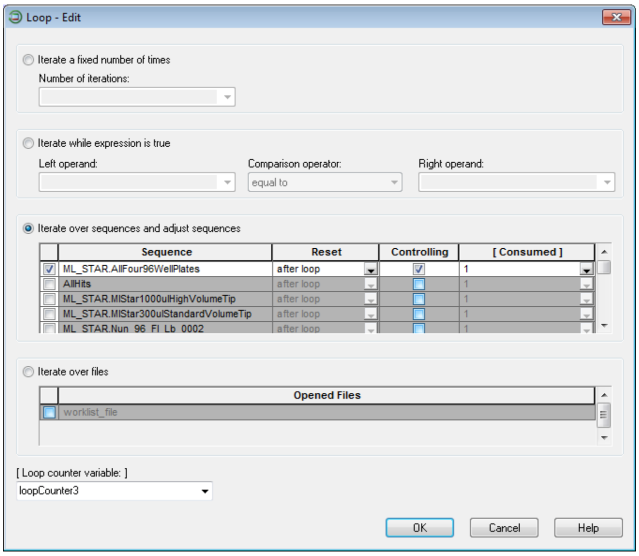


2. Click on the drop-down list in the "**Sequence**" Section and select the "**ML_STAR.AllFour96WellPlates**" Sequence.
3. Use the **[Add]** Button to create two more lines. Add the "**ML_STAR.Nun_384_Sq_0001**" sequence and the "**ML_STAR.MlStar300ulStandardVolumeTip**" Tip Sequence.
4. Finish the "**Load**" Step with **[OK]**.

NOTE

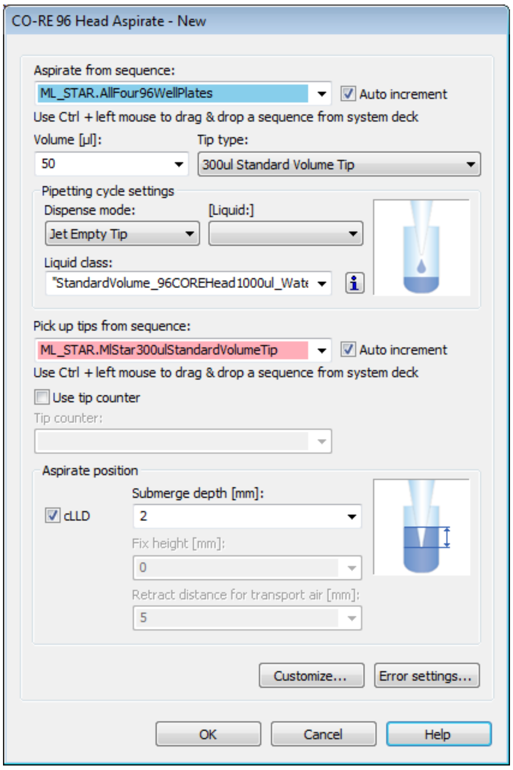
*The sequences to load can be selected in the Sequence View and then dragged-and-dropped into the list fields in the "**Load**" Dialog (Ctrl + left mouse drag-and-drop). The two windows (Deck Layout View and "**Load**" Dialog) are interactive: when a sequence in the "**Load**" Dialog is selected, this sequence flashes in a specific color in the Deck Layout View.*

5. Insert a "**Loop**" from the "**General Steps**".
6. Activate the **[Iterate over sequences and adjust sequences]** Radio Button and tick the box for the "**ML_STAR.AllFour96WellPlates**".



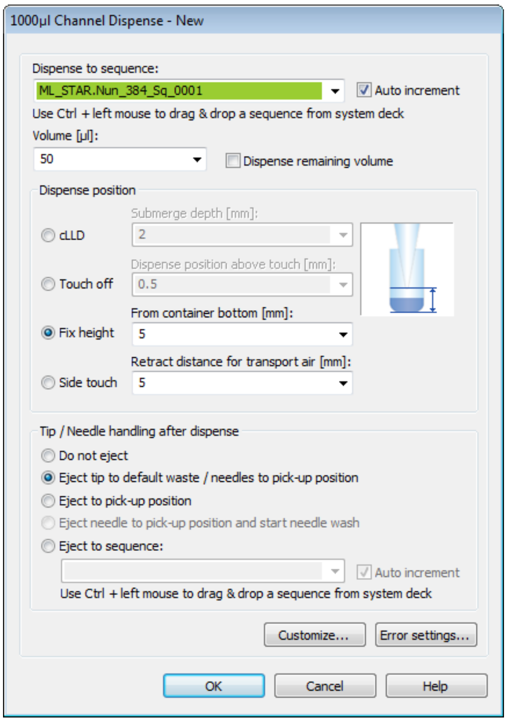
7. Click **[OK]** to close the "**Loop**" Step.

8. Add a “**CO-RE 96 Head Aspirate**” Step to the loop. Fill out the input fields as follows:



- Aspirate sequence: “**ML_STAR.AllFour96WellPlates**”
- Volume: 50ul
- Liquid: “**DMSO**”, the Dispense mode is “**Jet Empty Tip**”
- Liquid class: “**StandardVolume_96COREHead1000ul_DMSO_DispenseJetEmpty**”
- Tip pickup sequence: “**ML_STAR.MlStar300µlStandardVolumeTip**”

9. Click **[OK]** to close and save the aspirate step.
10. Add a "**CO-RE 96 Head Dispense**" Step after the aspirate step. Fill-in the input fields as follows:



- Dispense sequence: "**ML_STAR.Nun_384_Sq_0001**"
- Volume: 50ul
- Dispense position: "**Fix height**", 5mm from container bottom
- Tip handling after dispense: "**Eject tip to default waste**"

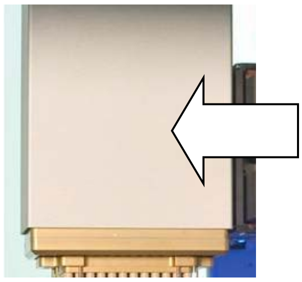
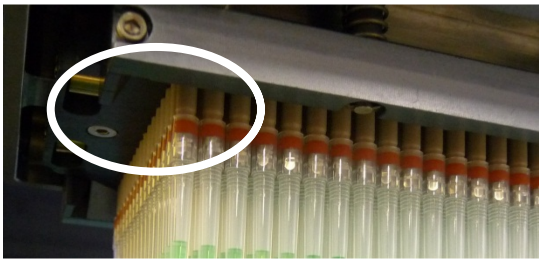
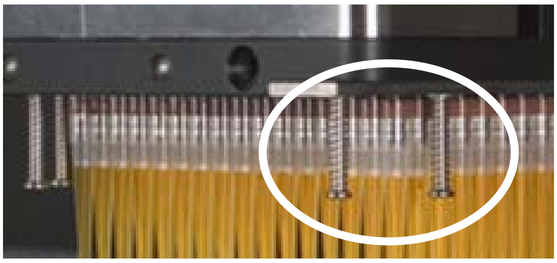
11. Then add the "**Unload**" Smart Step into the method.

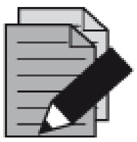
12.8 Method using the CO-RE 384 Probe Head

12.8.1 Differentiate between CO-RE 384 Head and CO-RE 384 STP Head

First, check if a CO-RE 384 Head or a CO-RE 384 STP Head (Shifted Tip Pickup) is installed on the instrument.

There is an easy way to distinguish the CO-RE 384 STP Head from the CO-RE 384 Head. Be guided according to the table shown below.

<table>
<tr><td>With 'window' = 384 STP Probe Head</td><td>Without 'window' = 384 Probe Head</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Without screws = 384 STP Probe Head</td><td>With screws = 384 Probe Head</td></tr>
<tr><td></td><td></td></tr>
<tr><td>+ Full 384 tip pickup possible</td><td>+ Full 384 tip pickup possible</td></tr>
<tr><td>+ Column wise tip pickup possible</td><td>+/-Column wise tip pickup ONLY with tip lifter</td></tr>
<tr><td>+ Row wise tip pickup possible</td><td>- Row wise tip pickup NOT possible</td></tr>
<tr><td>+ Quarter tip pickup possible</td><td>- Quarter tip pickup NOT possible</td></tr>
<tr><td>+ Single tip pick up (edge) possible</td><td>- Single tip pick up (edge) NOT possible</td></tr>
<tr><td>Definition in system configuration editor:
<table>
<tr><td>5ml Channel: raster</td><td>18mm</td></tr>
<tr><td>Autoload</td><td>Available</td></tr>
<tr><td>Camera Channel</td><td>Not available</td></tr>
<tr><td>CO-RE 384 Head</td><td>Head 384 STP</td></tr>
<tr><td>CO-RE 96 Head</td><td>Available</td></tr>
<tr><td>iSWAP</td><td>Small gripper</td></tr>
</table></td><td>Definition in system configuration editor:
<table>
<tr><td>5ml Channel: raster</td><td>18mm</td></tr>
<tr><td>Autoload</td><td>Available</td></tr>
<tr><td>Camera Channel</td><td>Not available</td></tr>
<tr><td>CO-RE 384 Head</td><td>Head 384</td></tr>
<tr><td>CO-RE 96 Head</td><td>Available</td></tr>
<tr><td>iSWAP</td><td>Small gripper</td></tr>
</table></td></tr>
</table>



NOTE

Use the table above to make sure the correct head is selected in the System Configuration Editor.

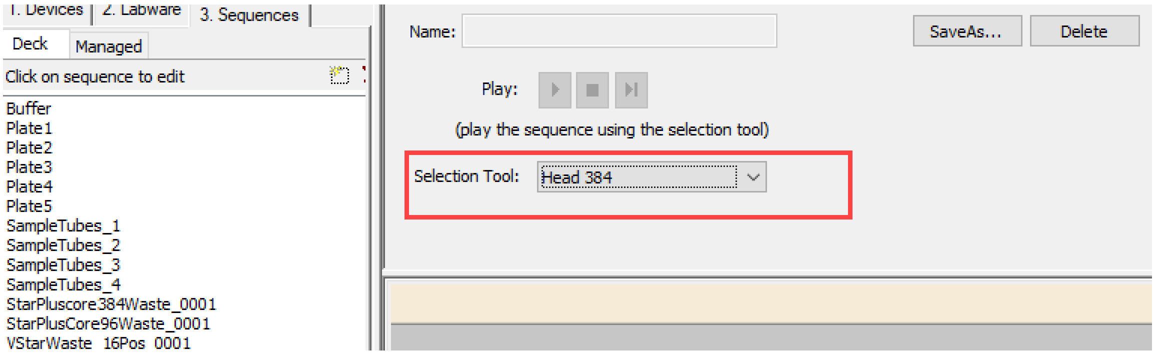
12.8.2 Programming a Column Serial Dilution with CO-RE 384 STP Probe Head

This section describes how to program a column-wise serial dilution using the CO-RE 384 STP. The samples are in column 1 of the plate. The 384 Head pipettes diluent from the diluent container and transfers the mix to column 2. This is repeated until column 24, so there is a decreasing concentration of sample in every column.

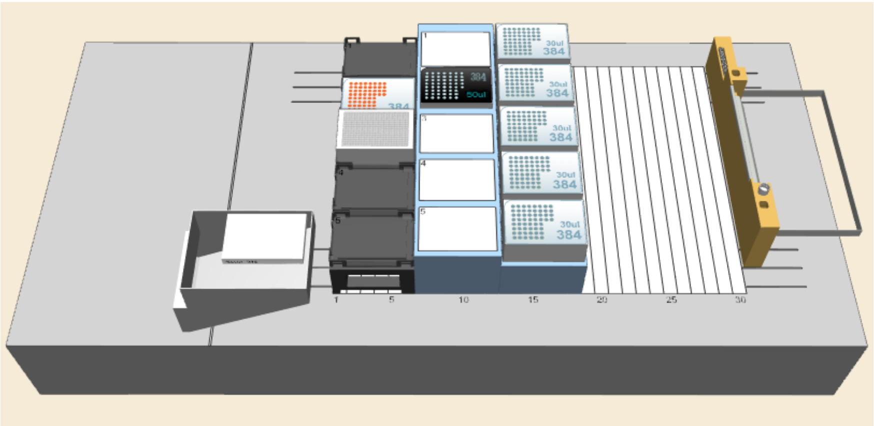
Close all open windows and create a method called "**Demo384SerialDilution.med**".

Creating the Deck Layout

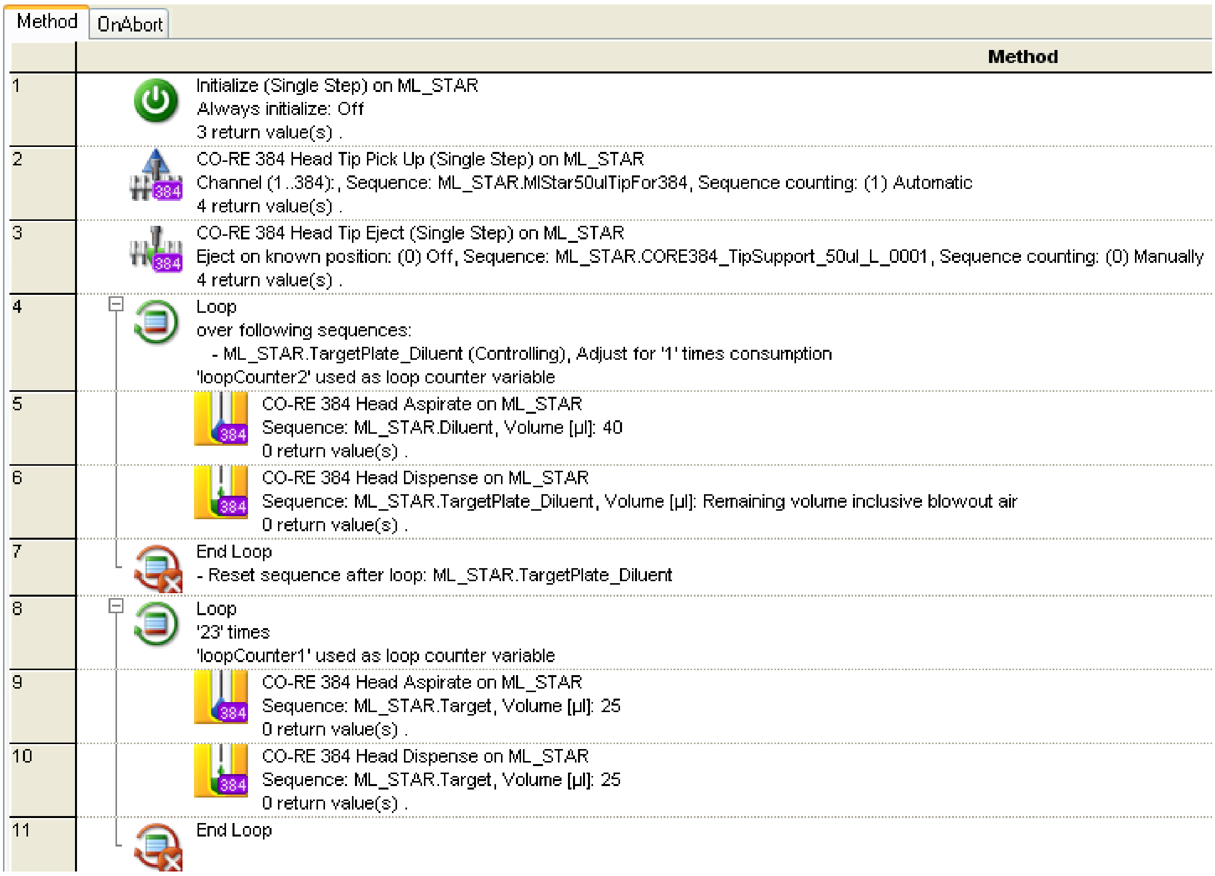
1. From the "**Sequence**" Tab, select the "**Head 384 Column**" Stamp Tool.



2. Activate the "**Labware**" Tab and add the "**Core384SlideWaste**" onto the deck.
3. Add a "**TIP384_CAR_1920_50ul**" Carrier to the deck.
4. Add a "**TIP_CAR_480_A00**" to the deck.
5. On the empty Tip Carrier, place a "**TipSupport_50ul_384_STP_Head**" in position 2.
6. Add a "**PLT_CAR_L5AC_A00**" Carrier to the deck.
7. On the Plate Carrier, add a "**Nun_300ml_384C_Rgt_L.rck**" in position 3. With a right-mouse click, enter the "**Properties…**" and change the "**LabwareID**" to "**Diluent**".
8. On the Plate Carrier, add a "**Nun_384_Sq.rck**" in position 2. Change the "**LabwareID**" to "**Target**".
9. The Deck Layout should look as shown below.



10. The final method should look like this:



Step 1 Is the initialization of the instrument.

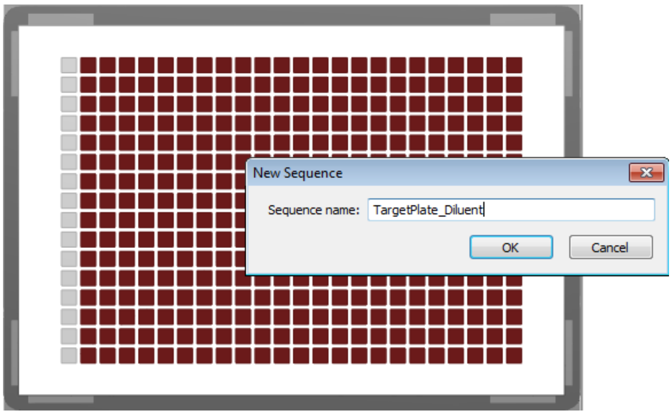
Steps 2 and 3: Bring the tips from the tip rack to the tip support for column-wise tip pick up.

Steps 4 to 7: Pipette the diluent over the full plate except column 1 (where the sample is).

Steps 8 to 11: Aspirate from the previous column and mix with the following column.

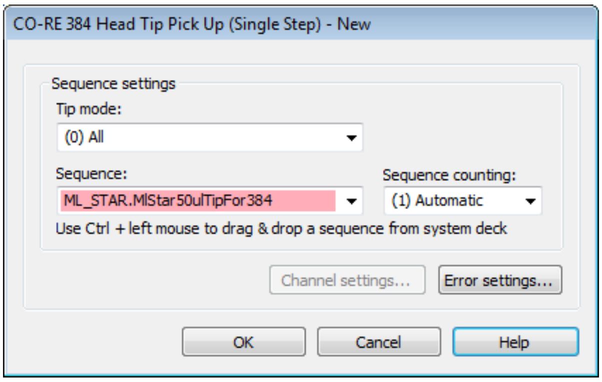
Creating the Sequences

Only one additional sequence is used in this method. Make sure the Stamp Tool is still set to "**Head 384 Column**". Create a sequence over the target plate WITHOUT selecting column one. Name it "**TargetPlate_Diluent**".

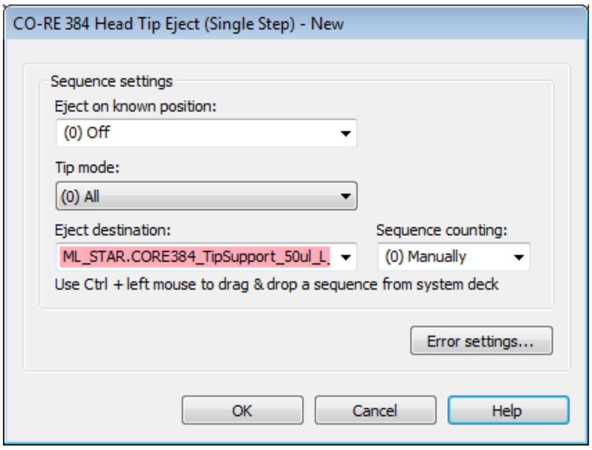


Creating the Method

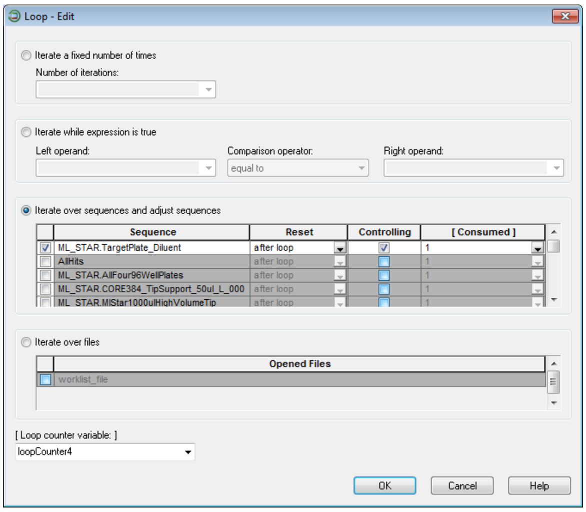
1. First, insert an "**Initialize**" Step, then use the "**CO-RE 384 Head Tip Pick Up (Single Step)**" to pick up the tips from the tip rack.



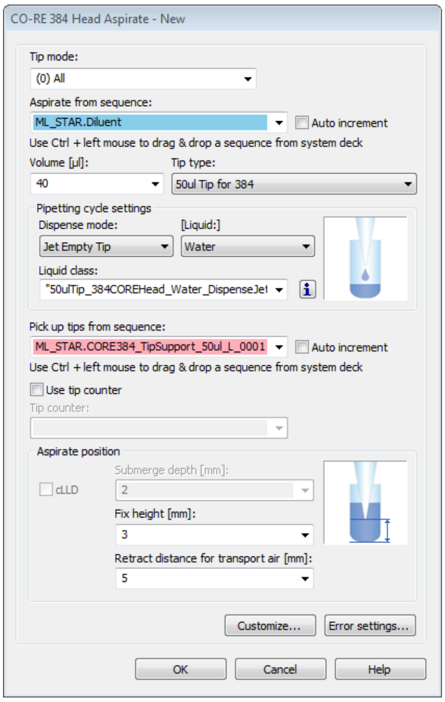
2. Move the tips to the TipSupport using the "**CO-RE 384 Head Tip Pick Eject (Single Step)**". Set the input field to "**(0) OFF**" because the tips do not go to the default waste. For "**Destination**", select the tip support sequence.



3. The next step is a "**Loop**" Step to make sure the distribution of diluent continues over the full "**TargetPlate_Diluent**" Sequence (all columns except column 1). Use the option "**Iterate over sequence and adjust sequence**" and select "**ML_STAR.TargetPlate_Diluent**".

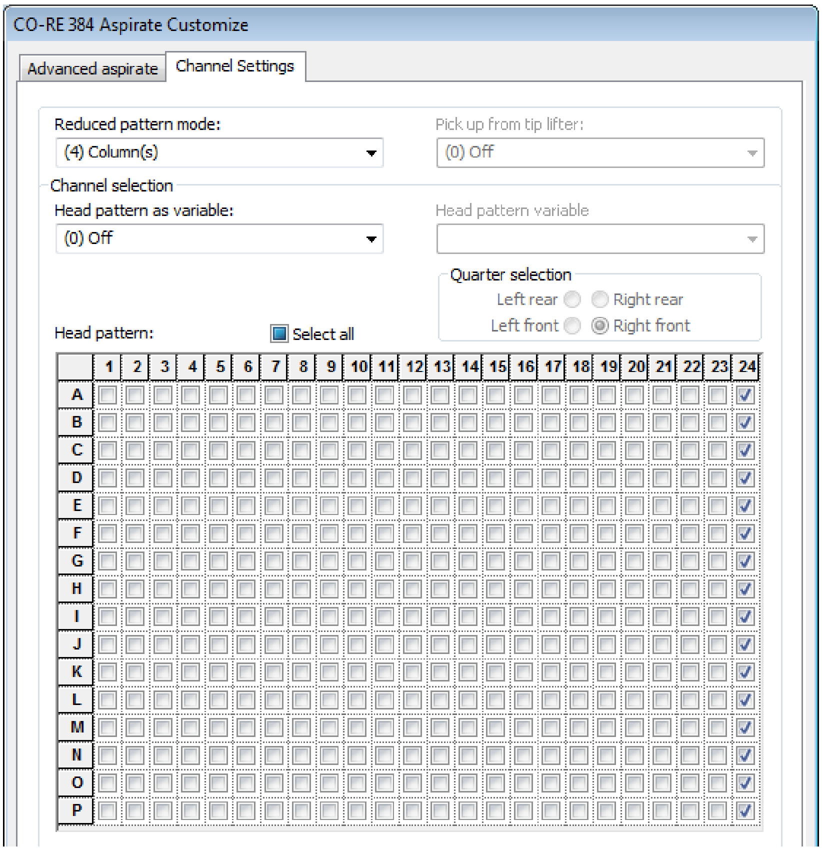


4. The loop is followed by the aspiration of diluent (with only one column of tips). Use the "**CO-RE 384 Head Aspirate**" Step and use the following settings:



Tip Mode:	(0) All the column wise setting is made in the "**Customize**" later
Aspiration sequence:	ML_STAR.Diluent
Auto increment:	Not ticked
Volume:	40ul
TipType:	50 µL tip for 384

5. The liquid class is selected automatically after specifying the liquid "**Water**".
6. The tip pickup sequence should be "**ML_STAR.CORE384_TipSupport_50ul_L_0001**".
7. Now, the column-wise tip pickup must be specified in the "**Customize**" Tab, found in the "**Channel Settings**" Section.
8. Make sure the "**Reduced pattern mode**" is set to "**(4) Column(s)**". The column 24 is then automatically activated to be used for the tip pick up.



9. Other possible selections for "**Reduced pattern mode**" are:

   "**(0) All**" to pick up all tips

   "**(1) One Channel**" to pick up one tip in the corner of the head

   "**(2) Quarter**" to have ¼ of the tips picked up. Quarter selection has different rears as shown below.

   "**(3) Row(s)**" for row-wise tip pickup and pipetting

ATTENTION

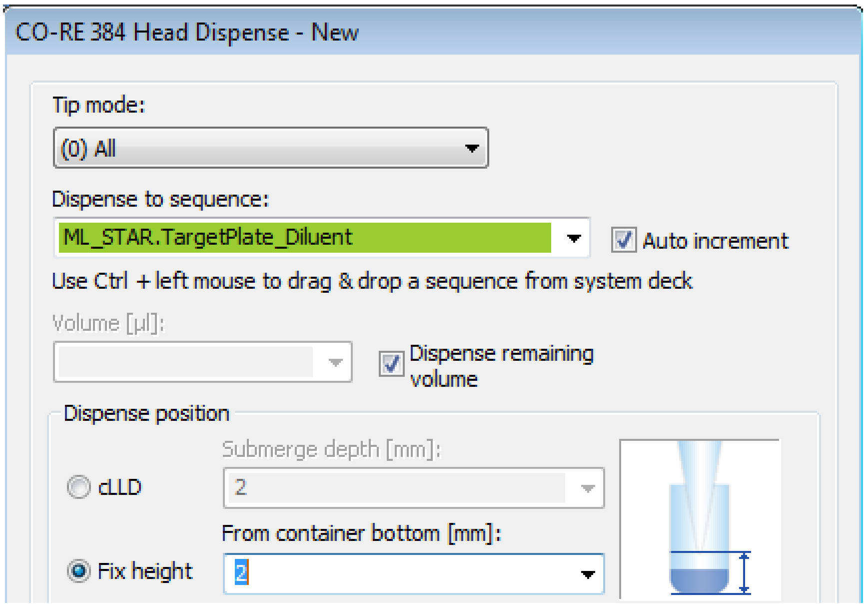
This is the pattern of the CO-RE 384 Probe Head and NOT the pipetting pattern on the plate.

10. To pick up two columns of tips, activate column 23 by clicking on the column number 23 as well. Through this, several columns can be activated and will be picked up.

NOTE

It is not possible to have gaps (empty lines) in the pattern. To create such a pattern, use a second tip support and move the tips to the according columns, rows, quarters, or single positions.

11. If a reduced pattern mode other than "**(0) All**" is used, please read about the risks and limitations in the Help of the step → <u>CO-RE 384 Head Pattern Mode Limitations.</u>
12. It is also possible to use a variable for the columns or rows to be used. Example for column variable: "**000000000000000000000011**", where every character represents a column.
13. "**0**" (zero) stands for not used column, "**1**" (one) for used column. In the example, the columns 23 and 24 will be used for tip pick up. For rows, only 16 values must be passed.
14. After the aspiration, insert a "**CO-RE 384 Head Dispense**" Step and fill the input fields as follows:

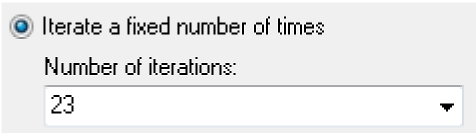


15. Make sure that the "**Auto increment**" Box is ticked.
16. The volume should be as big as the aspiration volume, or simply activate the "**Dispense remaining volume**" Checkbox.

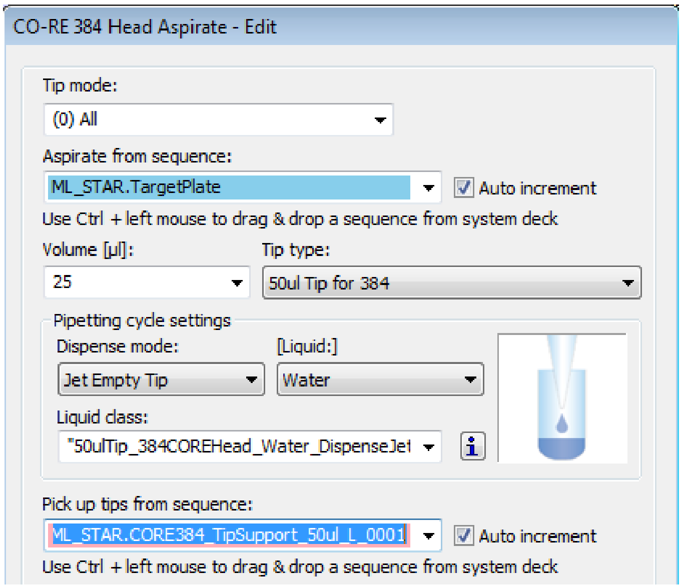
The tip pattern is taken from the aspiration step, so no changes are necessary here.

17. With this step, the distribution of the diluent over the plate is finished. There is a choice whether to throw away the tips or to keep them. Use the 'Tip handling after dispense' to apply the settings.

18. Now, a second loop is needed to perform the transfer from column to column. With 24 columns, 23 transfers are needed. Therefore, the second loop can iterate over the fixed number of times.

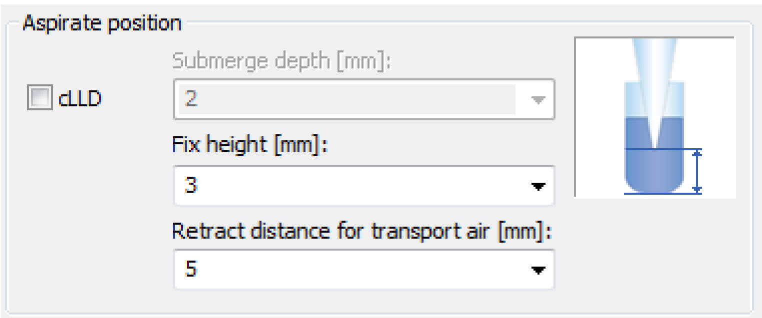


19. Inside the second loop, the aspirate of the sample is executed. To do so, use a “**CO-RE 384 Head Aspirate**” Step with the settings shown below. If the step from line 5 of the method is copied, there is no need to repeat the tip pattern settings in the “**Customize**” Tab (to use only one column).

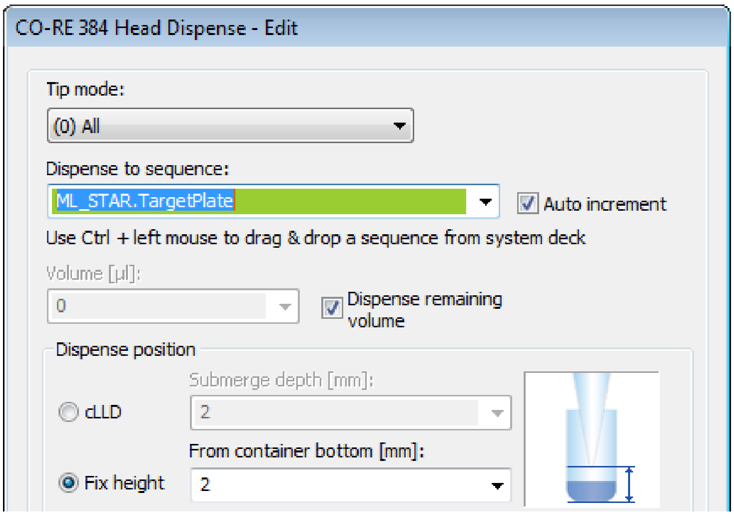


20. It is important that the “**Auto increment**” Box is ticked. This will ensure that the aspirated liquid is transported to the next column for dispensing.

21. In the lower section, use a fixed pipetting height of 3 mm. Keep in mind that this height may differ if other plates are used.



22. The dispense step may now be programmed. Add (or copy from above) the "**CO-RE 384 Head Dispense**", using the settings below:



23. The target plate is used both as aspirate and dispense location to get the serial dilution. Since the "**Auto increment**" on the Aspirate step was checked, the dispense step always uses the next column and because the next aspiration should happen in the same column, the "**Auto increment**" Checkbox must be unchecked on the dispense step.
24. Again, set a fixed height for dispensing and then switch to the "**Customize**" Section. The "**Minimize z-move after step**" can be activated since dispensing is performed on one plate only. This will prevent the head from moving up to traverse height, which saves time.
25. Activate the mixing function when dispensing the diluent into the sample. Use the "**Customize**" Tab to enter the "**Advanced Dispense**" Section.
26. Enter the values for the mixing as follows:
    - Cycles = 4
    - Mix position = 2
    - Volume = 20
27. Close the step by clicking **[OK]** twice.

12.9 Create a Sub-Method

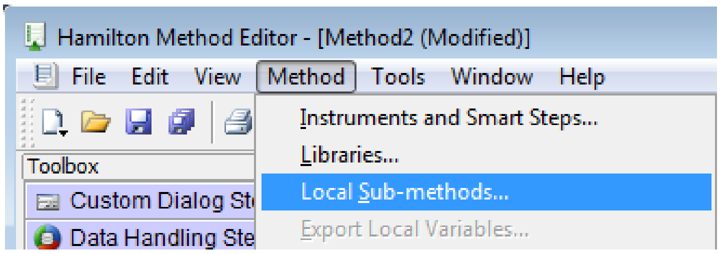
The following example demonstrates how to:

- Build sub-methods into the method.
- Activate a sub-method library for a particular function

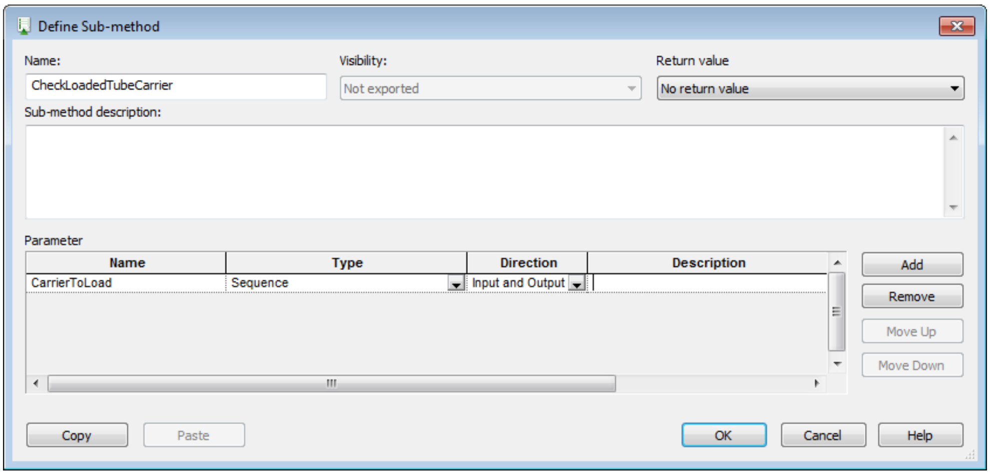
This example sub-method will be performing the pipetting from a source to a target plate.

Since sub-methods are created in the ‘main’ – method, it is not necessary to define a Deck Layout for the sub-method(s).

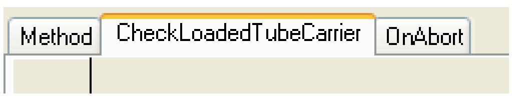
1. To create a local sub-method, right-click on the “**Method**” in the Method Window and select “**Local Sub-methods**”. Name the sub-method “**CheckLoadedTubeCarrier**”.



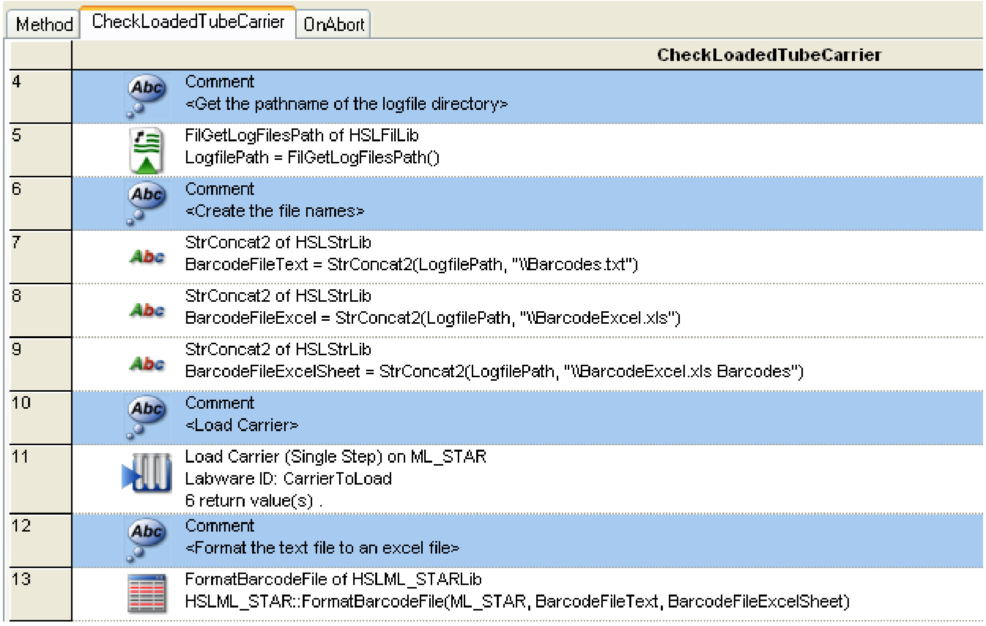
2. The parameter window shown below appears. Supply the following parameters with the following values:
   - Sub-method name: “**CheckLoadedTubeCarrier**”
   - Description: optional
   - Parameters: “**CarrierToLoad” of type** “Sequence”
   - Sub-method name: “**Pipetting**”



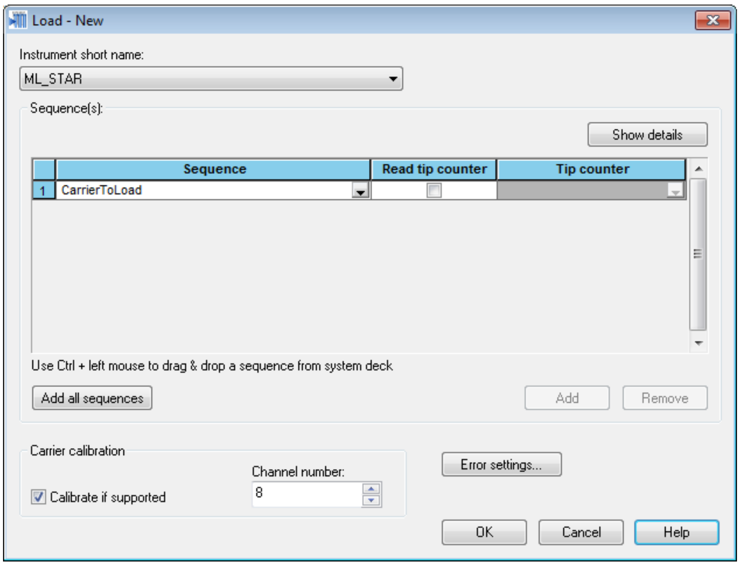
3. The empty sub-method will then appear.



4. Create the sub-method by "Dragging-and-Dropping" Steps from the toolbox to the Step Editor.

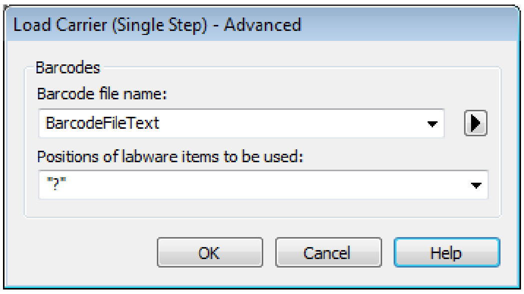


5. The pathname settings for the unformatted text file (step 7 of the image above), the formatted Excel file (Step 8 of the image above) and the sheet name of the Excel file (Step 9 of the image above) are all set using the "**Assignment**" Command from the General Steps.
6. Then the new Carrier is then loaded.

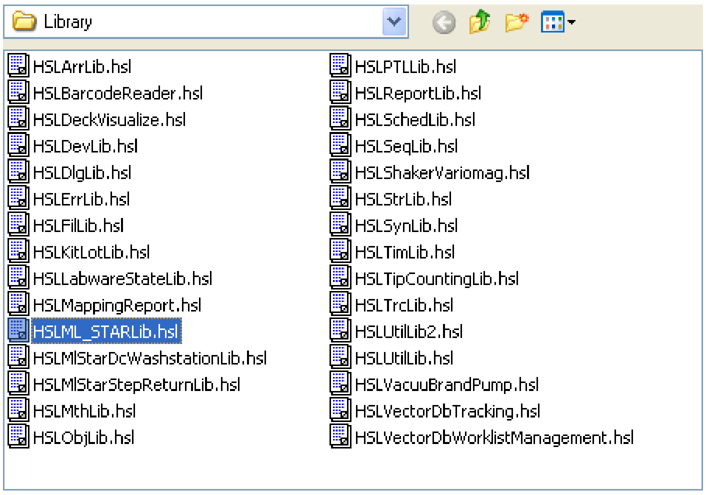


7. Use the "**CarrierToLoad**" Variable from the parameter window as the Carrier name.

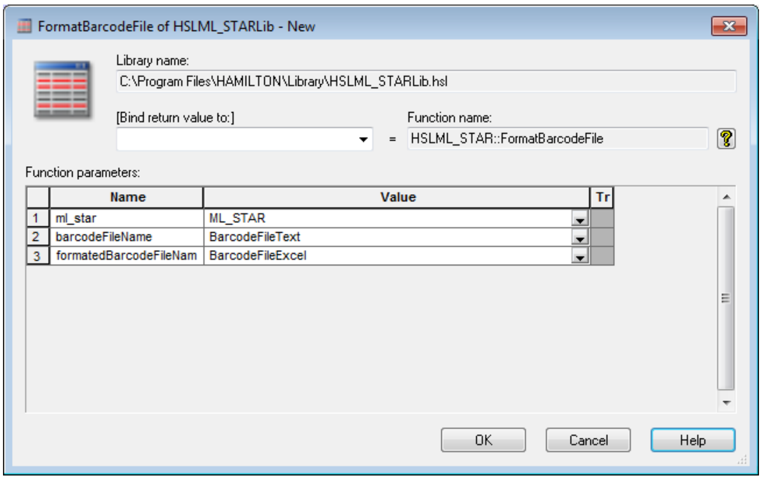
8. In the "**Advanced settings**", set the barcode file name to the variable "**BarcodeFileText**" that has been defined in the previous step.



9. The "**Load**" Step creates a barcode file in TXT format under the specified pathname (step 7). The file will be formatted (Step 13) after the Carrier has been loaded. To use this step, its library must be included first. To do this, follow the steps below:
10. Add the library "**HSLML_STARlib.hsl**" to be able to use the "**FormatBarcodeFile**" Command.
11. Also add the "**HSLTrcLib.hsl**" to be able to trace data.



12. Fill out the "**FormatBarcodeFile**" Dialog: the instrument and the two files must be used as parameters. A return value is not needed.

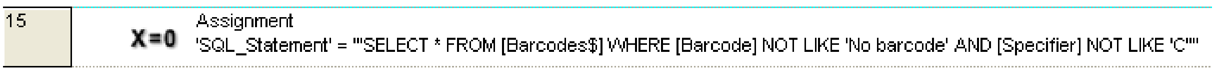


13. Note that the "**BarcodeFileExcel**" Variable must be handed over.
14. The next step is to open the formatted file. Because only the tube barcodes are of interest, the information in the file has to be filtered. SQL (Structured Query Language) offers a lot of these filter and sorting options. For more explanation, open the Help in the "**FileOpen**" Step.

Example

A filter string has to be defined. This is done in step 15, as shown in the snapshot below.

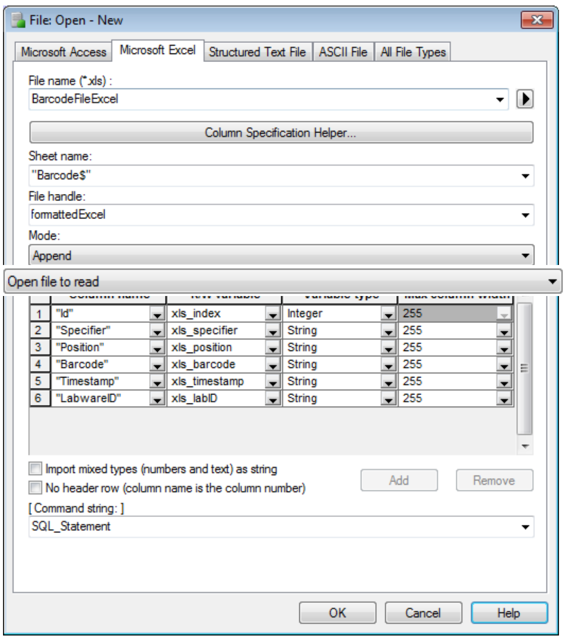
"**SELECT * FROM [Sheet1$] WHERE [Barcode] NOT LIKE 'No barcode' AND [Specifier] NOT LIKE 'C'**"



This means that the Excel file is opened using two filters:

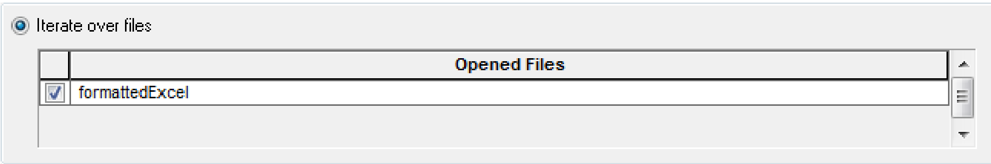
- All "**No barcode**" Entries (which is the code for an empty Carrier position) are not shown
- All Entries with "**C**" in the "**Specifier**" Column are not shown (this is the Carrier barcode)

1. In the "**File Open**", the name of the Excel file and all the columns have to be passed. For "**File name**", use the variable specified in the header. For "**Sheet name**", use "**Barcodes1$**". The $ sign is needed to be able to read from the file.
2. Set the "**File handle**" to "**formattedExcel**". Giving a name makes it easier to find and use (e.g. of several files).
3. Set the mode to "**File Open to Read**".
4. Then, type in all the columns that appear in the file to be opened. Note that the column name has to be enclosed in quotation marks. Link this name to a Read/Write variable and set its type. As seen below, all columns are of type String except the "**Id**". This one is an integer.
5. Finally, select the variable for the SQL Filter query in the "**Command string**" Field.

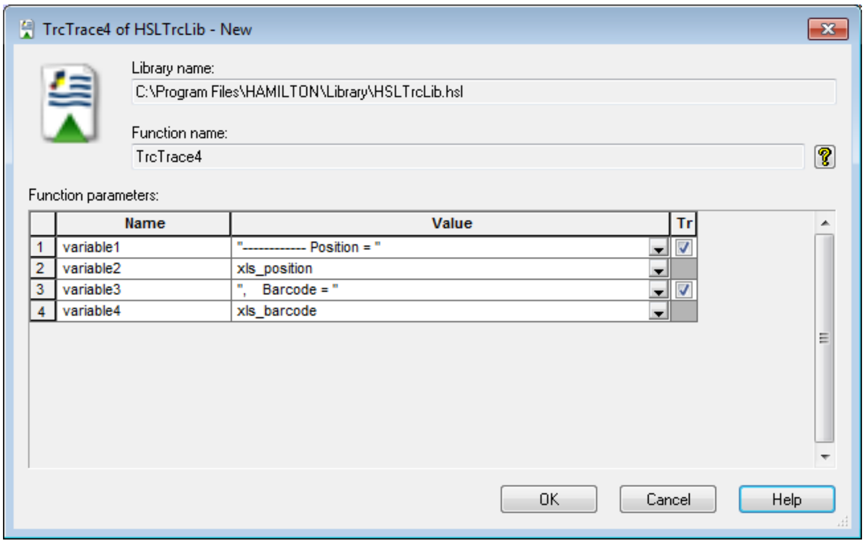


6. Click **[OK]** to close the "**File Open**" Step.

7. Add a loop and iterate over file1. This means that the loop will be executed until ”**formattedExcel**” ends.



8. In the loop, the data is read by a File read from the General Steps using the file handle **Excel file1** to include the information in the trace file (**C:\programm file\ HAMILTON\LogFiles**).

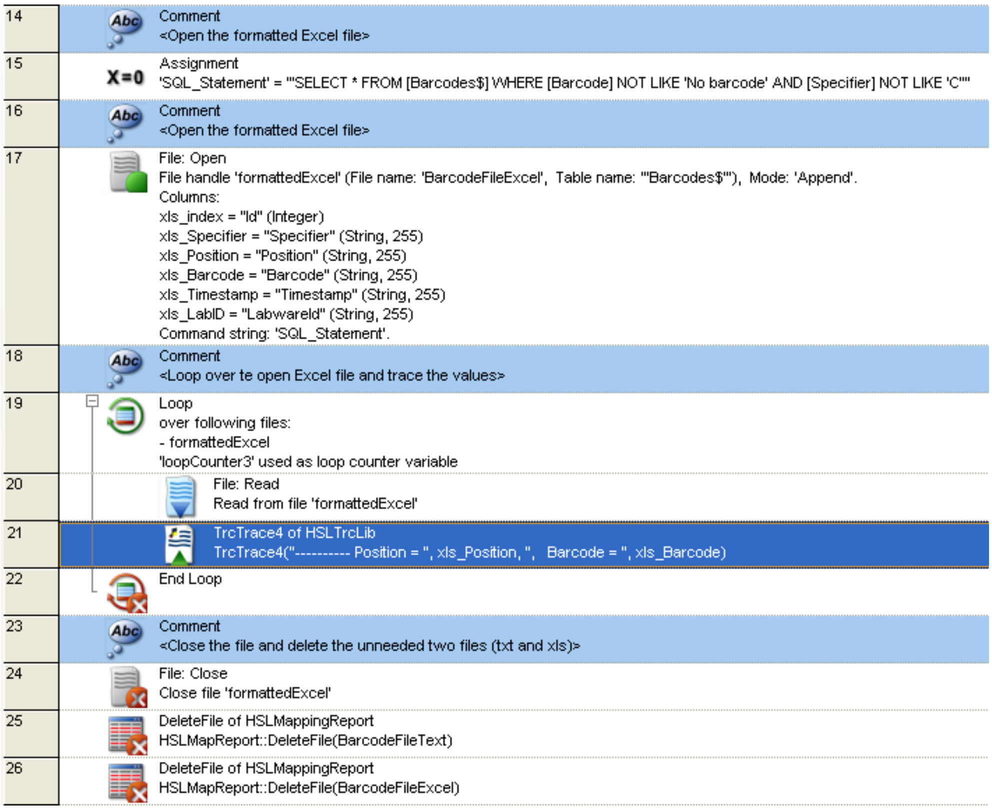


9. The Data in line 1 and 3 are in quotation marks. Therefore, they are displayed as text. Line 2 and 4 will show the variable’s value at run time. The result of this will be an entry for each valid barcode, as shown below.

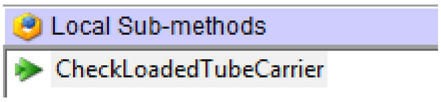
16:26:56> USER : Trace - complete; ---------- Position = 1, Barcode = Barcode01

10. This is done for each entry in the file that matches the restrictions in the SQL statement.

11. The result of the second part of the method should look like this:



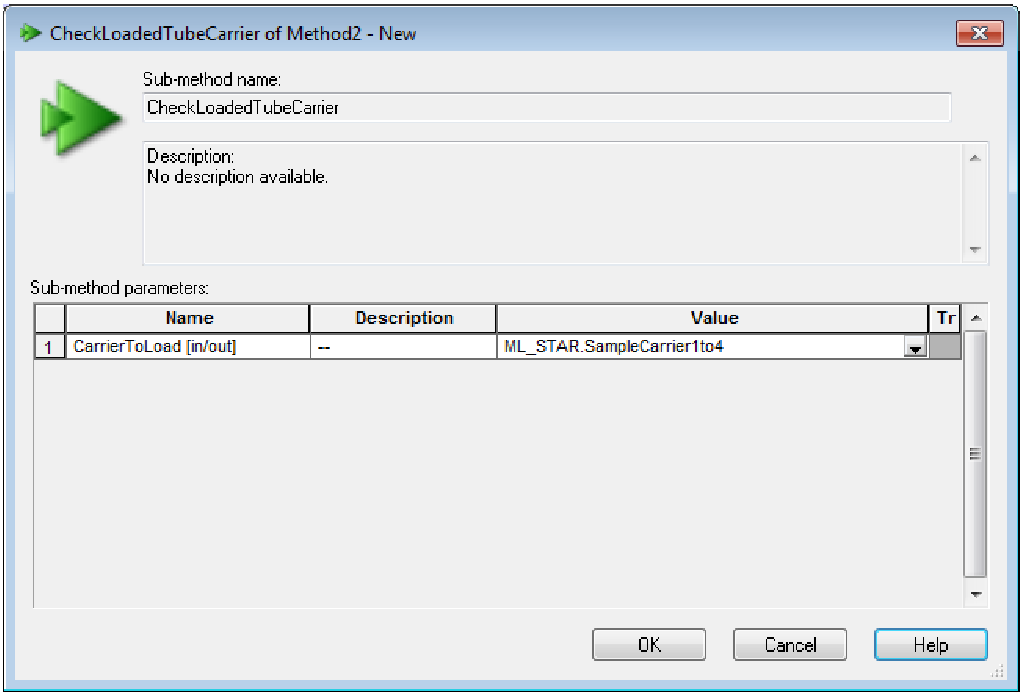
12. The file is closed in step 24.
13. Steps 25 and 26 will delete both the system created BarcodeTextFile and the formatted BarcodeExcelFile. To do this, use the DeleteFile command from the HSLMappingReport library.
14. The newly created local sub-method is now ready.



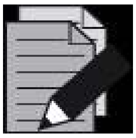
12.9.1 Using the Sub-Method in the Main Method

This sub-method can be used as the next step in the main method.

1. Select the "**Main Method**" Tab and drag the “**Sub-method**” Icon into the main method.
2. The following dialog opens, and the parameter list has to be supplied as shown on the next page.



3. The sub-method dialog cannot be closed. It will load the Carrier “**SMP_CAR_24_17x100_A00_0001**” onto the deck, read the barcodes and display all valid barcodes in the trace file.
4. The sub-method can be used for 24 and 32 tube carriers, MD and AC plate carriers etc. To do this, simply change the Carrier’s name in the sub-method call.



NOTE

The system will always show all possible barcodes in simulation mode.

13 Liquid Handling (Pipetting)

This section describes the process of pipetting liquids using the Microlab instruments.

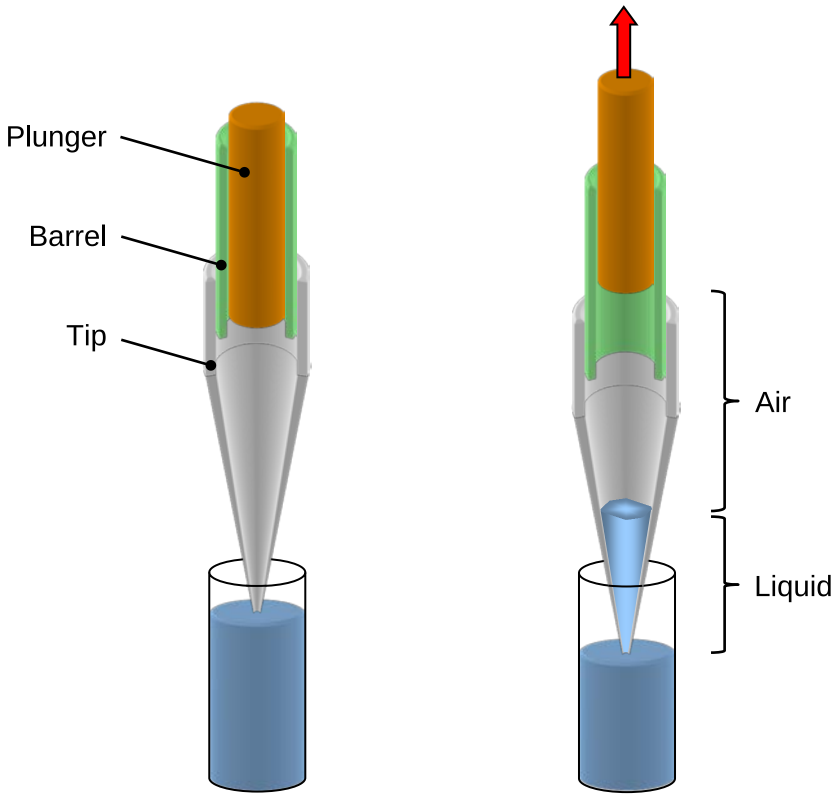
Definition

Pipetting means the transfer of (small or large) quantities of liquid from one container to another. A pipetting operation is achieved through the following:

- Tip or needle pickup
- Aspirating liquid from a source container
- Dispensing it into a target container
- Tip or needle eject

13.1 The Air Displacement Principle

The Microlab instrument is based on the air displacement pipetting principle, comparable to the work principle of handheld pipettes.



Air displacement means that the liquid is aspirated into and dispensed from a disposable tip or needle by the movement of a plunger. There is air between the plunger and the liquid surface. No system liquid of any kind is involved in the Microlab instrument. On the Microlab instrument, the pipetting of volumes from 0.5 µL up to 5 mL are processed. Filter tips can be used to avoid contamination by aerosol.

13.2 From Aspiration to Dispense

This section describes the processes involved in a simple pipetting step. These phases are:

- Movement to pick-up position
- Tip pick-up
- Movement to source position
- Aspiration
- Movement to target position
- Dispense
- Movement to drop-off position

  and
- Tip drop-off

Although they are also important, the movement phases are not described in detail here.

13.2.1 Tip Pick-Up

The first task of the Microlab instrument is to pick up a disposable tip or a reusable steel needle.

For disposable tips, special carriers (typically holding 5 tip racks of 96 tips each) are placed on the instrument deck.

Steel needles can be picked up directly from the wash station or from a separate needle rack.

The tip pick-up of the individual pipetting channels can happen simultaneously or one by one, as specified by the programmer.

13.2.2 Aspiration

Blow out air

The first step within an aspiration and dispense cycle is to aspirate a variable amount of "**blow-out**" air, which is used at the end of the (last) dispense, to blow all the liquid out of the tip. This is done with the tips still in the air.

Aspirate position

The most important condition for a successful aspiration is to make sure that the tip dips into the liquid. Another important point is to prevent the tip from withdrawing from the liquid during the complete aspiration step.

To make a good contact between the tip and the liquid, three positioning modes are available:

- Moving the tip to a fixed height: For this, the exact height of the liquid surface has to be known in advance.
- Using the capacitive Liquid Level Detection (cLLD): For conductive liquids, capacitive LLD should be used. Please make sure conductive (black) tips are used.
- Using the pressure Liquid Level Detection (pLLD): For non-conductive liquids, or in case of an insufficient electrical coupling between container bottom and carrier, pressure LLD should be used.

NOTE

The capacitive Liquid Level Detection is available on the 1000µl- and 5 mL-pipetting channels, the CO-RE 96, the CO-RE 96 TADM and the CO-RE 384 Probe Head. The pressure LLD is only available on the 1000µl-pipetting channels and 5 mL-pipetting channels.

Fixed height

For the fixed height, a value must be chosen which ensures that the tip is permanently below the liquid level. The programmer must prevent aspiration of air instead of liquid. See Following the Liquid Level below.

Liquid Level Detection, LLD

Vigorously and with more care, the liquid level of the vessel to be aspirated from can be detected. This can be provided by the Microlab instrument Liquid Level Detection (LLD) feature based on either capacitive (cLLD) or pressure (pLLD) signal detection.

Capacitive Liquid Level Detection, cLLD

If conductive liquids are to be pipetted, Hamilton recommends using the advantage of the capacitive LLD. The sensitivity of the capacitive LLD that is to be used depends on the vessel size and the conductivity (or polarity) of the liquid that is to be detected. For a solution of 0.1% NaCl in distilled water, the required sensitivities are:

cLLD setting	Sensitivity level	Vessel
1	Very High	384-well plates
2	High	96-well round-bottom plates
3	Medium	96-well flat-bottom plates
4	Low	Tubes, reagent reservoirs or any other large vessels

The following table gives the minimum volume a single pipetting channel can detect in various containers for the capacitive Liquid Level Detection.

Labware	V_{min} [µl]	Carrier
Tubes, 16 mm x 100 mm	200	SMP_CAR_24
Tubes, 12 mm x 75 mm	150	SMP_CAR_32
Eppendorf tubes 1.5 ml	50	SMP_CAR_32_EPIL
Eppendorf tubes 0.5 ml	50	SMP_CAR_32_EPIS
96-well PCR plate	50	PLT_CAR_L5PCR
96-well flat-bottom micro plate	75	PLT_CAR_L5MD
384-well flat-bottom micro plate	50	PLT_CAR_L5MD
96-deep well microplate (archive)	150	PLT_CAR_L5AC

NOTE

Using an ionic buffer in the assay in place of distilled water may help to overcome Liquid Level Detection problems.

Use only original Hamilton labware Carriers, disposables or needles. For a proper capacitive Liquid Level Detection, a sufficient conductive coupling of carrier and labware (tubes or micro plates) is crucial.

Pressure Liquid Level Detection, pLLD

When pipetting non-conductive liquids or in the case of an insufficient electrical coupling between the container bottom and the carrier, Hamilton recommends using the advantage of pressure LLD.

NOTE

Pressure LLD only works with new and empty tips. The suitable settings depend on the tip size and on the type of liquid.

Settings available for example, the 1000 µL-pipetting channels:

pLLD Setting	Sensitivity level	Tip	Liquid
1	Very High	Standard	Low boiling point, low viscosity
2	High	High	Low boiling point, low viscosity
3	Medium	Standard	Water or higher viscosity
4	Low	High	Water or higher viscosity

In the case of aspiration from foaming liquids, capacitive Liquid Level Detection in particular may not detect the surface properly. As an alternative, try pressure LLD, or a combination of both. If a combination of both LLD types is used, the maximum height difference between the two independent LLDs can be used as a parameter.

Submerge Depth

Once the liquid surface is detected, an additional immersion depth of 2 mm (specified by default) is used to prevent the aspiration of air.

Following the Liquid Level

The tip follows the decreasing liquid level (specified by default) according to the aspirated volume. The distance covered while following the liquid level is computed from the known geometry of the liquid container.

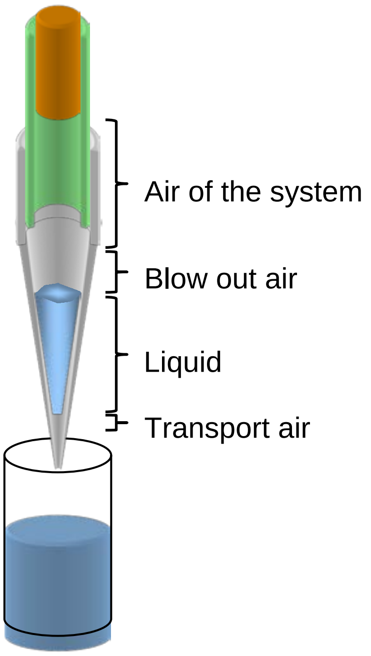
Swap Speed

In order to prevent droplets at the tip/needle orifice, the pipetting channel is moving at a slow speed out of the liquid.

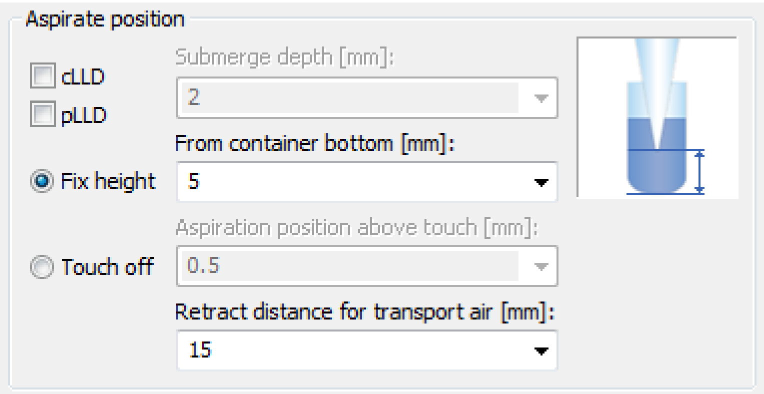
Transport Air

After pulling out of the liquid and before moving to the target container, a variable amount of transport air is aspirated to prevent droplet formation.

At the end of an aspiration step the situation in the tip is as shown below.



When using a fixed height aspiration (or dispense), the position of the transport air intake can be defined by the parameter "**Retract distance for transport air**". Using this value makes sure that the tip's end is out of liquid before aspirating the transport air.

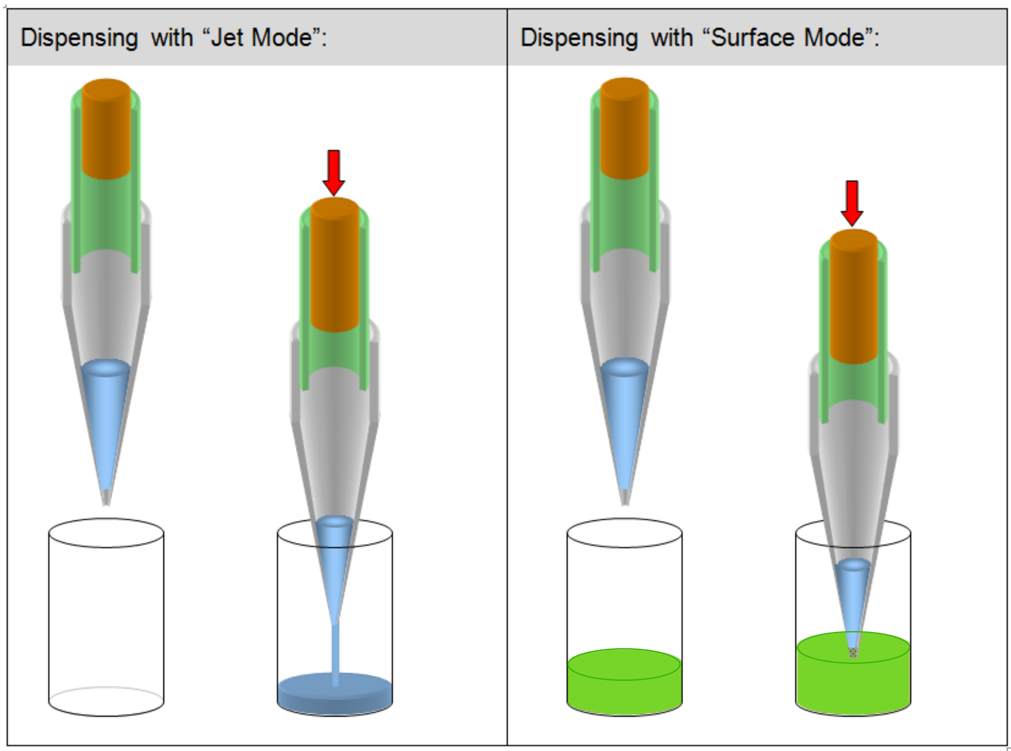


13.2.3 Dispense

At the end of the aspiration step, transport air has been aspirated. The first step of the dispensing procedure is to search for the liquid surface or to move to a fixed height.

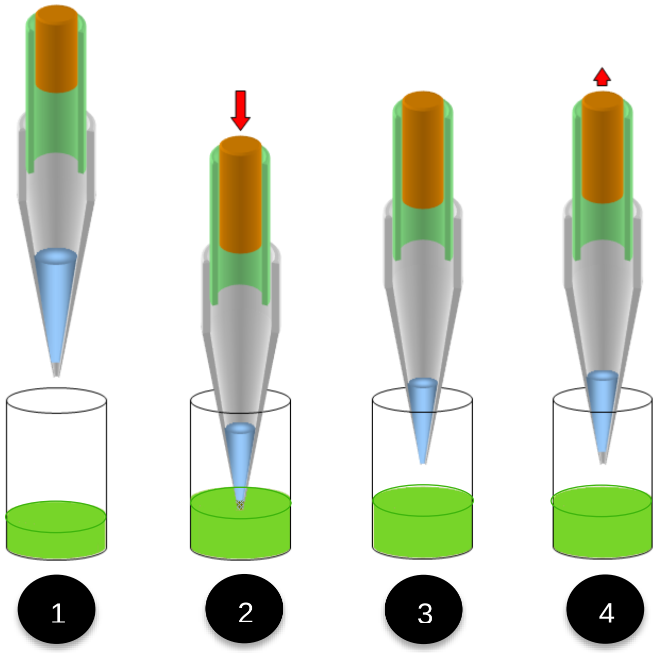
Dispensing of the liquid may occur with three different modes:

- Either onto/into a (liquid) surface
- In a free jet
- Onto the side of the well



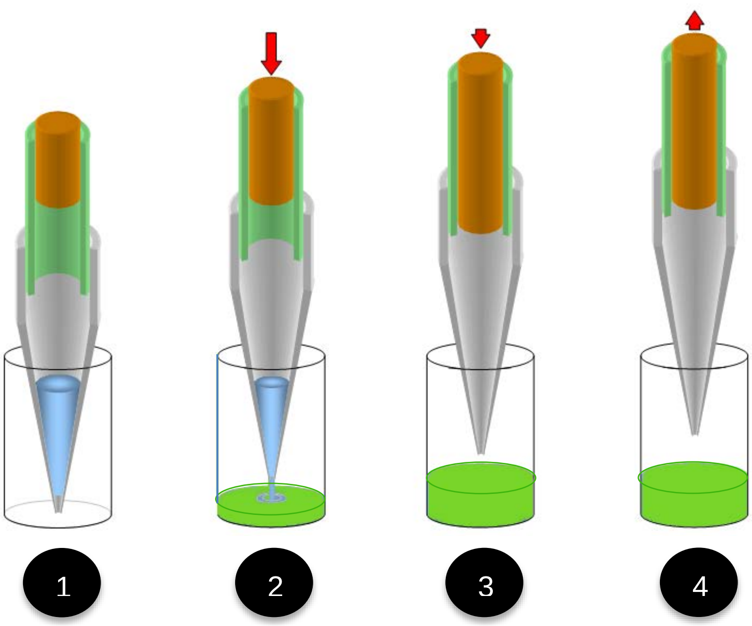
- In order to ensure that the specified accuracy is achieved, volumes below 20 µL should always be dispensed onto or into a (liquid) surface. For dispensing at liquid surface, use cLLD to detect the position of the surface and then dispense while following the rising liquid level.
- When the liquid level is known, dispensing from a fixed height while following the rising liquid level is also possible.
- For volumes larger than 20 µL, the liquid can be dispensed in a jet without touching the surface. To dispense in a jet, specify a position a few millimeters above the surface and dispense following the rising liquid level. For dispensing in a jet low volume, a varying amount of blow-out air is used to make sure that all liquid is dispensed from the tip.
- If only a part of the liquid is dispensed with the jet mode, a stop back volume can be aspirated at the end of the dispense action. This will improve the droplet cut-off at the end of each dispense. In the last step of the dispense procedure, before any x- or y-movement occurs, a variable amount of transport air is aspirated to prevent droplet formation. The transport air is aspirated with the tip above the liquid surface.

The “**Surface Part Volume**” Dispensing Mode works as follows:



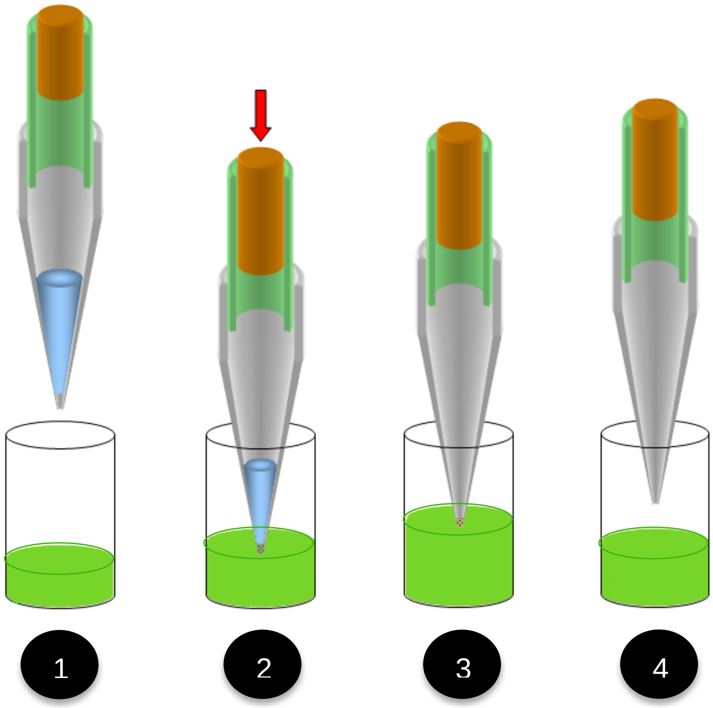
1. When the target well is reached, the pipetting channel starts searching for the liquid surface (cLLD).
2. At an immersion depth of 2 mm (default setting), transport air and liquid (part volume) are dispensed.
3. The pipetting channel moves with swap speed to a distance of 5 mm (default retract distance) above liquid level.
4. Aspiration of the transport air.

The “**Jet Part Volume**” Dispensing Mode works as follows:



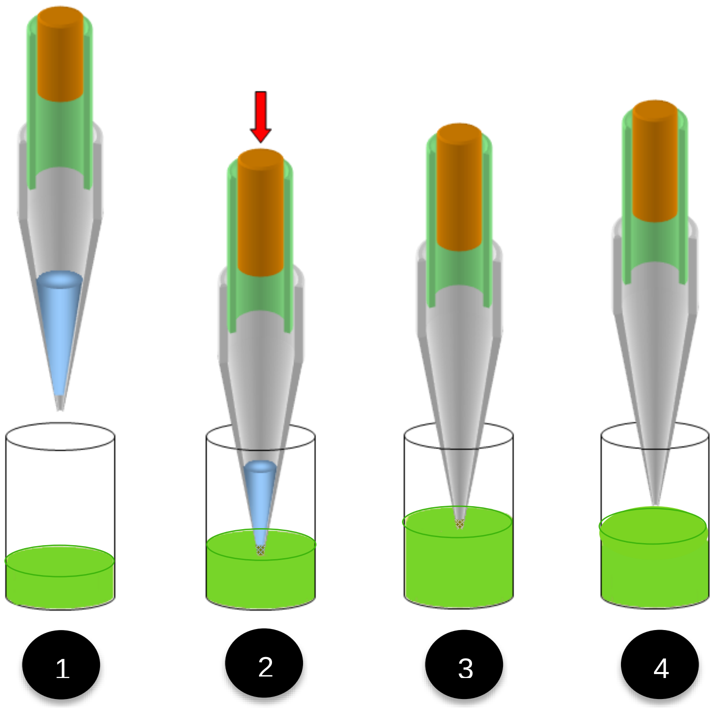
1. When the target well is reached, the pipetting channel starts to move to the fixed height (dispensing height).
2. At the dispensing height the transport air and liquid (part volume) are dispensed while the pipetting channel is moving up in z-direction (following the liquid level).
3. The pipetting channel moves up.
4. Aspiration of the transport air.

The "**Jet Empty Tip**" Dispensing Mode works as follows:



1. When the empty target well is reached, the pipetting channel moves to a fixed height (e.g. 2 mm above the bottom of the well).
2. Transport air and liquid are dispensed while the pipetting channel is moving up in z-direction (following the liquid level).
3. The blow out volume is dispensed: empty tip.
4. The pipetting channel moves up, and then the aspiration of transport air follows.

The “**Surface Empty Tip**” Dispensing Mode works as follows:



1. When the target well is reached, the pipetting channel starts searching for the liquid surface (cLLD).
2. At an immersion depth of 2 mm (default setting), transport air and liquid are dispensed while the pipetting channel is moving up in z-direction (following the liquid level).
3. The complete liquid and blow out volume is dispensed: empty tip.
4. The pipetting channel out of the liquid using swap speed, then the aspiration of transport air follows.

13.2.4 Tip Eject

Ejecting the used tip into the waste container of the Microlab instrument is the final step. A used needle will be returned to the wash station, where the wash process can be started directly.

13.2.5 Avoiding Contamination

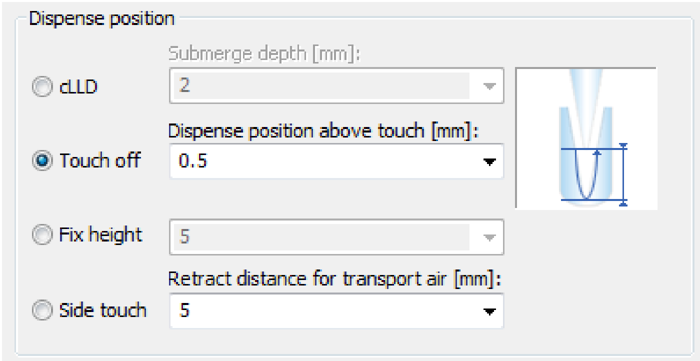
If cross-contamination is a concern, consider the following approaches:

- Use only Hamilton tips on the Microlab instrument.
- Use new tips for every pipetting step, to avoid carry-over between different wells or containers.
- Use filter tips in order to avoid contamination of the pipetting channel by jets, aerosols, etc.
- Dispense any compounds at risk for cross-contamination onto a surface. Dispensing in a jet may produce aerosols and thus can cause cross-contamination.

Dispense using a residual volume (i.e. do not completely empty the tip on dispense). This can be achieved e.g. by aspirating 11 µL and dispensing only 10 µL.

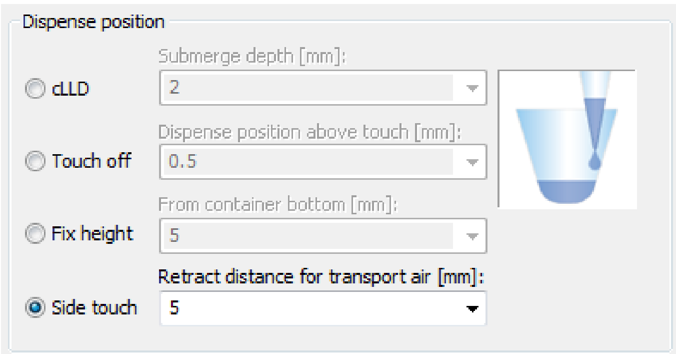
13.2.6 Touch Off

The “**Touch off**” Function is used if very small amounts of liquid will be aspirated or dispensed into a manually placed labware or labware with great tolerances. The “**Touch off**” Function will move to a certain height over the well bottom and smoothly move downwards. As soon as the tip hits the bottom of the container, the motor current of the z-drive increases. This change will be detected, and the z-move stops. From that position, the pipetting channels moves back up the specified distance “**Dispense position above touch**” and starts dispensing.



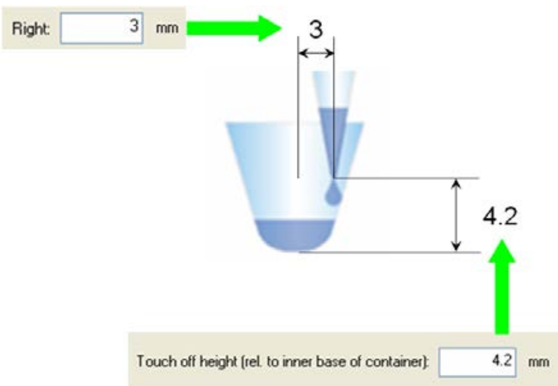
13.2.7 Side Touch

For a small amount of liquid, there is the possibility of dispensing liquid to the side of the well with the “**Side touch**” Mode (available in the Single Steps / Easy Steps).



The “**Side touch**” Mode will move the tip to a specified height in the center of the container, and then moves right (always right). At this position, the dispensing of the liquid starts.

The values for “**Touch off height**” and “**Right**” move have to be defined in the container data in the Labware Editor.



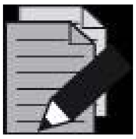
13.3 Liquid Classes, Pipetting Modes and More

Pipetting

In general, pipetting by the principle of air displacement (as with handheld pipettes) is sensitive to the following:

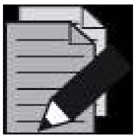
- Manner of pipetting (i.e. surface or jet, empty tip or part volume)
- Tip or needle type (geometry, tip orifice)
- Environmental effects (temperature, pressure, humidity)
- Liquid type (viscosity, evaporation)

The pipetting mode (e.g., surface or jet mode) and the liquid class determine the behavior of the instrument. Pipetting mode and liquid class represent two independent sets of information, both of which have to be specified.



NOTE

Always use the same liquid class for one aspiration and dispense cycle. Otherwise, uncalculated residual volumes may be left within the tip, or other errors regarding plunger position / movements may occur.



NOTE

Long delays after the liquid aspiration (e.g., run pause, wait for timer, user dialog) may lead to an ADC error if the piston reaches its end position or dripping if ADC is not used. Program the method accordingly to avoid these errors.

13.3.1 Aspiration

For aspiration, three modes are available:

- “**Aspiration**”, for all standard cases.
- “**Consecutive Aspiration**” for aspiration with a tip that has already aspirated liquid (e.g. if aspirating out of three different containers before the dispense step).
- “**Aspirate All**” for aspiration of all the liquid in a container (specify a volume larger than what is expected to be in the container). In this case, aspiration monitoring is deactivated, and the tip will follow the falling liquid level (if specified) to the bottom of the container, staying there for the rest of the aspiration.

13.3.2 Dispense

For dispense, five modes are available:

- “**Surface Part Volume**” for dispensing only a part of the liquid in the tip to a surface, leaving a residual volume in the tip,
- “**Surface Empty Tip**” for dispensing all the liquid in the tip to a surface,
- “**Jet Part Volume**” for dispensing only a part of the liquid in the tip in a jet, i.e. without touching a surface, leaving a residual volume in the tip,
- “**Jet Dispense Empty Tip**” for dispensing all the liquid in the tip in a jet.
- “**Drain tip in Jet mode**”

The liquid class stores all relevant background parameters, such as flow rates and volume corrections for one pipetting cycle, (i.e. for one aspiration and the subsequent dispense(s)). Depending on the pipetting mode chosen, only a subset of the parameters of the liquid class is active. According to the different dependencies listed above, liquid classes have attributes related to their intended use: tip type, liquid name, and dispense mode.

Different liquid classes are provided with the VENUS Software. They are optimized for different liquids, tip types, and important pipetting processes, such as aspiration followed by dispensing either to a surface or in a jet. Hamilton has optimized the standard liquid classes with great care to assure the best pipetting accuracy. To change Hamilton standard liquid classes, store the class under a different name first. For special applications, the programmer can define a liquid class to achieve the highest accuracy with the compounds and volumes of interest. The liquid classes can be adapted to the user-specific requirements. For this purpose, a liquid editor comes with the VENUS Software. It is described in Section 13.4 The Hamilton CO-RE Liquid Editor.

13.3.3 Liquid Handling Examples

Here are some examples of typically used combinations of liquid classes and pipetting modes:

1. **Aspirate 50 µL of a water-like liquid, dispense 50 µL into an empty 96-well plate; use standard tips; change tips every cycle.**

Liquid Class:	StandardVolume_Water_DispenseJet_Empty
Aspiration Mode:	Aspiration
Dispense Mode:	Jet Dispense Empty Tip
Detection:	Aspiration: LLD = pressure or capacitance or both, submerge to a depth of 2mm, following liquid level
Dispense:	Fixed height of 5 mm, not following liquid level

2. **Aspirate a water-like liquid, single dispense into a pre-filled 96-well plate; use low volume tips; change tips every cycle.**

Liquid Class:	LowVolume_Water_DispenseSurface_Empty
Aspiration Mode:	Aspiration
Dispense Mode:	Surface Dispense Empty Tip (in the liquid class selected here, the blow-out volume is 0)
Detection:	Aspiration: LLD = pressure or capacitance or both, submerge depth 2 mm, following liquid level
Dispense:	Capacitive LLD on, following liquid level

3. **Aspirate ≥20 µL of a water-like liquid, dispense the same amount into an empty 96-well plate; keep tips.**

Liquid Class:	StandardVolume_Water_DispenseJet_Empty
Aspiration Mode:	Aspiration
Dispense Mode:	Jet Dispense Empty Tip (Empty Tip only)
Detection:	Aspiration: Capacitive LLD, submerge depth 2 mm, following liquid level
Dispense:	Fixed height of 5 mm, not following liquid level

Comment:	On first aspiration, pre-wetting of the tip by 1-3 mixing cycles is necessary to equalize conditions for initial and subsequent dispenses.

4. **Aliquoting** of liquid means aspirating a given volume at once and dispensing several partial volumes (aliquots) in a jet to different containers. In this frequently used pipetting procedure, the accuracy of the first and the last aliquot are often not within the specified range. Therefore, in order to dispense e.g. 10 aliquots of 20 µL of a water-like liquid with the Microlab instrument, aspirate 240 µL and dispense 20 µL directly back into the container. This is followed by dispensing 10 aliquots of 20 µL each. The last aliquot of 20 µL is discarded to another container or ejected with the tip. In addition, after dispensing of every aliquot, a given amount of air is aspirated and dispensed with the next aliquot.

Liquid Class:	StandardVolumeWaterAliquotJet
Aspiration Mode:	Aspiration
Dispense Mode:	Jet Dispense part volume
Detection:	Aspiration: capacitive LLD, submerge depth 2 mm, following liquid level
Dispense:	Fixed height of 5 mm, not following liquid level

5. **Table of Aliquots**

Tip types are:	Std. = Standard Volume Tip (300 µL) High=High Volume Tip (1000 µL)
Pre-wet:	If “**Yes**”, 3-fold mixing on aspiration with aspiration volume necessary
V(main aliq.):	Volume of main aliquot
V(pre-aliq.):	Volume of pre-aliquot
V(post-aliq.):	Volume of post-aliquot
CV:	Precision (coefficient of variation, for definition see the Technical Specifications in the Microlab instrument Operator’s Manual)
R:	Trueness (for definition see Technical Specifications in the Microlab instrument Operator’s Manual). The R and CV values mentioned here are typical results for measurements
Class:	Liquid class used
A:	”**StandardVolume_Water_AliquotJet**“
B:	“**StandardVolume_Serum_AliquotJet**“
C:	“**HighVolume_Water_AliquotJet**“
D:	“**HighVolume_Serum_AliquotJet**“

The dispense mode for all cases is “**Jet Dispense, Part Volume**“.

The following table shows accuracy of aliquoting volumes in dependency upon various pipetting parameters. Sample values and results for pre- and post-aliquot volumes are listed. Please note that the examples for water* and the following are not technical specifications:

Channel type	Tip type	Liquid	Pre-wet	V [µL] main aliq.	No. of aliq.	V [µL] pre-aliq.	V [µL] post-aliq.	CV [%]	R [%]	Class
1000	Std	Water	Yes	10	12	20	>10	3.9	-3.8	A
1000	Std	Water	Yes	20	12	20	20	2.5	-3.2	A
1000	Std	Water	Yes	50	4	50	20	2.0	-1.5	A
1000	High	Water	No	20	12	20	20	5	-1.6	C
1000	High	Water	No	50	12	50	50	2.5	-1.2	C
1000	High	Water	No	100	8	50	100	1.5	-0.9	C
1000	High	Water	No	200	4	50	100	1.5	-1.5	C

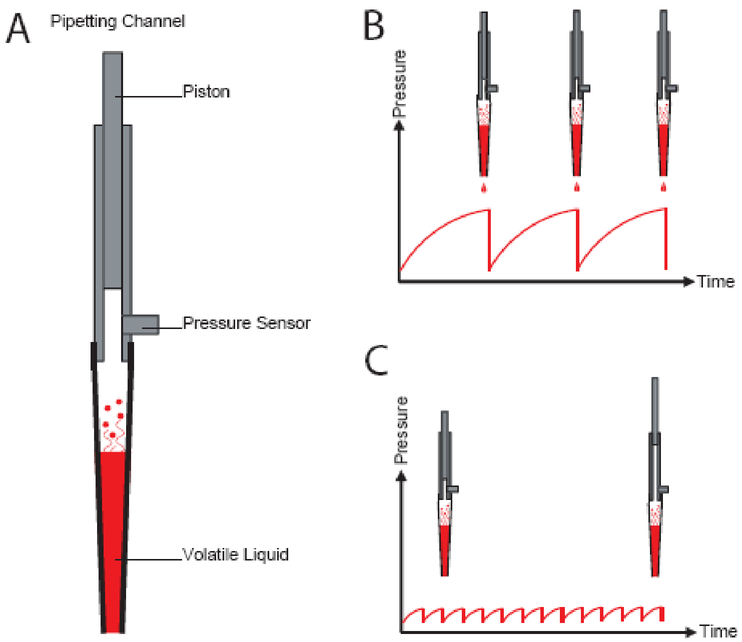
Other liquids data is available upon request. Please ask a Hamilton Representative for further information.

13.3.4 Anti-Droplet Control (ADC)

The Anti-Droplet Control (ADC) function offers a way to prevent droplet formation at the tip of tips or needles while pipetting highly volatile liquids.

These liquids cause droplets because the high vapor pressure of the volatile solvents (e.g. acetone, ethanol, diethyl ether) results in a pressure increase in the tip. This pressure will push the liquid out of the tip (fig A and B).

The working principle of ADC is to measure the pressure inside the pipetting channel (with the built-in pressure sensor) and compensate the increasing pressure by moving the piston upwards. The evaporated volume of the liquid will now be compensated.



Anti-droplet control (ADC)

ATTENTION

The following statements will help in order to get best pipetting results using ADC:

Use ADC only for pipetting steps where a change of pressure inside the tip is expected (e.g. if using a liquid with high vapor pressure).

The air transport volume should be very small or zero. If necessary, create a new liquid class.

The swap speed should be slow. If necessary, create a new liquid class.

Keep in mind that the piston may not be in its initial position when tips are reused without ejecting. ADC might not be working after several steps.

*ADC can be switched on using the appropriate steps from "**HSLMLStarLib.hsl**".*

ADC works on both pipetting channel types, 1000µl and 5 mL.

13.3.5 Monitored Air Displacement (MAD)

The Microlab instrument is equipped with an aspiration monitoring feature. During the aspiration process, the pressure within the pipetting channel is measured in real time. Analyzing the shape of the p(t) curve, the system can distinguish the following situations:

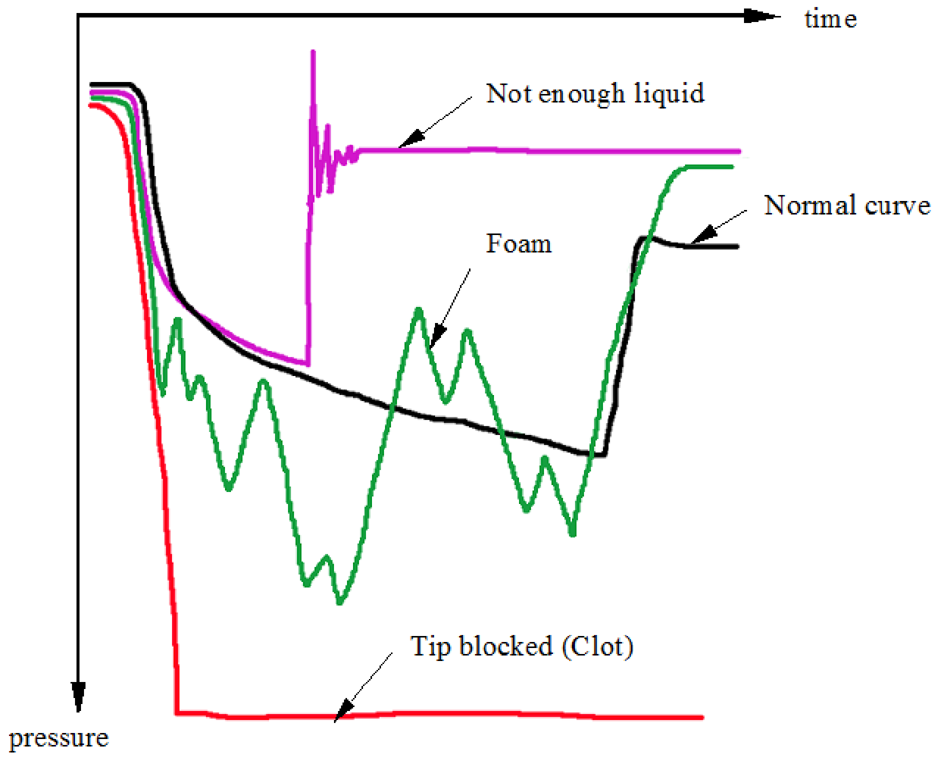
- A correct aspiration takes place
- Air is aspirated into the tip (because, for example, the container has not been filled properly)
- A clot blocks the tip

NOTE

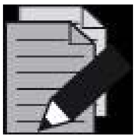
MAD is available on the 1000µl-pipetting channels and the 5 mL-pipetting channels.

The aspiration monitoring can be switched on and off for each individual aspiration step of a method using the steps from "**HLSMl_StarLib.hsl**". For pressure-based clot detection, a threshold can be given in arbitrary A/D (analog/digital) values (typically 100 A/D values). The range of A/D values of the pressure sensor is from around 800 (at ambient pressure) to <10 A/D values for 18 mbar below ambient pressure. For comparison, the hydrostatic pressure of 100 µL of water in our standard tip is around 2 mbar.

The following diagram shows the principle of aspiration monitoring, based on pressure.



Aspiration monitoring based on pressure



NOTE

Pressure-based monitoring works with unused as well as used disposable tips and needles.

If multiple Aspirate/Dispense cycles are executed within a loop, make sure that the (last) dispense step uses the Pipetting mode 'Drain tip'.

The volume range for the MAD depends on the specific assay. The lower limit in many cases is an aspiration volume of 50 µL.

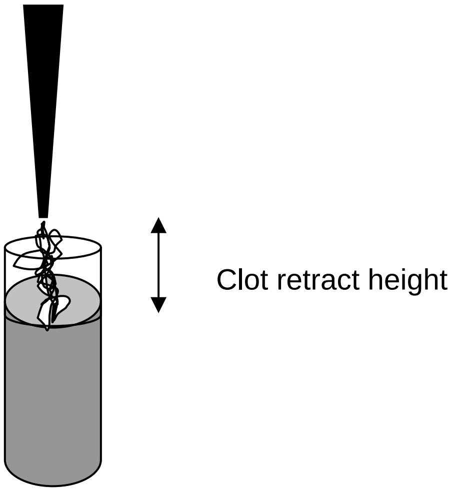
Pressure-based monitoring has been optimized for liquid solutions only.

13.3.6 Capacitance-based Clot Detection

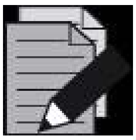
In addition to pressure-based clot detection, the Microlab instrument is equipped with capacitance-based clot detection. This detection approach works when the aspiration with capacitance Liquid Level Detection is switched on.

Functionality

The system measures the conductive signal when the tip leaves the liquid after aspiration. Due to the air gap between tip and liquid, the capacitance signal will vanish once a given height is reached (the "**Clot retract height**", which is specified within the liquid class). If a clot is present, it bridges the distance and the signal will remain, resulting in an error message. A typical clot retract height is 2-5 mm, as illustrated below. This clot detection is independent from pressure-based monitoring.



The capacitance-based clot detection is only valid, if the liquid parameter "**Clot retract height**" is set (>0). Refer to Section 13.4 The Hamilton CO-RE Liquid Editor.



NOTE

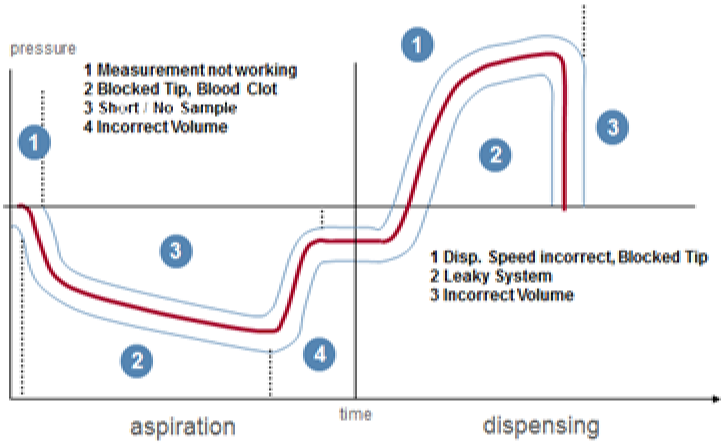
The capacitance-based clot detection is available on the 1000µl-pipetting channels and the 5 mL-pipetting channels.

13.3.7 TADM – Total Aspiration and Dispensing Monitoring

TADM (Total Aspiration and Dispensing Monitoring) is a tool to increase the safety and the robustness of pipetting processes. However, it is not part of the standard software package.

The Principle of TADM

The pressure inside each individual pipetting channel is constantly recorded during aspiration and dispensing. The values obtained by the pressure sensor during a pipetting step (aspiration or dispensing – red curve) can be compared to values defined by the user (grey curves). This allows real-time monitoring of the pipetting process.



The differences between error-free pipetting steps and erroneous ones are used to distinguish the results of the pipetting steps. Basically, there are two types of errors that can be detected:

- Pressure above normal (overpressure)
- Pressure below normal (underpressure)

TADM verifies that a sample has been transferred with a traceable digital audit trail which is particularly beneficial for *In Vitro Diagnostic (IVD)* laboratories.

Peaks and fraying of TADM curves can be a sign that the liquid class and / or the pipetting process are not defined in a precise and robust way. With the help of TADM, the pipetting process can thus be optimized and made more robust.

NOTE

TADM is available on the 1000µl-pipetting channels, 5 mL-pipetting channels, MagPip Channels (STAR^{V} / VANTAGE only), and on the CO-RE 96 Probe Head TADM.

13.4 The Hamilton CO-RE Liquid Editor

13.4.1 Concept of Liquid Classes

Liquid classes manage the background parameters for pipetting.

Definition

A liquid class is a set of parameters determining the aspiration and dispensing behavior of the pipette appropriate for a given liquid, tip type, and dispense mode. For all aspiration and dispense steps a valid liquid class must be selected.

Predefined liquid classes

Several predefined liquid classes are supplied along with the VENUS Software for the following liquids: water, DMSO, glycerin, acetonitrile, ethanol, plasma and serum. The standard liquid classes cover a wide range of applications, and there will probably be no need to make any changes of the parameter settings to adapt the liquid class to a specific application.

Custom liquid classes

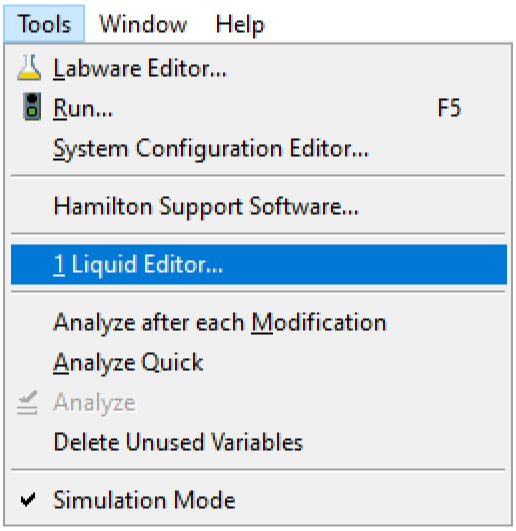
For special applications, a liquid class can be custom-defined. This custom liquid class can be used just like the predefined classes.

Defining liquid classes independent from a method serves to simplify the steps in the method and allows the complete set of parameters to be defined once for all pipetting tasks.

The Liquid Editor is used to display the parameters of the liquid classes and to create custom liquid classes.

13.4.2 Editing Liquid Details

1. To start the Liquid Editor, select " **Liquid Editor**" from the “**Tools**” Menu of the Method Editor.

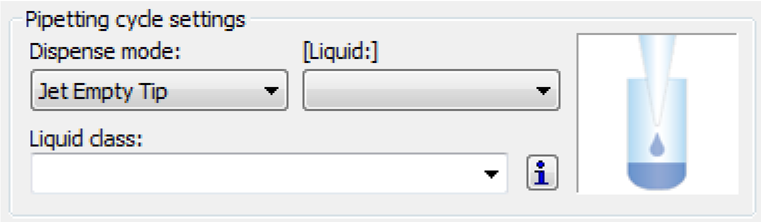


or

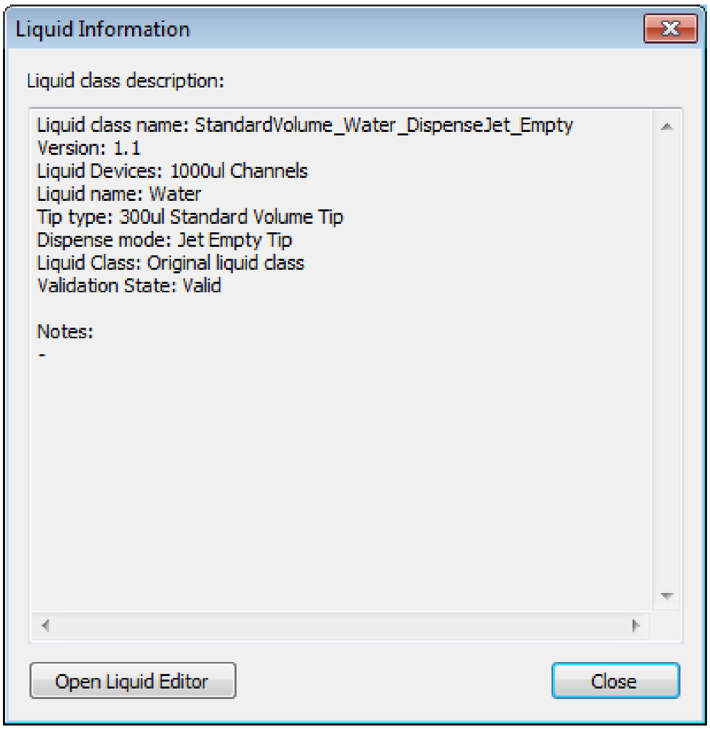
Open the Liquid Class Editor from the VENUS application > System Tools

or

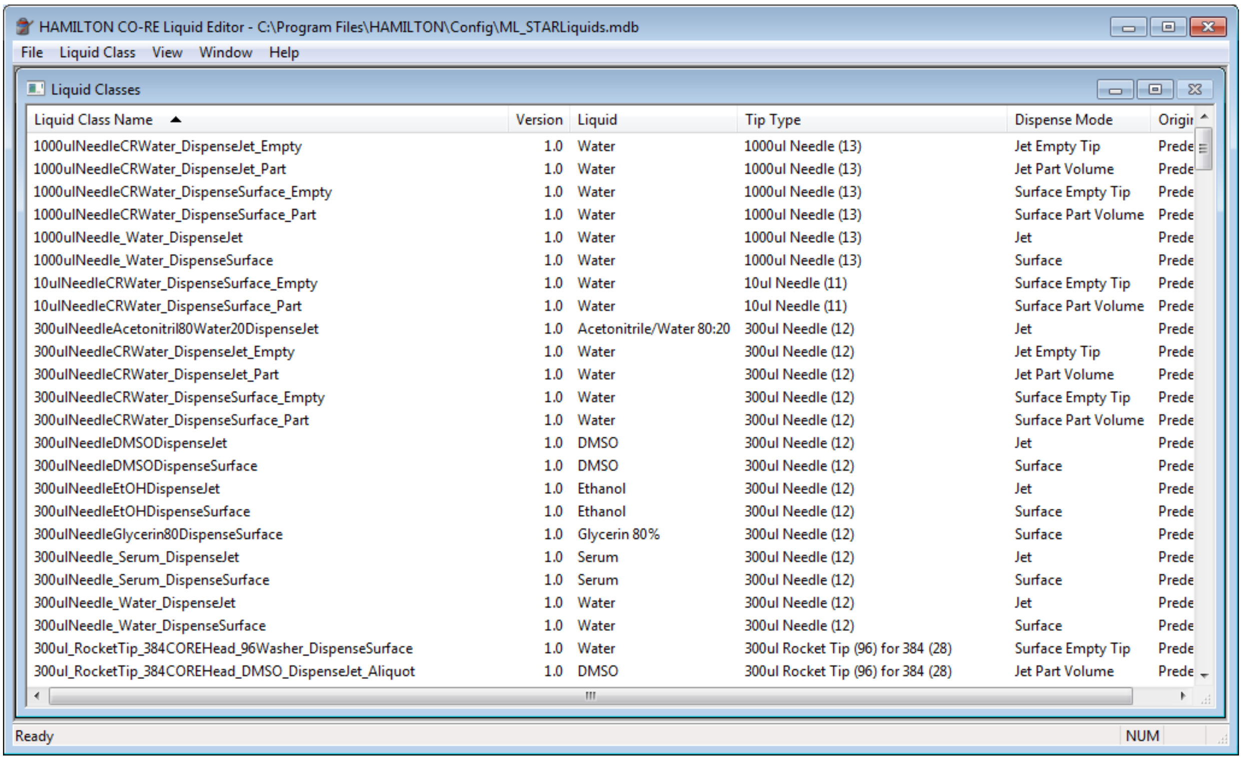
Open a pipetting step and use the info sign to show the liquid details.



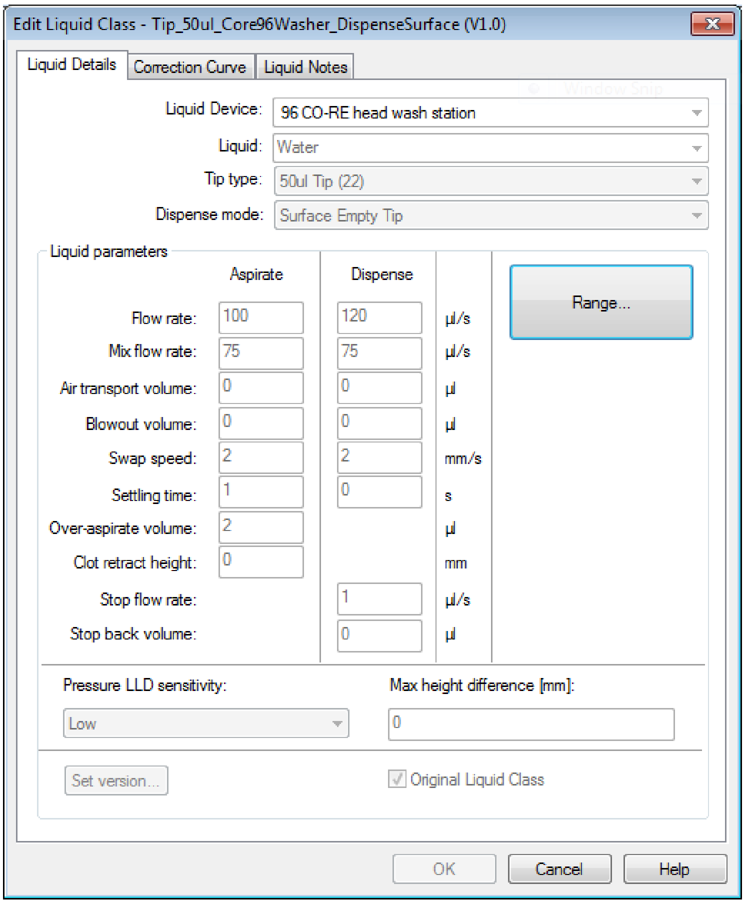
2. From there, open the Liquid Editor:



3. The Liquid Editor Main Window opens with a list of liquid classes as presented below.



4. Double-click on a liquid class. This will activate the “**Edit Liquid Class**” Dialog.

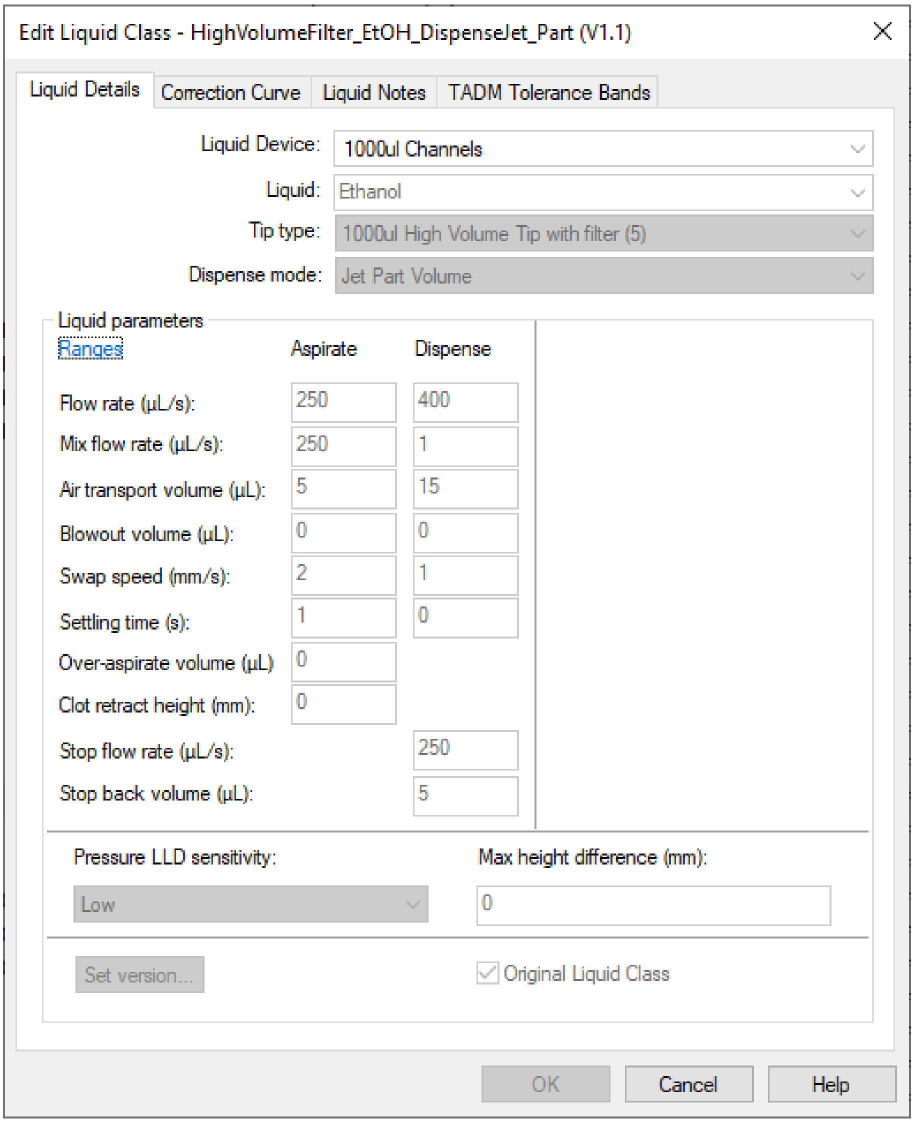


Liquid Details

The "**Liquid Details**" Tab has two sections. In the image on the previous page, the following attributes of the liquid class are shown:

- Liquid device
- Liquid (name)
- Tip (or needle) type
- Dispense mode.

The image below shows the parameters of a liquid class:



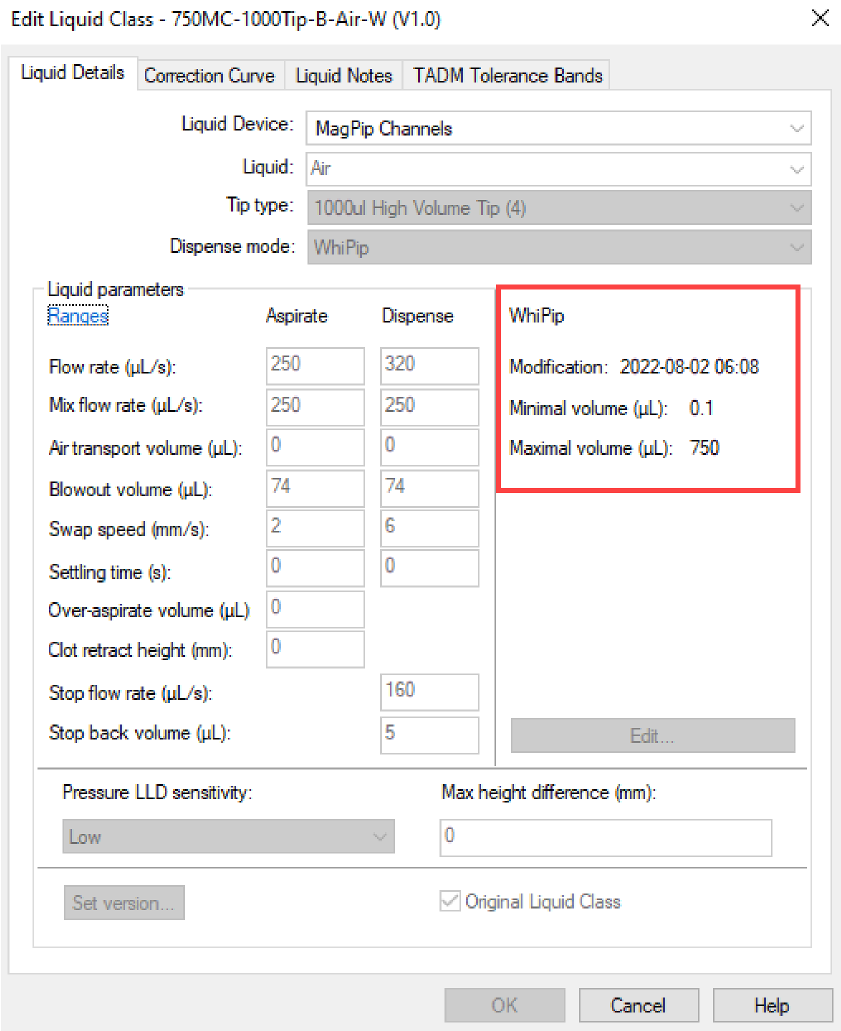
Liquid Parameters

The liquid parameters section shows the appropriate instrument parameters for aspirating and dispensing.

Here is what the various parameters mean:

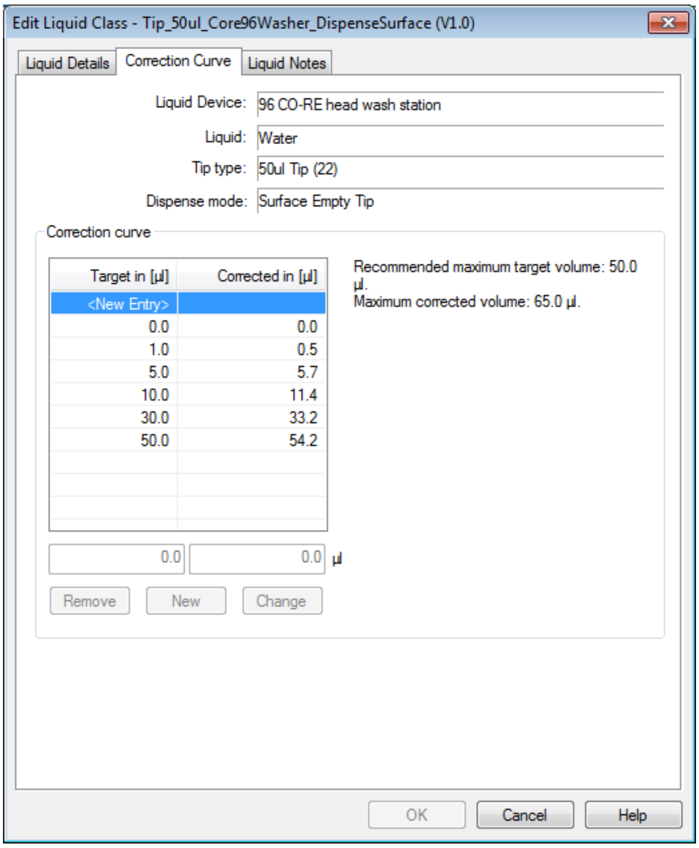
- **“Flow Rate”** and **“Mix Flow Rate”** are volume flows of liquid in µL/s; they correspond to plunger speeds for aspirating, dispensing and mixing.
- **“Air Transport Volume”:** air for transport is aspirated at the end of the aspirate and/or dispense step and automatically dispensed again as an extra volume in the first part of the dispense step.
- **“Blowout Volume”:** blow-out air is taken up first during aspiration. If dispensing will later be done using the “**Empty tip**” Dispense Mode, the entire volume including blow-out air is dispensed in the dispense step.
- **“Swap Speed”** is the speed at which the dispensing head (single pipetting channel, CO-RE 96 Probe Head or CO-RE 384 Probe Head) is drawn up out of the liquid.
- **“Settling Time”** is the time the dispensing head has to wait in the liquid after aspiration/dispense until it begins to withdraw.
- **“Over-Aspirate Volume”** is a kind of pre-wetting volume: on aspirating e.g. 20 µL of liquid, first more than 20 µL is aspirated (20 µL + Over-asp. vol.), so as to pre-wet the tip. Then this volume is dispensed again immediately (still in the aspirate step).
- **“Clot Retract Height”:** a parameter for recognizing clots which determines how high the dispensing head is allowed to travel up out of the liquid if there is a residual cLLD signal after aspiration. It is measured in mm from the height of the liquid surface upwards. If this distance is exceeded, an error message is generated.
- **“Stop Flow Rate”:** dispensing speed of the plunger (expressed as a stream of liquid volume in µL/s), at which the dispense step terminates abruptly. If the “**Dispense flow rate**“ is equal to the “**Stop flow rate**“, the dispense breaks off abruptly after dispensing the volume without slowing down beforehand. If the “**Stop flow rate**“ is set to its minimum permitted value, the plunger movement becomes gradually slower during the dispense until it stops.
- **“Stop Back Volume”:** volume which is aspirated again immediately after dispensing “**Jet Part Volume**” Mode. This volume is aspirated automatically as quickly as possible in order to have a cut-off of the liquid flow.
- **“Pressure LLD Sensitivity”:** Default value for the pressure LLD sensitivity to be used in aspiration steps. Determined by the liquid.
- **“Max Height Difference”:** Default tolerated maximum height difference in mm between pressure and capacitive LLD if both are activated during aspiration.

The liquid classes using MagPip Channels include a 'WhiPip' parameters section. These parameters are installed by default, optimized for each liquid type to use the WhiPip dispense feature of the MagPip Channels. In a WhiPip dispense, the plunger moves at a very fast speed, allowing to dispense volumes as low as 350 nL without contact (in jet mode).



Range Button

1. Click on the **[Range]** Button (see image on the previous page) to see the limitations of the parameters depending on the pipetting tool.
2. Activate the "**Correction Curve**" Tab as shown below.



3. A correction curve shows a target volume and a corrected volume.

Target Volume

The “**Target volume**“ is the volume to be dispensed (the one entered in the pipetting steps).

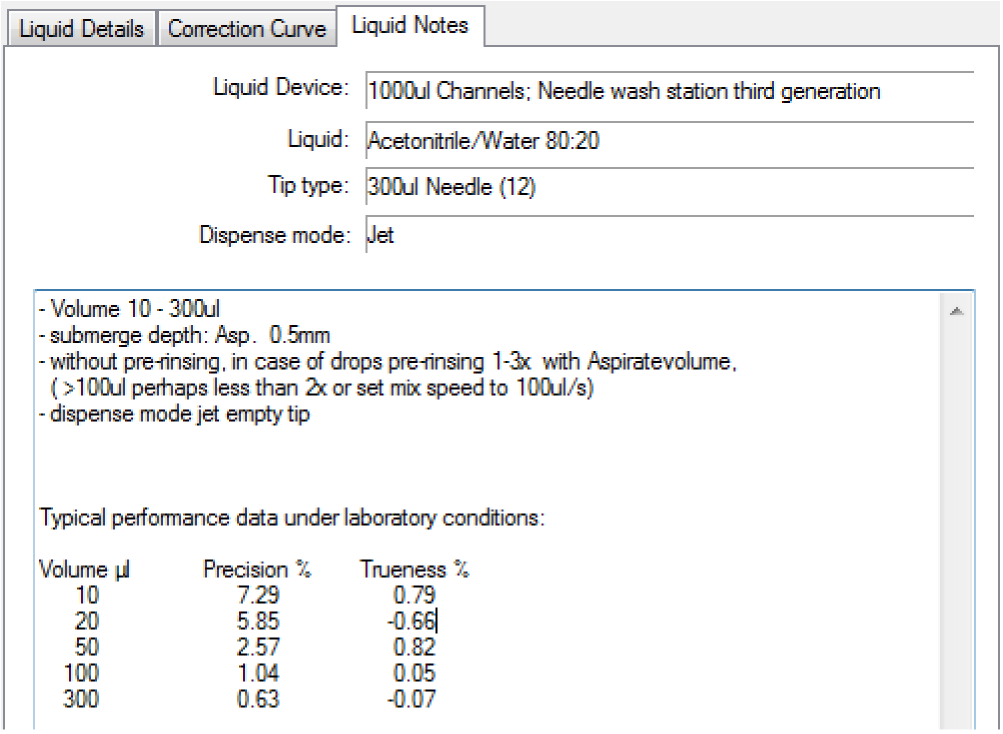
Corrected Volume

The “**Corrected volume**“ is the volume that actually needs to be moved by the plunger for this purpose. In aspiration or dispense steps, the “**Target volume**“ which will actually be dispensed into the vessel must be entered. Corrected volumes for a desired target volume are usually determined gravimetrically.

Correspondingly, a corrected volume of 107.2 µL for a target volume of 100 µL does not mean that 107.2 µL of liquid will be dispensed. When the tip is emptied, 100 µL are dispensed. The correction is mainly due to the properties of the air column above the liquid.

The high flexibility of the liquid classes allows pipetting any liquid with high accuracy. Custom liquid classes are also available upon request from Hamilton’s Application Engineering Group for the customer-specified applications.

Liquid Notes



Under this tab, information can be stored concerning the way the custom-made liquid class is determined. Information about the accuracy reached with this liquid class can also be stored here.

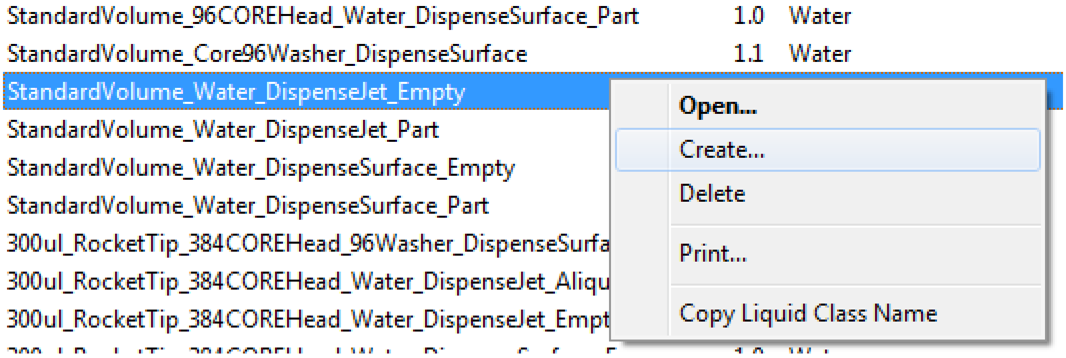
NOTE

The predefined liquid classes that come along with the instrument cannot be changed by the user, but they can be copied, saved under a different name and then edited.

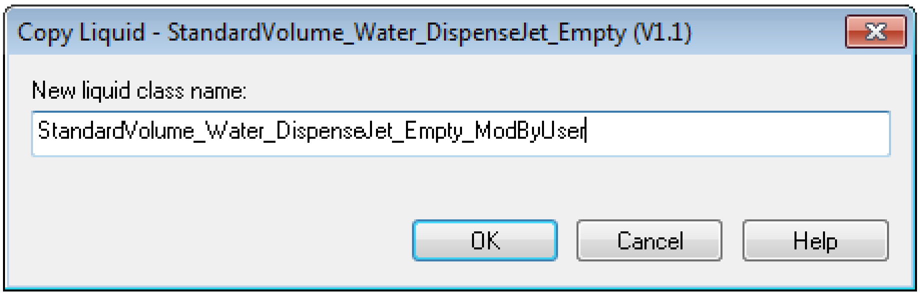
13.4.3 Defining a Custom Liquid Class

To define a custom liquid class, for instance, for a new liquid:

1. Select a predefined liquid class in the liquid editor main window



2. From the Liquid Class Menu, click "**Create**".



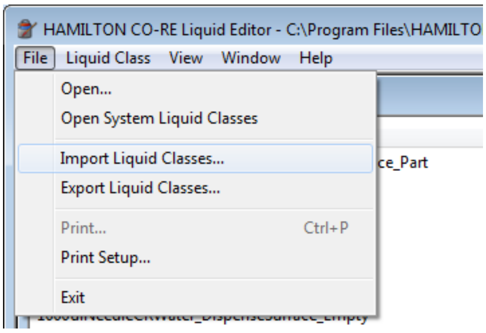
3. Enter the name of the new custom liquid class and click **[OK]**.
4. Back in the Liquid Editor Main Window, select the new liquid class and click **[Open]**.
5. Changes can be made, now that all the parameter input fields in the Liquid Details Window are already enabled.

13.4.4 Importing and Exporting Liquid Classes

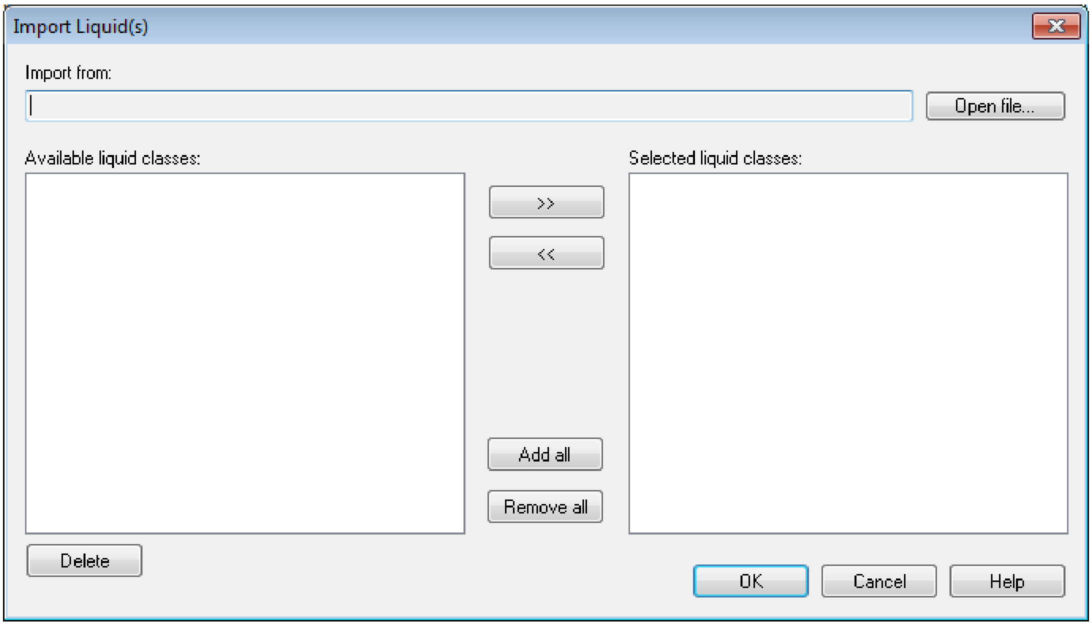
Liquid classes are stored in a database file. Opening the Liquid Editor loads liquids defined in the standard liquid database file (...\Hamilton\Config\ML_STARLiquids.mdb). If other liquids are needed from another liquid configuration file, they can be imported.

To Import a Liquid:

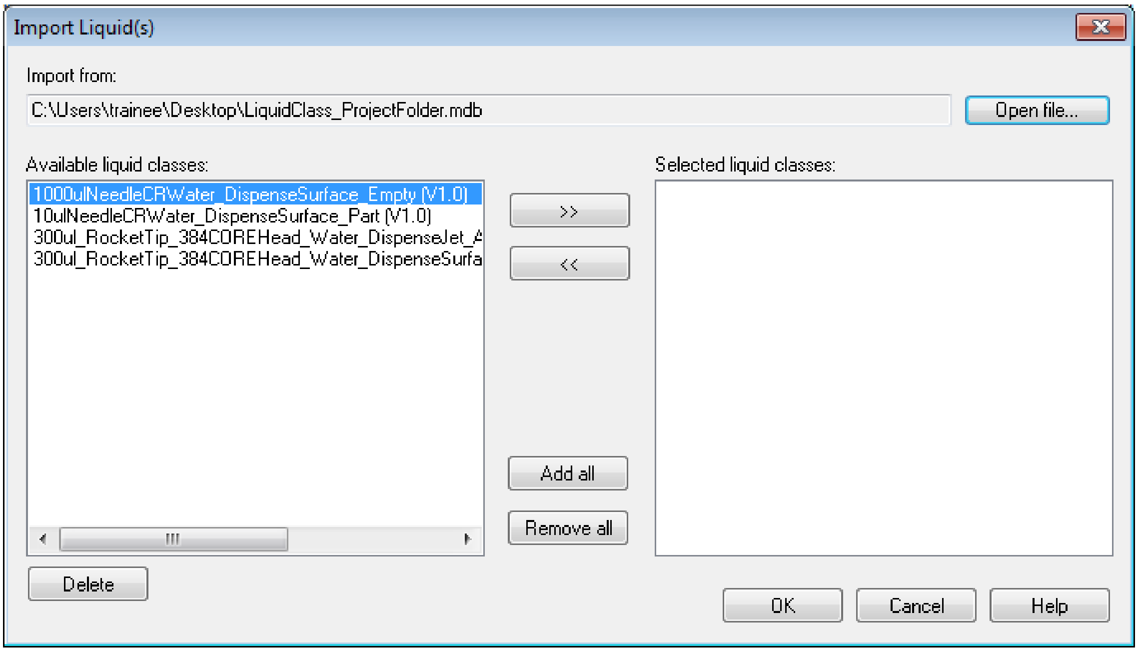
1. Click "**File → Import Liquid Classes...**" in the Liquid Editor Start Window:



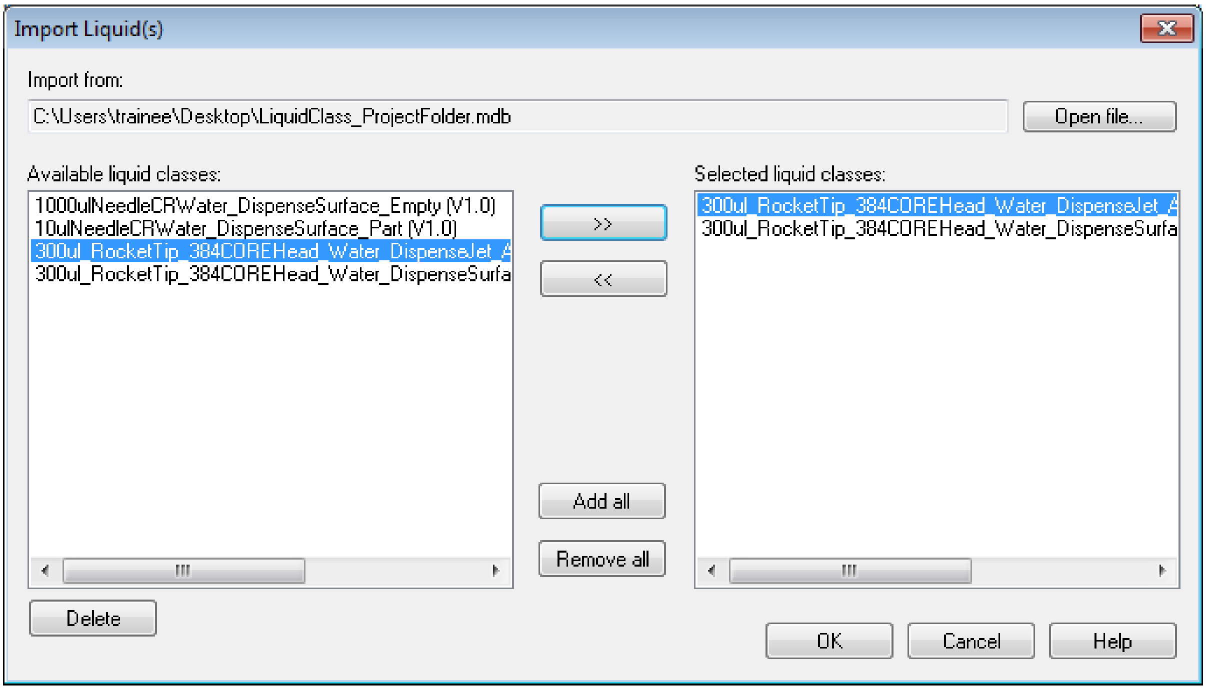
2. After doing so, the "**Import Liquid(s)**" Window will be prompted.



3. Click the **[Open file...]** Button to open the desired liquid database (in *.mdb format).



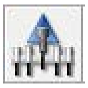
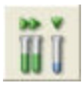
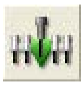
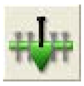
4. In the left window, all liquids of the database to be imported are listed.
5. Click on the list to select the required liquids.
6. Click on **[>>]** to select single liquid classes or use the **[Add all]** Button.



7. The selected liquid classes appear in the right window.
8. To import all the liquids selected in the right window, click **[OK]**.
9. In addition, a similar dialog is available for exporting liquid classes to the configuration files. This dialog is accessed by clicking on "**File → Export Liquid Classes...**".

13.5 Pipetting Steps

13.5.1 Smart Steps

ML_STAR Smart Steps		
Command	**Icon**	**Action Performed**
1000µl Channel Tip Pick up		Pick up disposable tips from tip rack.
1000µl Channel Needle Pick up		Pick up needles from wash station (or racks).
1000µl Channel Pipette Simple (1-1)		Simple pipetting for simple aspiration/dispense cycle**.**
1000µl Channel Pipette Replica (1-n)		Pipetting for cycles of aspirations/dispenses where the liquid from one source is dispensed into multiple target containers (no aliquoting).
1000µl Channel Pipette Pooling (n-1)		Pipetting for cycles of multiple aspirations/dispenses where liquid from multiple source containers is dispensed into one target container.
1000µl Channel Pipette Aliquot		Pipetting for cycles with one aspiration followed by multiple dispenses.
1000µl Channel Tip Eject		Eject disposable tips into tip waste.
1000µl Channel Needle Eject		Release needles in racks or wash station (and start wash).

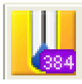
13.5.2 Easy Steps

There are two Easy Steps for the pipetting channels, the CO-RE 96 Probe Head and the CO-RE 384 Probe Head:

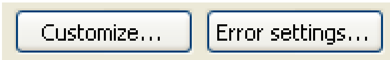
- Aspirate (including the pickup of disposable tips/needles).
- Dispense (including the disposable tips/needles wash and eject).

The Easy Steps are installed by default. These commands can be found in the “**ML_STAR**” Toolbox Group.

ML_STAR Easy Steps		
Command	**Icon**	**Action Performed**
1000µl Channel Aspirate		Aspirate liquid with the 1000µl-pipetting channels
5 mL Channel Aspirate		Aspirate liquid with the 5 mL-pipetting channels
CO-RE 96 Head Aspirate		Aspirate liquid with the CO-RE 96 Probe Head

ML_STAR Easy Steps		
Command	**Icon**	**Action Performed**
CO-RE 384 Head Aspirate		Aspirate liquid with the CO-RE 384 Probe Head
1000µl Channel Dispense		Dispense with the 1000µl-pipetting channels
5 mL Channel Dispense		Dispense with the 5 mL-pipetting channels
CO-RE 96 Head Dispense		Dispense liquid with the CO-RE 96 Probe Head
CO-RE 384 Head Dispense		Dispense liquid with the CO-RE 384 Probe Head

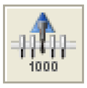
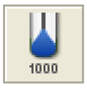
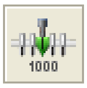
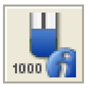
The default settings of the aspirate steps correspond to an ordinary aspiration "**Simple Aspiration**" mode. If any other aspiration mode (Consecutive, All), mix step, LLD setting, etc. is required, clicking the **[Customize…]** Button can modify the command.

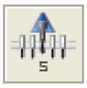
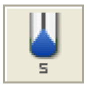
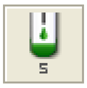
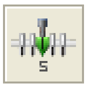
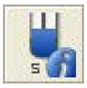


The **[Error settings…]** Dialog allows an error handling approach to be specified.

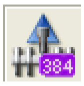
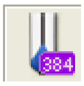
13.5.3 Single Steps

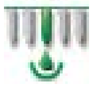
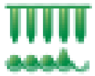
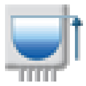
If highest flexibility of the system is required, use the Single Step commands. When using these commands, even most complex liquid handlings can be programmed.

1000 µL-Pipetting Channels		
Command	**Icon**	**Action Performed**
1000µl Channel Tip Pick Up		Picks up a CO-RE tip or needle
1000µl Channel Aspirate		Aspirate liquid from a container
1000µl Channel Dispense		Dispense liquid into a container
1000 µL Channel Dispense on the Fly		Pipettes liquid without stopping in x over a full plate or sequence. Very fast for e.g. reagent distribution
1000µl Channel Tip Eject		Discards the tip into the tip waste or the needle into the wash station or rack
1000µl Channel Get Last Liquid Level		Gets the z-position of the last Liquid Level Detection

5 mL Pipetting Channels		
Command	**Icon**	**Action Performed**
5 mL Channel Tip Pick Up		Picks up a CO-RE tip or needle
5 mL Channel Aspirate		Aspirates liquid from a container
5 mL Channel Dispense		Dispenses liquid into a container
5 mL Channel Dispense on the Fly		Pipettes liquid without stopping in x over a full plate or sequence. Very fast for e.g. reagent distribution
5 mL Channel Tip Eject		Discards the tip into the tip waste or the needle into the wash station or rack
5 mL Channel Get Last Liquid Level		Gets the z position of the last Liquid Level Detection

CO-RE 96 Probe Head		
Command	**Icon**	**Action Performed**
CO-RE 96 Head Tip Pick Up		Pick up CO-RE tips
CO-RE 96 Head Aspirate		Aspirates liquid from a container
CO-RE 96 Head Dispense		Dispenses liquid into a container
CO-RE 96 Head Tip Eject		Discards the tips

CO-RE 384 Probe Head		
Command	**Icon**	**Action Performed**
CO-RE 384 Head Tip Pick Up		Pick up CO-RE tips
CO-RE 384 Head Aspirate		Aspirates liquid from a container
CO-RE 384 Head Dispense		Dispenses liquid into a container
CO-RE 384 Head Tip Eject		Discards the tips

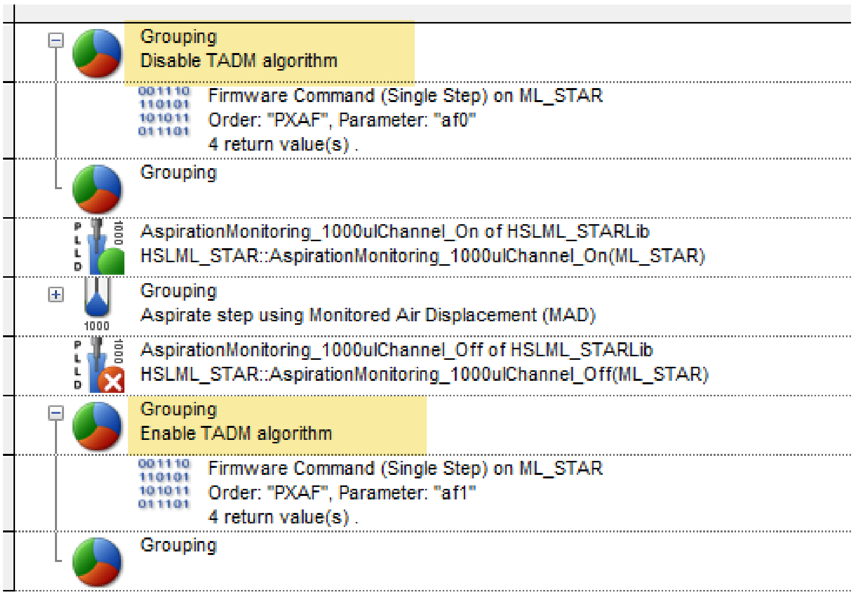
Nano Pipettor		
Command	**Icon**	**Action Performed**
Nano Pipettor Aspirate		Aspirates liquid from a container
Nano Pipettor Dispense		Dispenses liquid into a container
Nano Pipettor Dispense On The Fly		Drops liquid into a sequence without reaspiration and without stopping the movement in x direction
Nano Pipettor Wash		Flush the ceramic tips and clean them in the Ultrasonic bath
Nano Pipettor Prepare		Primes the liquid system and flushes the pipettor or primes the pipettor only
Nano Pipettor Move To Position		Moves the pipetting head to an absolute position, or to one relative to the current position

13.5.4 Process Control

For additional process control of the pipetting steps, the monitoring functions can be switched on and off for a single pipetting step or for a specific part of the method. The commands for the Monitored Air Displacement (MAD) and the Capacitive Clot Detection are functions of the library "**HSLML_STARLib**". To make the commands available, link the library to the method. Add the library to the method through the "**Method → Libraries...**" Menu (refer to Section 11.12.10 Microlab Instrument Library).

NOTE

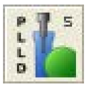
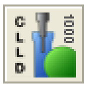
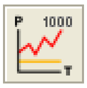
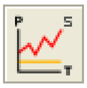
On the Microlab STAR instruments, when TADM is enabled, MAD is automatically deactivated for all the aspiration steps. If MAD is needed in some aspiration steps, the TADM algorithm can be temporarily disabled in the method with firmware commands.



For 1 mL Channels use 'PXAFaf0' to disable TADM, and 'PXAFaf1' to enable.

For 5 mL Channels, use 'LXAFaf0' to disable TADM, and 'LXAFaf1' to enable.

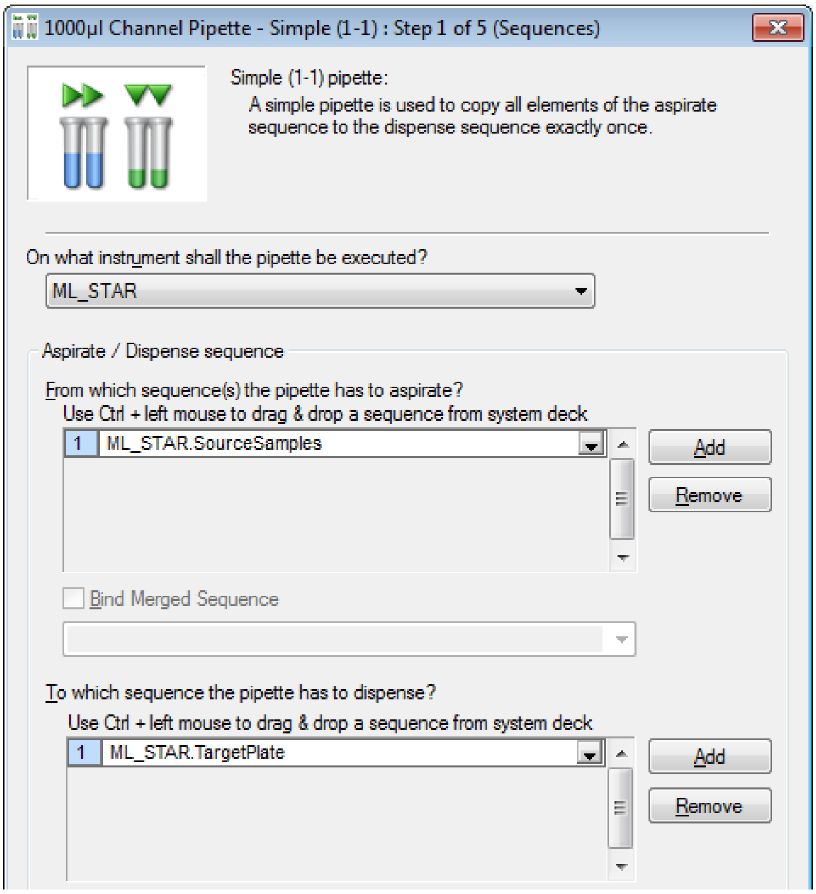
Process Control		
Command	**Icon**	**Action Performed**
AntiDropletControl_1000ulChannel_On		Activates the Anti-Droplet Control (ADC) on the 1000µl-pipetting channels
AntiDropletControl_1000ulChannel_Off		Deactivates the Anti-Droplet Control (ADC) on the 1000µl-pipetting channels
AntiDropletControl_5mlChannel_On		Activates the Anti-Droplet Control (ADC) on the 5 mL-pipetting channels
AntiDropletControl_5mlChannel_Off		Deactivates the Anti-Droplet Control (ADC) on the 5 mL-pipetting channels
AspirationMonitoring_1000ulChannel_Off		Disables the aspiration and clot detection monitoring with the 'pLLD' on the 1000µl-pipetting channels
AspirationMonitoring_1000ulChannel_On		Enables the aspiration and clot detection monitoring with the 'pLLD' on the 1000µl-pipetting channels
AspirationMonitoring_5mlChannel_Off		Enables the aspiration and clot detection monitoring with the 'pLLD' on the 5 mL-pipetting channels

Process Control		
Command	**Icon**	**Action Performed**
AspirationMonitoring_5mlChannel_On		Disables the aspiration and clot detection monitoring with the ‘pLLD’ on the 5 mL-pipetting channels
ClotDetectionMonitoring_1000ulChannel_Off		Enables the clot detection monitoring with the ‘cLLD’ on the 1000µl-pipetting channels
ClotDetectionMonitoring_1000ulChannel_On		Disables the clot detection monitoring with the ‘cLLD’ on the 1000µl-pipetting channels
ClotDetectionMonitoring_5mlChannel_Off		Enables the clot detection monitoring with the ‘cLLD’ on the 5 mL-pipetting channels
ClotDetectionMonitoring_5mlChannel_On		Disables the clot detection monitoring with the ‘cLLD’ on the 5 mL-pipetting channels
SetPressureThreshold_1000ulChannel		Sets the pressure threshold for the aspiration and clot detection monitoring with the ‘pLLD’ on the 1000µl-pipetting channels
SetPressureThreshold_5mlChannel		Sets the pressure threshold for the aspiration and clot detection monitoring with the ‘pLLD’ on the 5 mL-pipetting channels

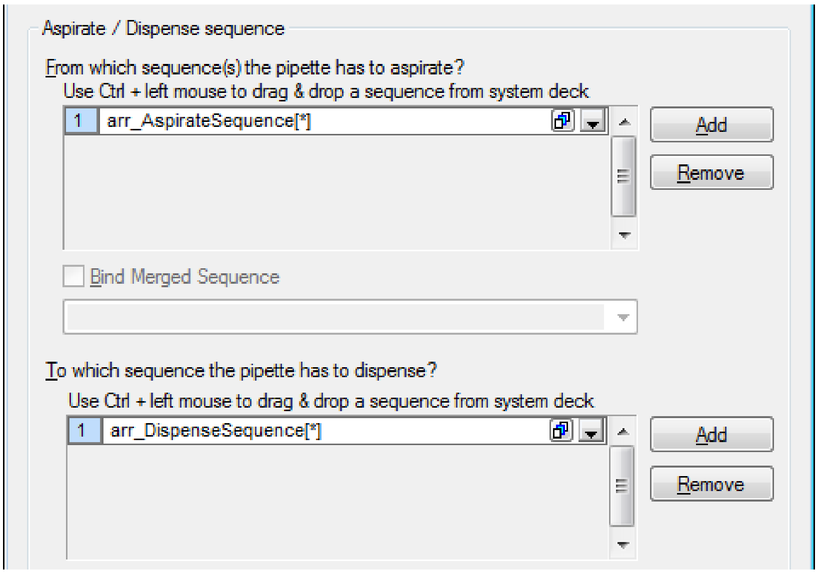
13.5.5 Using Smart Steps

In this section, the general features of the "**Pipette**" Smart Step are explained by means of the “**Simple Pipetting**” Smart Step (the other variants of the “**Pipette**” Command have similar wizards). More examples can be found in Section 12 Demo Methods for the Microlab Instrument.

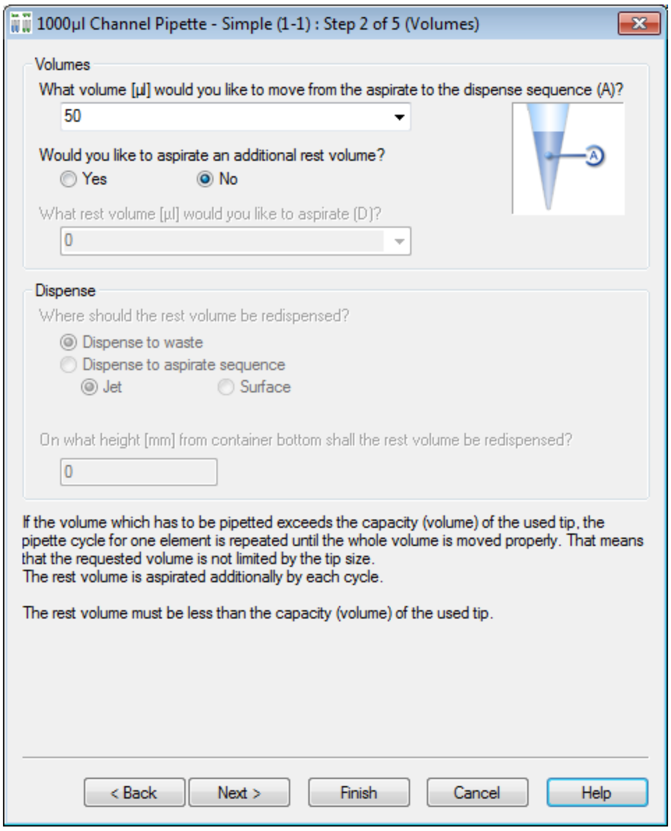
1. Drag one of the "**Pipette**" Smart Steps into the method window. The corresponding step of the "**Pipette**" Smart Step Wizard opens:



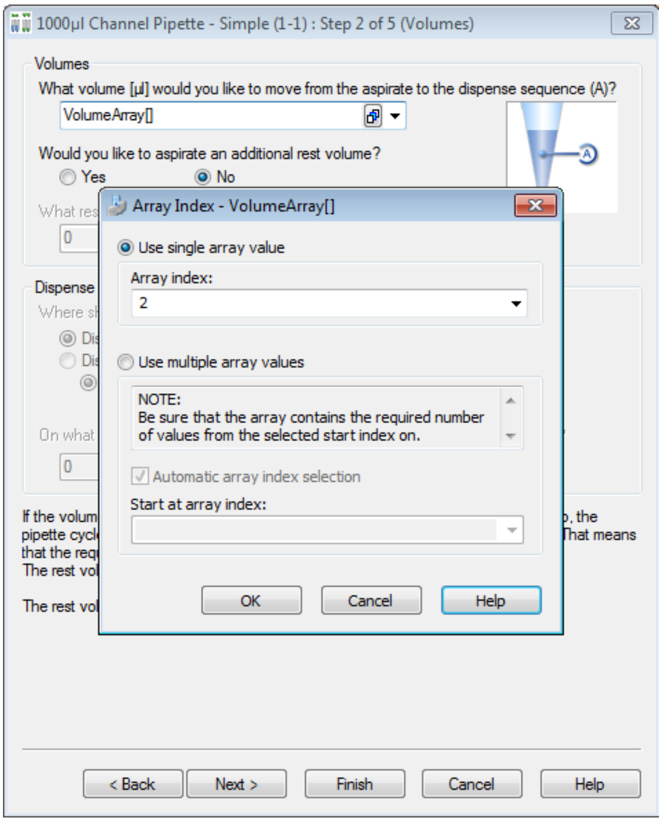
2. Use the drop-down list to identify the instrument on which pipetting must be performed, as well as the aspirate and dispense sequences.
3. Note that the sequences can also be indicated using arrays:



4. Click the **[Next >]** Button. The next page of the Pipetting Wizard will be activated.



5. Note that in this dialog the volume can also be specified using an array as shown below.



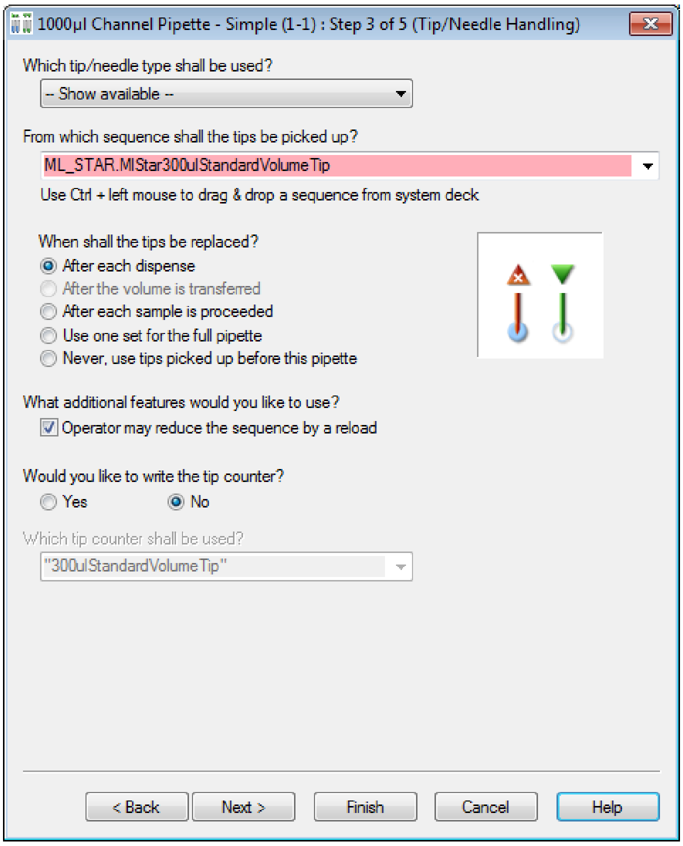
6. Define: the **volume(s)** for the pipetting process, a possible rest volume and its re-dispense target.
7. If a volume for "**Simple**", "**Replica**" or "**Pooling**" is specified and exceeds the volume of the tip, multiple transfers with equally divided volumes will be performed automatically.

Rest Volume

A residual volume "**Rest volume**" in the dialogs may be used for all pipetting modes too. This is handled at the end of a pipetting step after the (last) dispensing step. In the aliquot mode the additional handled residual volume is called "**Post-aliquot**" in contrast to the "**Pre-aliquot**", which is handled before the aspirating step. The pre-aliquot is dispensed back into the aspiration sequence.

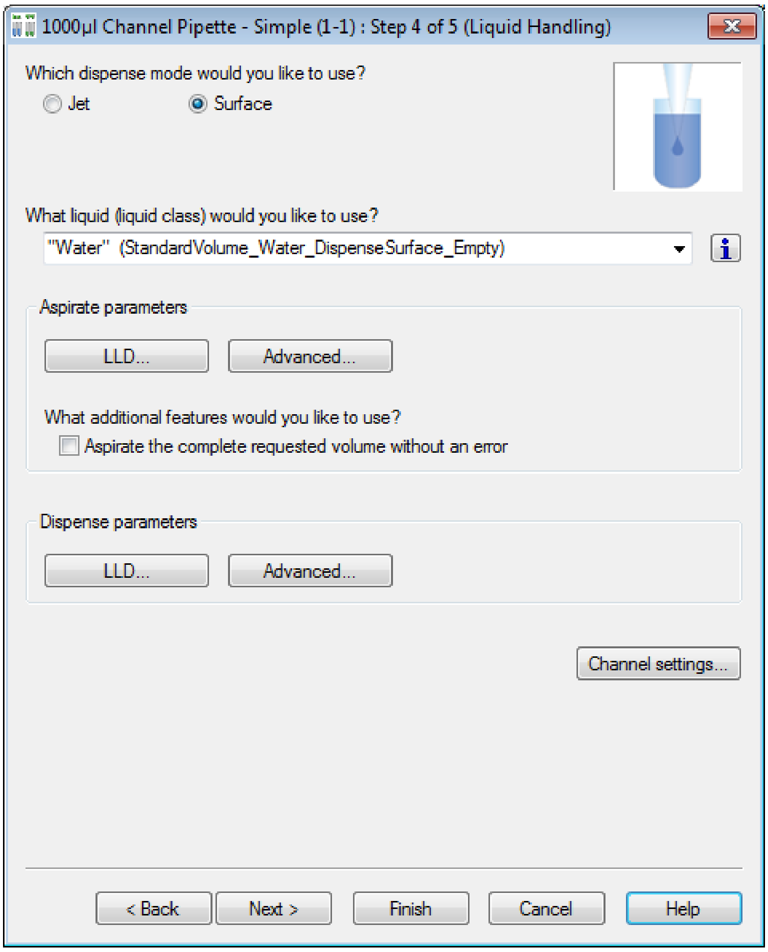
With the "**Rest volume**", the choice is to discard it back into the aspiration sequence, or to dispense it into the waste container. In the previous example, a re-dispense height counted from the bottom of the container has to be given.

1. Click the **[Next >]** Button. The following pipetting wizard dialog will be displayed.

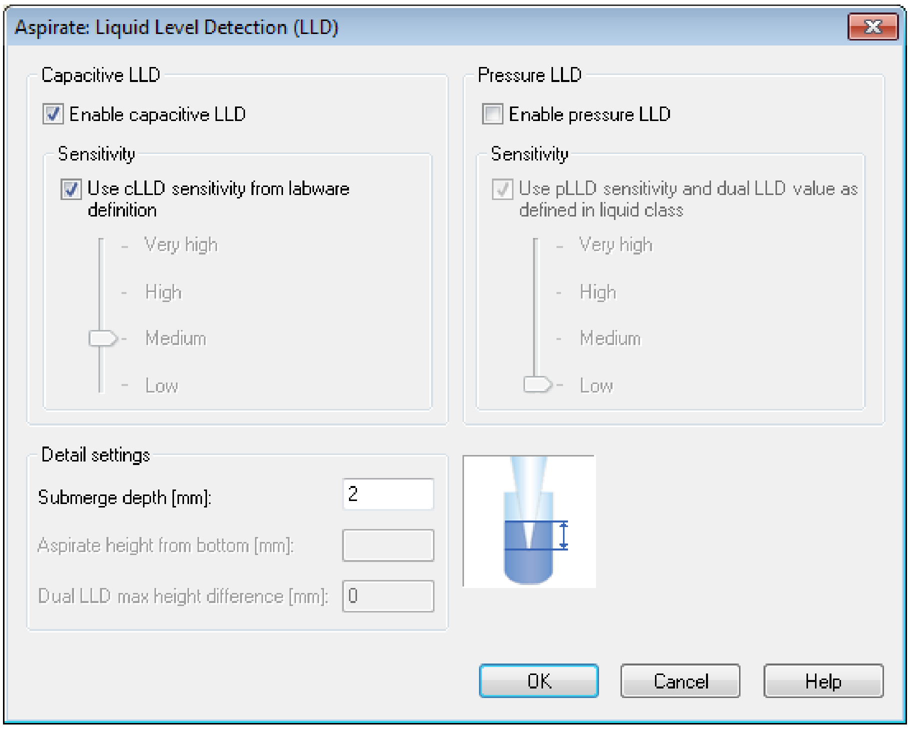


2. Select the appropriate tip/needle handling which is the same for all dispense modes:
3. First, select the tip sequence - where the tips are to be picked up from (here, an array can also be selected). The tip type is automatically retrieved from the sequence. If the tip sequence is used up, it will be automatically reloaded. As an additional feature, the tip sequence can be reduced during a reload.

4. **Tip/needle handling** can be specified, too (some items may not be selected, depending on the pipetting mode):
   - Replace tips “**After each dispense**“ (if multiple dispenses are needed to transfer e.g. 900 μl with a standard tip of 300 μl),
   - Only for replicas (and pooling) the option “**After transferred volume**“ becomes active. This allows the use of fresh tips even for multiple aspirations and multiple dispense cycles being performed with the same sample.
   - “**After each sample**“ (multiple dispenses of the same sample are done with the same tip),
   - With “**One tip set**“, **or**:
   - Without tip handling, “**Never**“. In this case, Microlab instrument will have to pick up tips prior to the pipette step and eject them after pipetting, using single steps.
5. A **tip counter** can be specified, which enables the user to start with a set of tips, partly used in previous runs at the correct position. To read a tip counter, it can be specified within the Smart Step “**Load**” (see corresponding section). A tip counter is specified by a name (e.g., “**cntTips**“). Be reminded to enclose the name in quotation marks.
6. Within the “**Pipette**” Smart Step, the current position of the tip sequence will continuously be stored under the name of the tip counter. Note that a tip counter has to be initialized if it is to be read by the “**Load**” Smart Step within the next run.
7. Click the **[Next >]** Button. The next page of the pipetting wizard will be displayed.

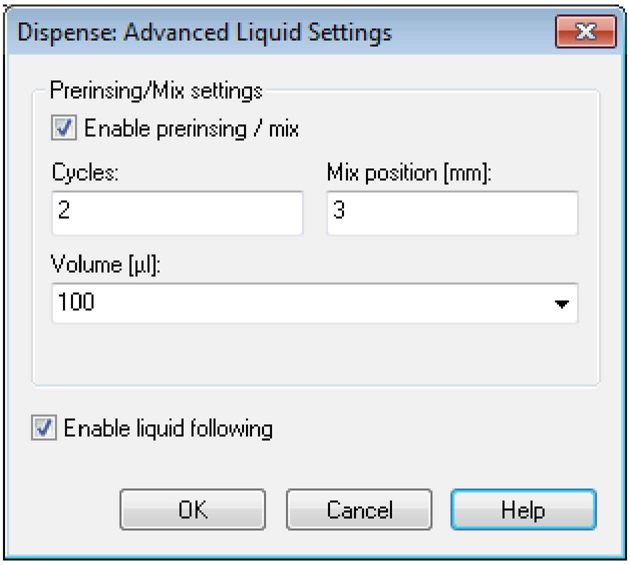


8. Define the dispense mode and the liquid class.
9. Define the **aspiration and dispense (liquid handling)** parameters used in the step.
10. The "**Aspirate the complete requested volume without an error**" Checkbox enables aspiration of the residual liquid from a container (specify a volume larger than the expected residual volume within the container) without an error, even if there is not enough liquid available.
    - For dispense, select the "**Surface**" or "**Jet**"Dispense Mode.
    - For aspiration and dispense, specify the LLD (Liquid Level Detection) settings by clicking the **[LLD]** Button. Below is how the "**Aspirate LLD**" Screen should look like.

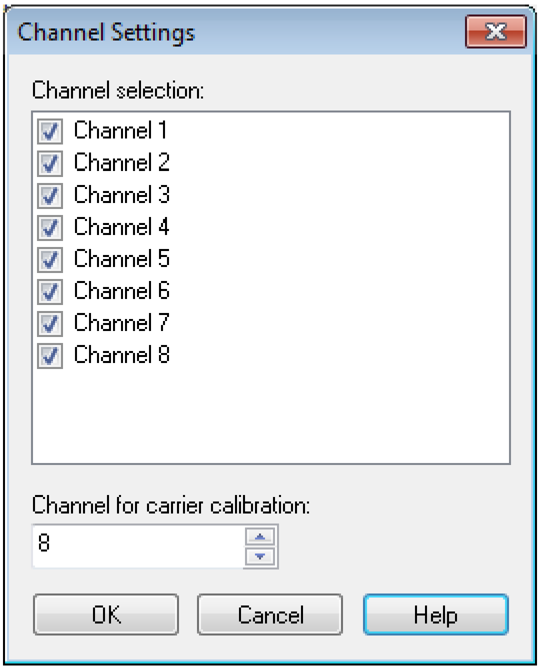


11. Within the LLD dialog, capacitance, pressure, both LLD types (or none) may be selected.
    - A sensitivity setting is necessary. Either use the predefined settings from the Labware or override these settings with Very high / High / Medium or Low.
    - If no LLD is used in aspiration and jet dispensing cannot work with LLD, the height from the bottom has to be defined.
    - If any LLD is used either for aspiration or dispense, a submerge depth has to be specified. For the parallel use of both LLD types, only selectable for aspiration, a maximum height difference has to be given within which both LLDs have to respond.
12. Click **[OK]**, to close the "**Aspirate LLD**" Screen.

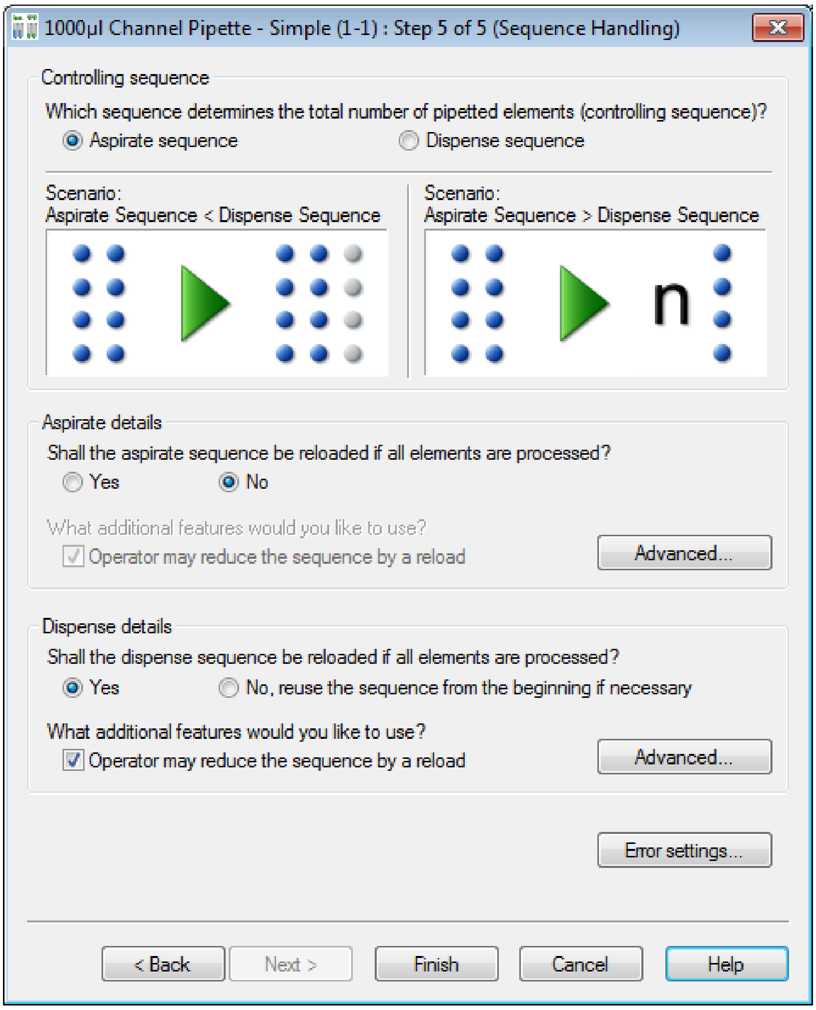
13. In the “**Pipetting Wizard**” Screen, click the **[Advanced...]** Button, to specify the options “**Pre-rinsing**” and “**Liquid following**”. Below is how the “**Dispense**” Screen looks like. Click **[OK]**, to close the “**Dispense**” Screen.



14. Here in the “**Channel Settings**” Dialog, the user is allowed to specify a channel pattern manually, as well as the number of the channels which will be used to calibrate (1536-well) Carriers.



15. Click the **[Next >]** Button. The last page of the Pipetting Wizard will be displayed.



16. Select the appropriate **controlling sequence** by activating the radio button. As long as both sequences are of the same length, this selection does not influence the pipetting (if no pipetting error occurs).

Sequence handling for Aliquot procedure

In the case of aliquoting, the dispense sequence is always the controlling sequence.

Sequence Handling in Pipetting Mode "Simple"

Consider a situation where the aspiration sequence comprises 8 tubes, and the dispense sequence is a 96-well plate. What should be done now with the 88 remaining positions within the plate? If the aspiration sequence is controlling, the dispense sequence is reduced to the length of the aspiration sequence and the pipetting stops after dispensing into the first 8 positions of the plate sequence. If the dispense sequence is controlling, the aspiration sequence will be repeated until it reaches the length of the dispense sequence. This results in multiple transfers (12 for an 8-channel instrument) from the same 8 tubes to fill the complete plate. Then the (controlling) dispense sequence is finished and pipetting stops.

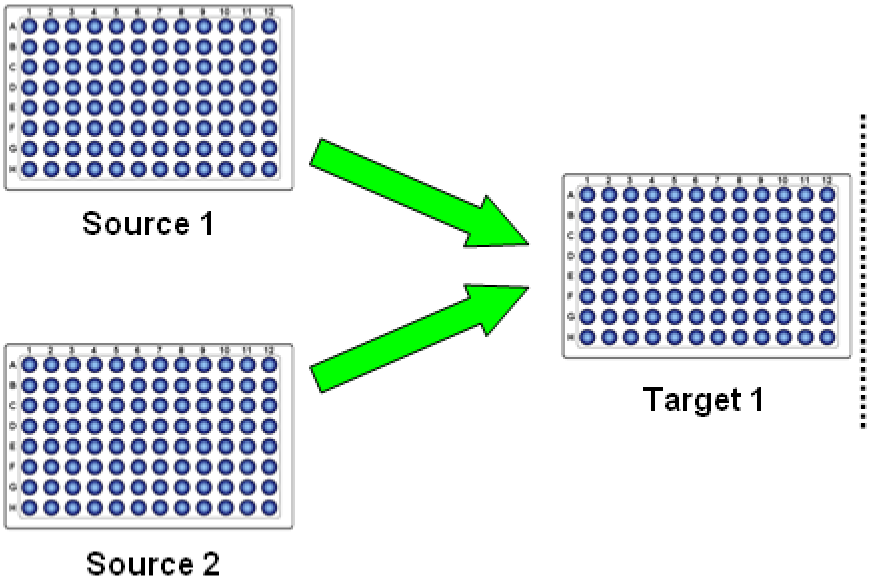
NOTE

*The "**Pipette**" Smart Step always equalizes the length of aspiration and dispense sequences.*

The following table provides an overview of the different situations and examples.

	Aspiration sequence is GREATER THAN dispense sequence	**Aspiration sequence is SMALLER THAN dispense sequence**
Aspiration Controls	The dispense sequence is REPEATED to the length of aspiration sequence (see Case 1) *Examples for use:* - Copy 2 plates into 1 (after another) - With reloading of the dispense sequence: Transfer available tubes to as many plates as needed.	The dispense sequence is REDUCED to the length of the aspiration sequence (see Case 3) *Example for use:* - Transfer a variable number (<96) of tubes into a 96-well plate (left partly filled).
Dispensation Controls	The aspiration sequence is REDUCED to the length of the dispense sequence (see Case 2) *Examples for use:* - Transfers from many tubes to a plate and stops when plate is filled. - With reloading of plate sequences: Automatically fill as many plates as needed with liquid from tubes.	The aspiration sequence is REPEATED to the length of dispense sequence (see Case 4) *Examples for use:* - Transfer reagent from a container (or, e.g., 8 tubes) to a complete plate. - Copy tubes repeatedly into plate. - With reload of aspiration sequence: Automatically reload tubes until plate is filled.

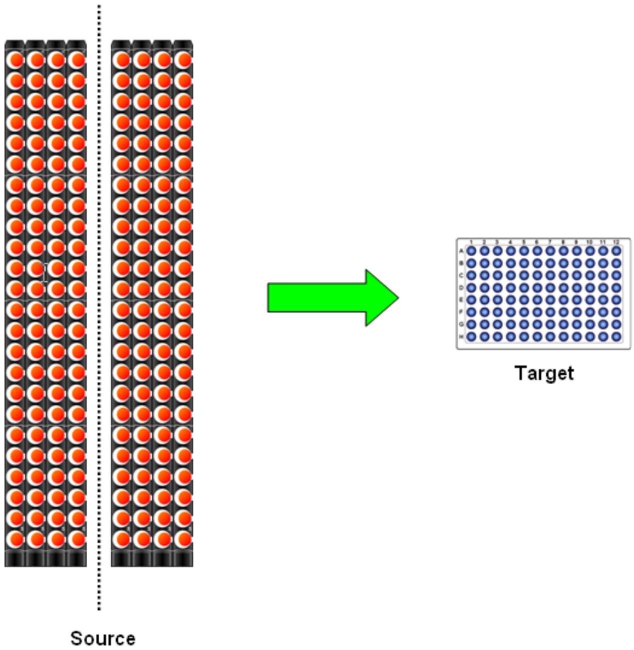
Case 1: Aspiration sequence > dispense sequence, controlled by the aspiration sequence.



The target sequence is repeated to match the length of the source sequence.

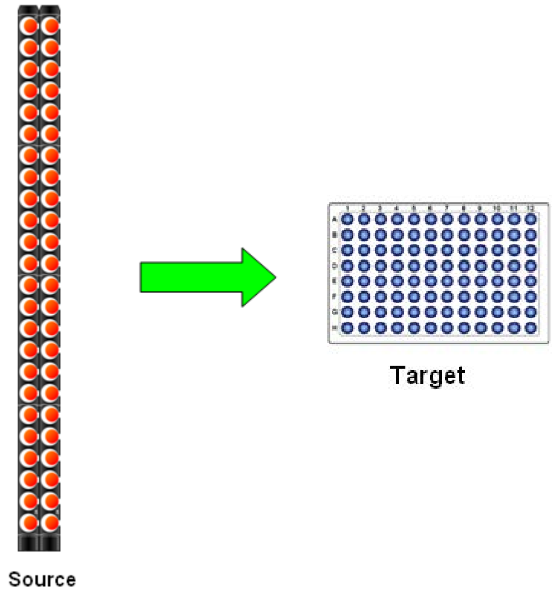
Another example for this case: All Tubes to Plate(s). This is a reloadable plate sequence used to load all tubes to as many plates as needed.

Case 2: Aspiration sequence > dispense sequence. This case is controlled by the dispense sequence.



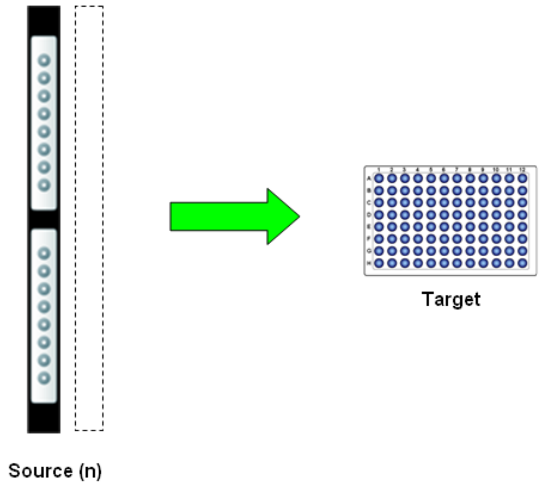
The source sequence is cut to match the length of the target sequence and stops when the plate is full (prepare plate for assay).

Case 3: Dispense sequence > aspiration sequence this case is controlled by the aspiration sequence



The target sequence is cut to match the length of the source sequence and stops when the tubes are processed (how many samples today?). If $L(S_{asp})>96$, plate is filled again.

Case 4: Dispense sequence > aspiration sequence. This case is controlled by the dispense sequence.



The source sequence is repeated n-times to match the length of the target sequence and stops when the plate is full (distribute buffer, reagents...). Also used for aliquoting.

13.6 Sequence Handling in Pipetting Mode Pooling and Replica

In these two cases, the aspiration sequence should be selected as controlling.

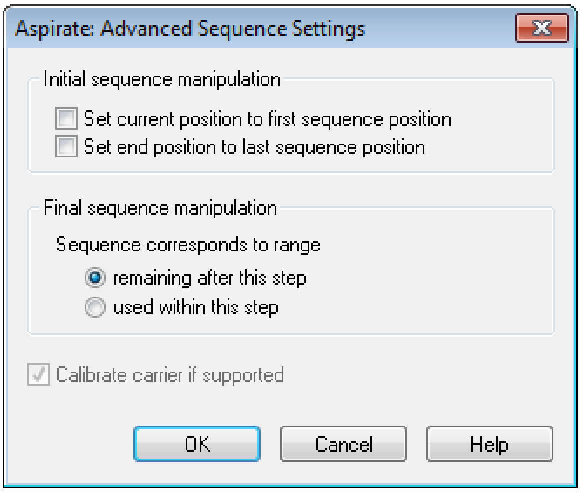
It is possible to choose how to reload a sequence:

- If during pipetting a sequence is used up (no more tubes are left to aspirate from), the system prompts for a new Carrier of tubes to be loaded.
- If the "**Operator may reduce the sequence by a reload**" Checkbox is ticked, the user has the option of reducing the newly loaded sequence at run time. This is especially helpful if the exact amount of sample tubes varies from run-to-run (or from Tube Carrier to Tube Carrier).

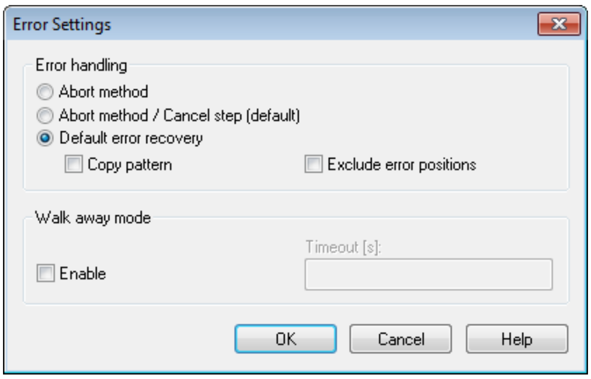
Advanced Buttons

Sequence manipulations can be made (separately for aspiration and dispense sequences) by means of the dialogs invoked by the **[Advanced...]** Buttons.

- If the Smart Step is going to work with sequences that have been used within this method from the preceding steps, the sequence counters can be reset. This is done by using the two checkboxes for initial sequence manipulation.
- If other steps are following this Smart Step, the status of the sequence which is passed back can be defined by the **[Final sequence manipulation]** Radio Button. Below is an example of the advanced dialog for "**Aspirate**" Sequence settings.

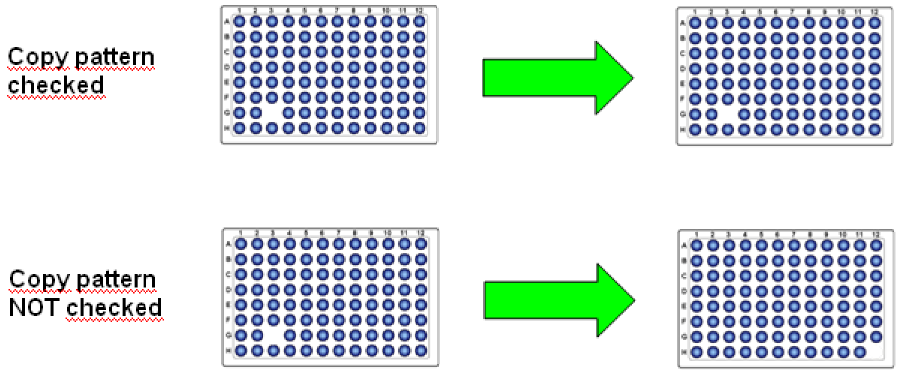


The "**Error Settings**" Dialog allows specifying an error handling approach. The choices are listed on the next page.



In case of an error, the following choices are:

- **Abort** the method (only abort button on run time error dialog)
- Offer choices (in run time) whether to **abort or cancel** (continue with user-defined error handling if programmed, otherwise abort)
- Use the fixed **default error recovery** (recommended) pre-programmed for the Smart Step. In this case two choices can be made
- “**Copy pattern**”, means that in case of an error on aspiration, the corresponding well of the dispense sequence will be left out (the pattern is kept). If “**Copy pattern**“ is not checked, the dispense sequence positions will all be pipetted, leaving a well not pipetted at the end of the sequence.



- “**Exclude error positions**”, means exclude the erroneous positions from aspiration and dispensation sequences. If this option is enabled and an error occurs during an aspirate or dispense step, the erroneous position will be excluded from the sequence by removing the corresponding element from the aspirate or dispense sequence. The next time these sequences are used, the erroneous well will not be pipetted.
- In addition, **Walk-away mode** can be enabled. If this checkbox is ticked, a timeout has to be specified after which the error dialog on run time will automatically close down and continue with the selected error handling: abort, cancel, or default error recovery.

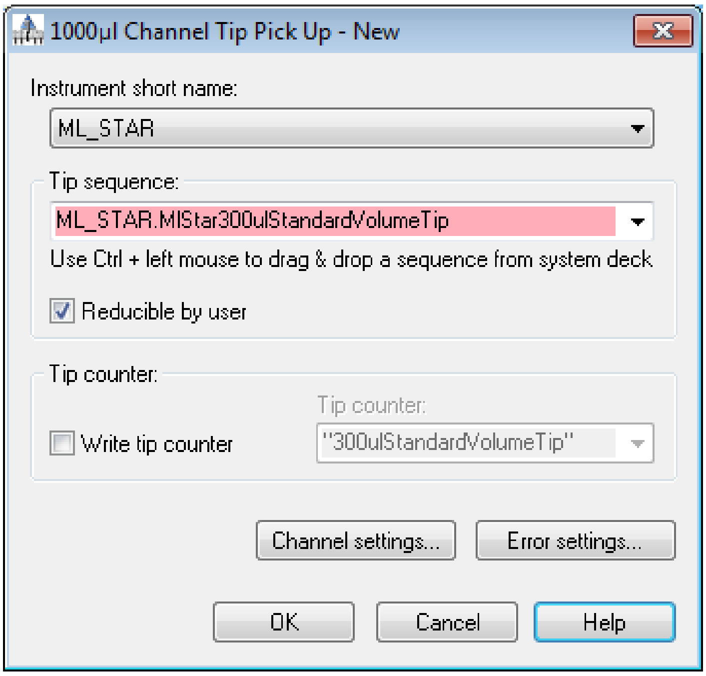
NOTE

*Exception: If the option “**Default error recovery**” is chosen and an error occurs for which the default is set to abort (e.g. a hardware error), the walk-away mode is left on and the dialog waits for user interaction.*

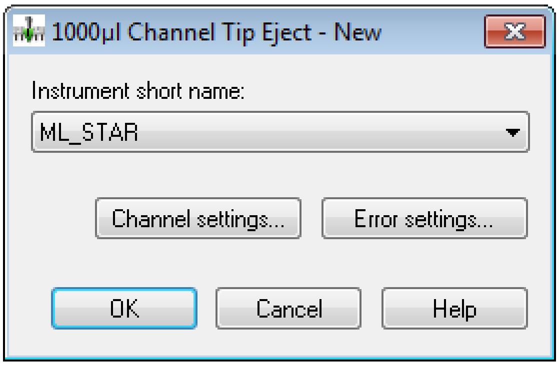
13.6.1 Tip Pick Up and Tip Eject Smart Steps

These two Smart Steps allow picking up tips from a rack and then disposing them in the tip waste after use.

Below is the dialog for the "**Tip Pick Up**" Step.



The dialog is to specify a tip sequence. It is also possible to specify the tip counter. Below is the dialog for the "**Tip Eject**" Step.



For "**Channel Settings**" and "**Error Settings**", see the description of the previous Smart Steps found in Section 3.4.2 Smart Steps Settings.

13.7 Power Steps

Power Steps are guided walkthroughs to perform common pipetting tasks. Each Power Step includes options to adjust the pipetting settings, configure automatic error handling, or generate a pipetting report. They can be used in combination with other programming steps within a method.

Available Power Steps:

- Transfer Samples
- Add Reagent
- Serial Dilution
- Replicates
- Hit Picking

The Power Steps are compatible with these instruments:

- Microlab STAR
- Microlab VANTAGE (VENUS on VANTAGE)
- Microlab NIMBUS

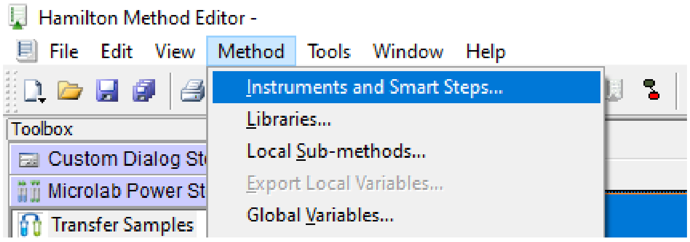
and can be used with any of these pipetting tools:

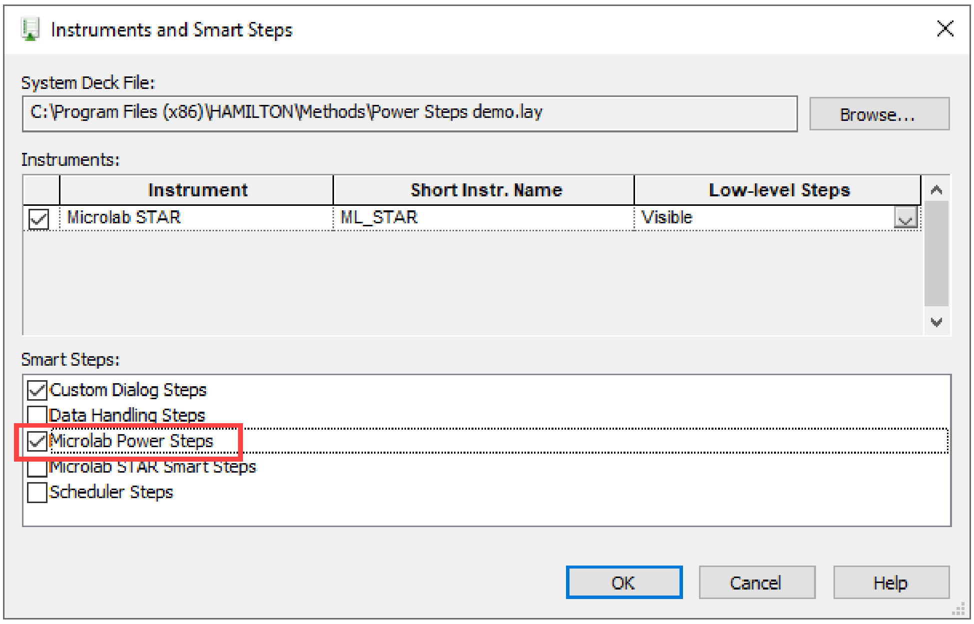
- 1 mL Channels
- 5 mL Channels
- Multi-Probe Head 96
- Multi-Probe Head 384

Keep in mind that all Power Steps have on-screen tooltips for the different screens. Mouse over a component at any time to see its associated Help text.

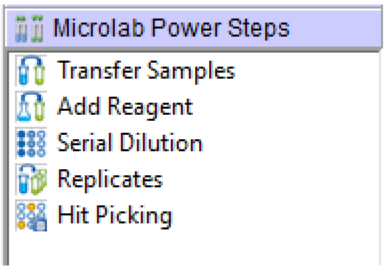
13.7.1 Setup

To display the Power Steps in the Method Editor Toolbox, with a method opened in the Method Editor, go to Method > Instruments and Smart Steps and check the Power Steps option





After closing the dialog, the Power Steps group will be displayed in the Toolbox.

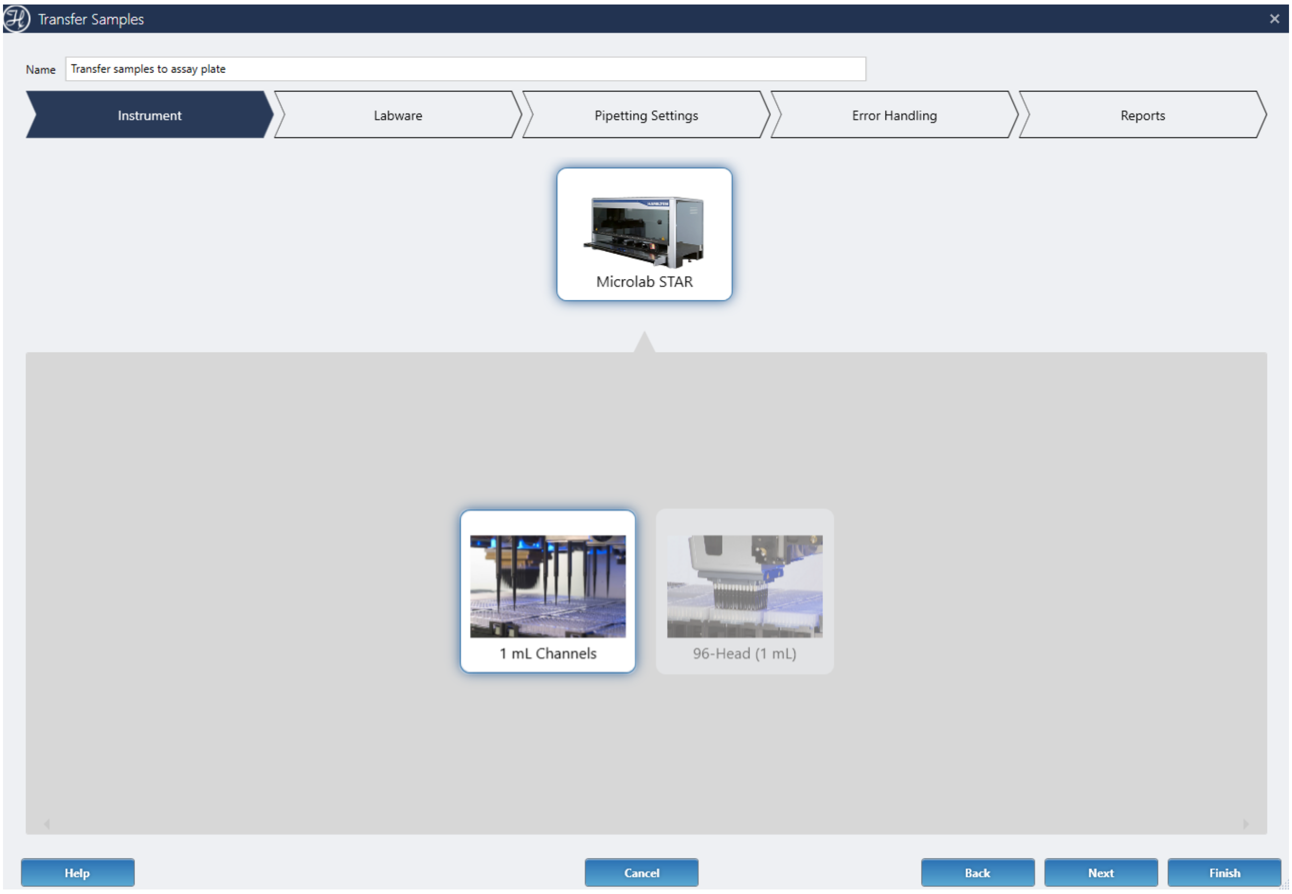


13.7.2 Transfer Samples

The Transfer Samples Power Step sets up a single transfer of a given set of sources to a set of targets.

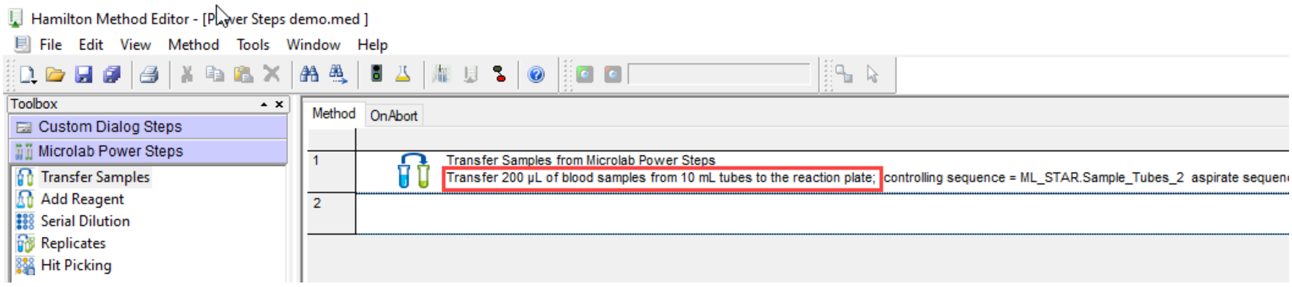
13.7.2.1 Instrument

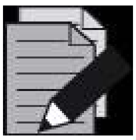
The Instrument screen displays the pipetting tools and instruments available to perform the pipetting. These will vary, depending on the instruments present in the system deck used in the method and their configurations.



Note that the instrument screen will only be displayed if multiple instruments or pipetting tools are available; otherwise, it will be skipped automatically.

The ‘Name’ field at the top can be customized with any descriptive text of the transfer. This will help in identifying the step in the method.



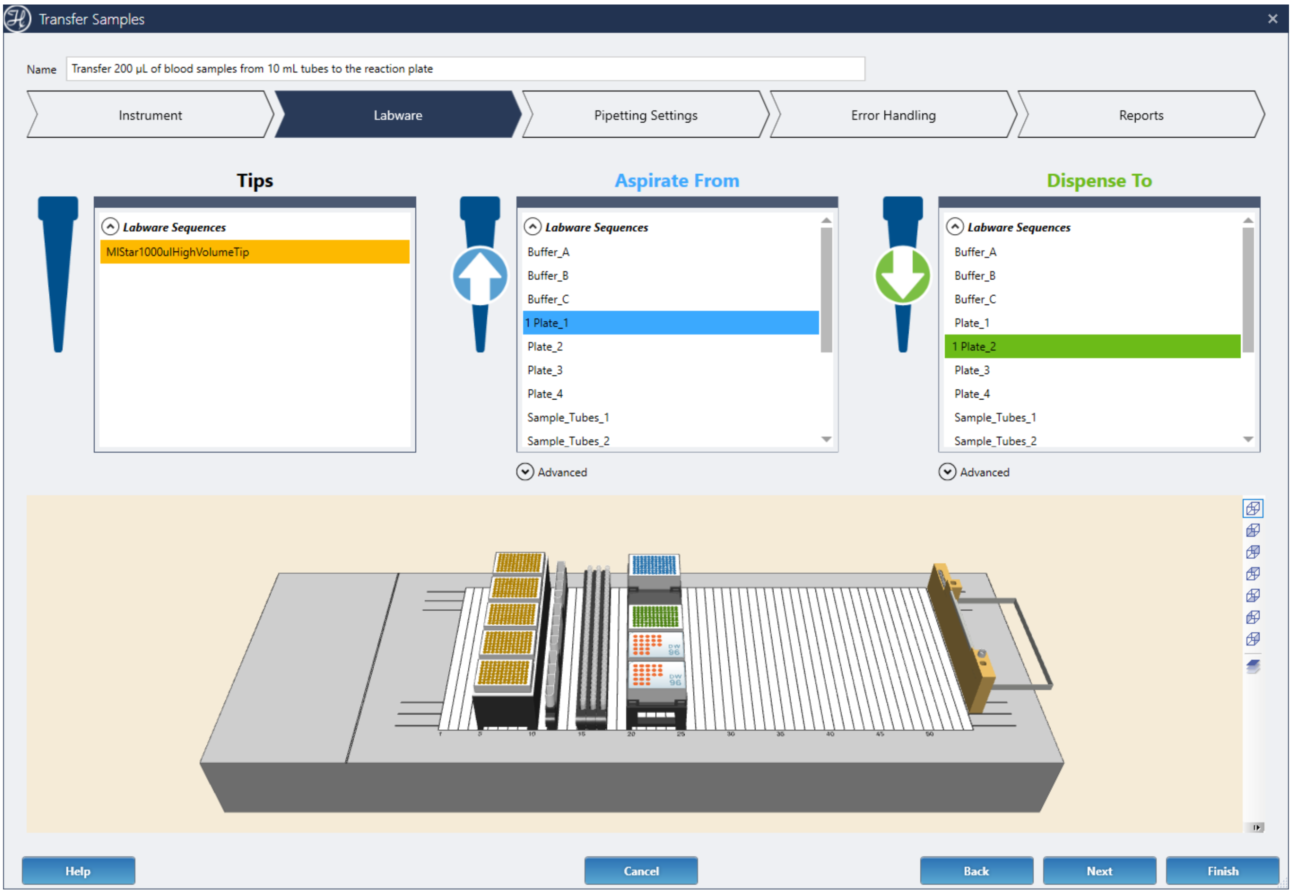


NOTE

Press F8 or go to View > Icon and Description to show the step description in the method.

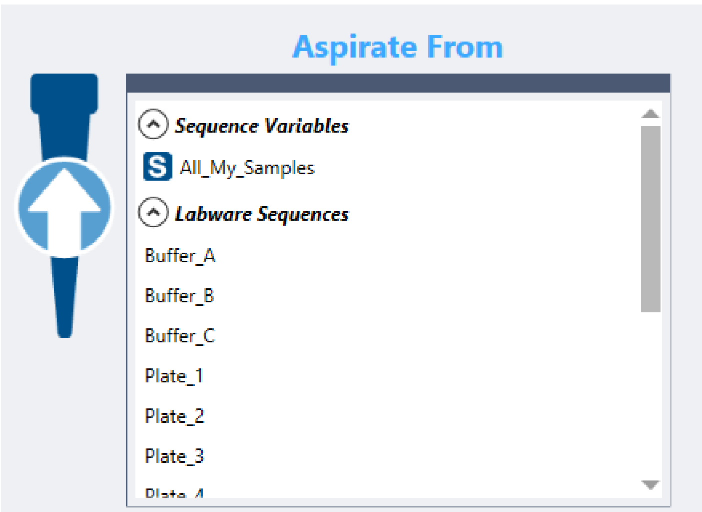
13.7.2.2 Labware

The Labware screen displays the deck layout and the tips, aspirate and dispense sequences used to perform the pipetting.



The lists include two types of sequences to select from:

- **Labware Sequences**:
  sequences previously defined in the Deck Layout Editor
- **Sequence Variables**:
  sequences dynamically created in the method at runtime by other steps. This will only be displayed if your method uses any sequence variables.

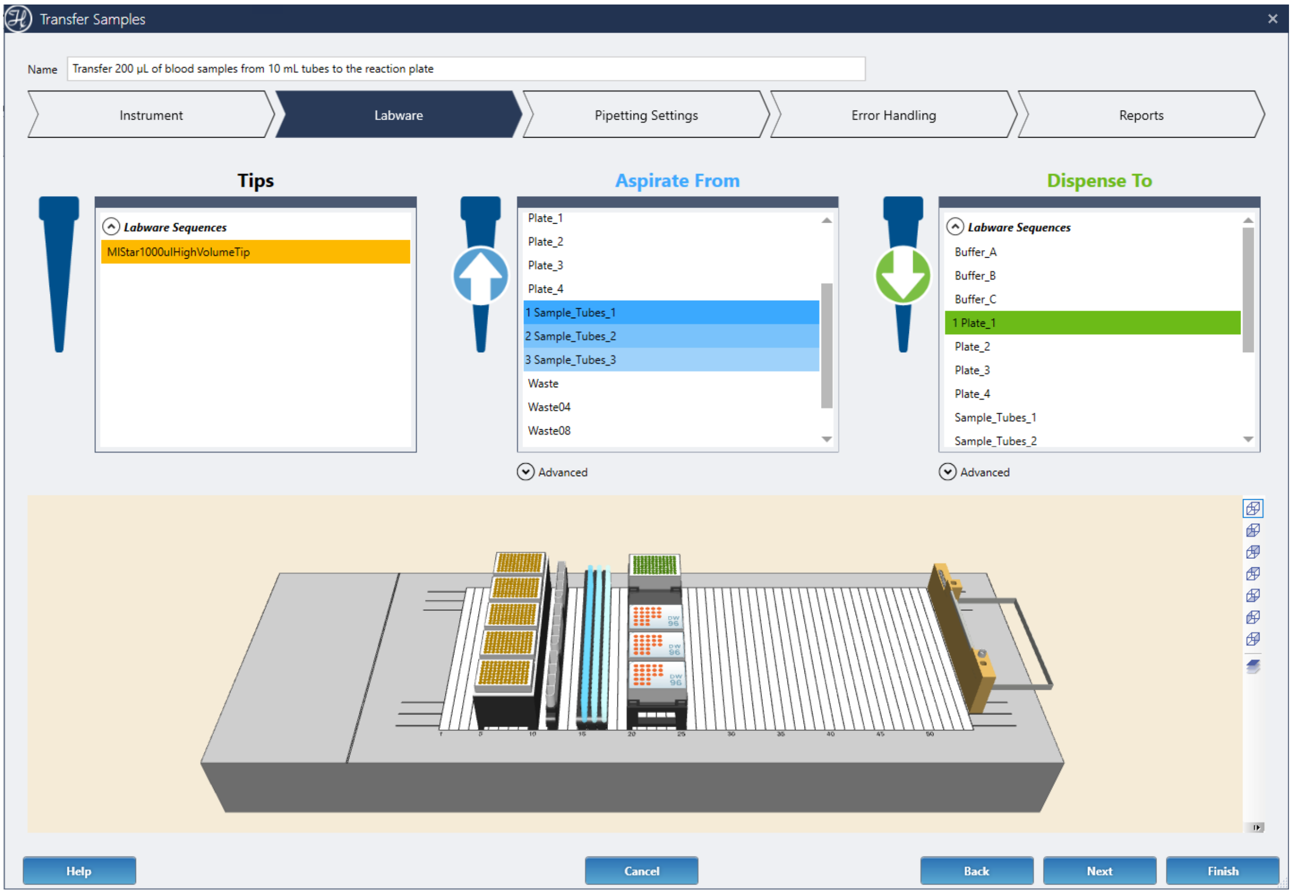


NOTE

Provide meaningful names for your sequences in the Deck Layout Editor to identify them easier in the method steps.

In the ‘Aspirate From’ and ‘Dispense To’ lists, multiple sequences can be selected to complete a set of source or destination positions. When a sequence is clicked a number will appear on the left, indicating the processing order.

By default, all the sequences selected within a list will be combined as a single source or destination sequence. In this example, 3x32 Tube Carriers will be transferred in order to a 96-well plate:



Sample_Tubes_1, position 1 → Plate_1, position 1

Sample_Tubes_1, position 2 → Plate_1, position 2

…

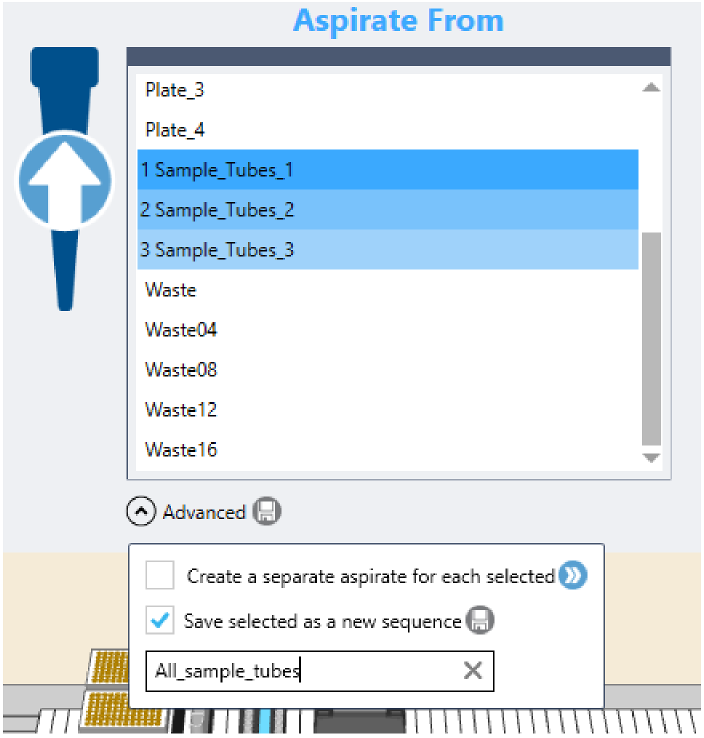
Sample_Tubes_1, position 32 → Plate_1, position 32

Sample_Tubes_2, position 1 → Plate_1, position 33

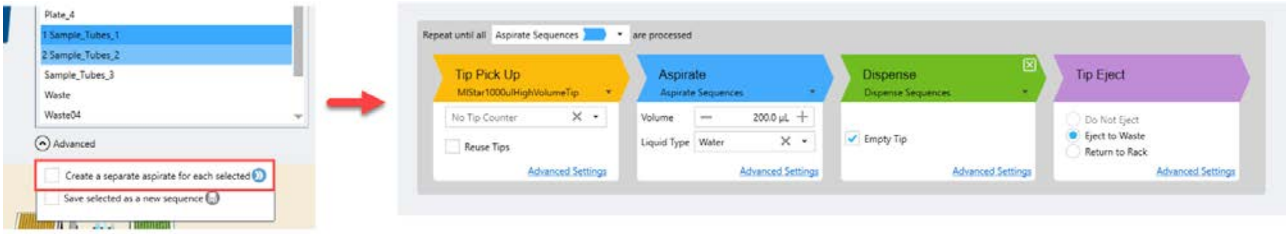
…

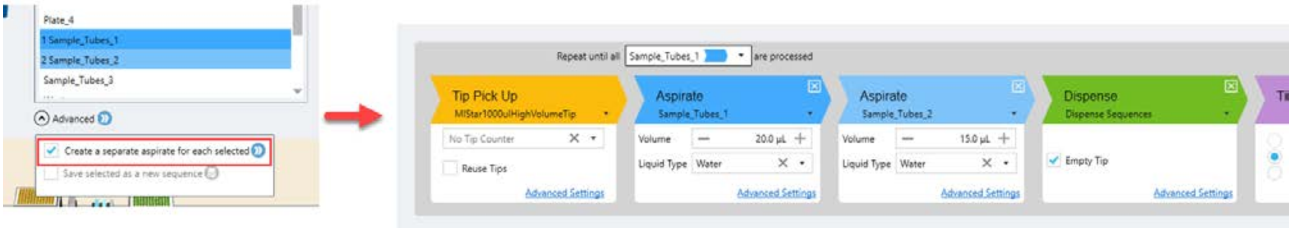
Sample_Tubes_3, position 32 → Plate_1, position 96

Optionally, under the Advanced section, the selected sequences can be saved into a single new sequence that can be used in the method later on.



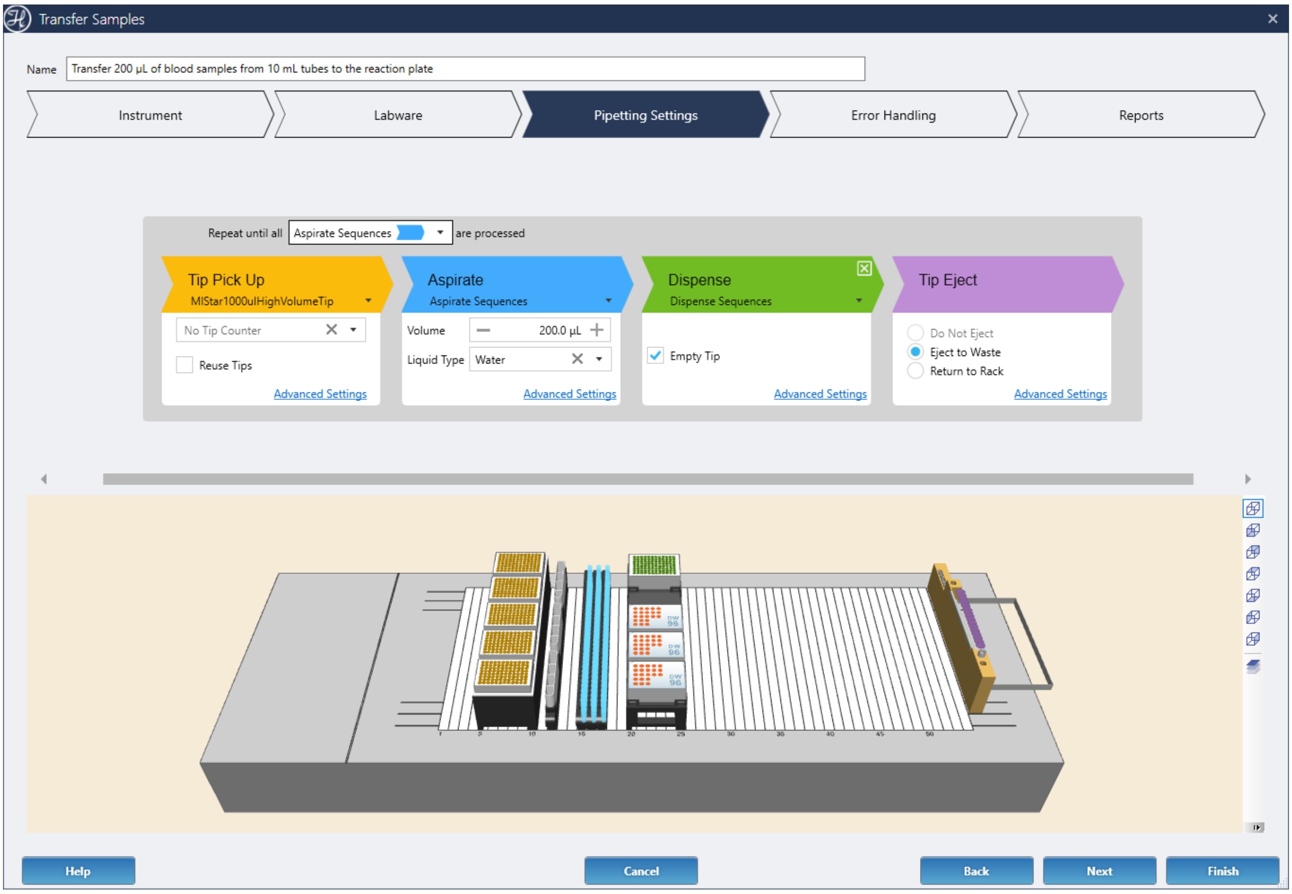
It also provides an option to create a single aspirate or dispense action for each sequence selected in the list, generating individual aspirate or dispense chevrons for each sequence on the Pipetting Settings screen and enabling the option to perform multi-aspirations or multi-dispenses.





13.7.2.3 Pipetting Settings

The Pipetting Settings screen displays a diagram of the pipetting procedure and steps.



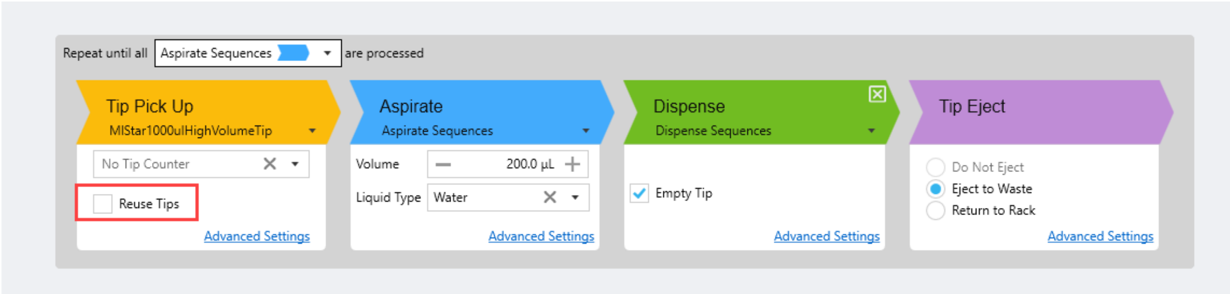
The top gray block represents the steps that will be repeated in a loop until all the indicated positions are processed. By default, this is set to the ‘Aspirate sequences’, which means that the pipetting will finish whenever all the aspirate positions have been transferred to the destination.

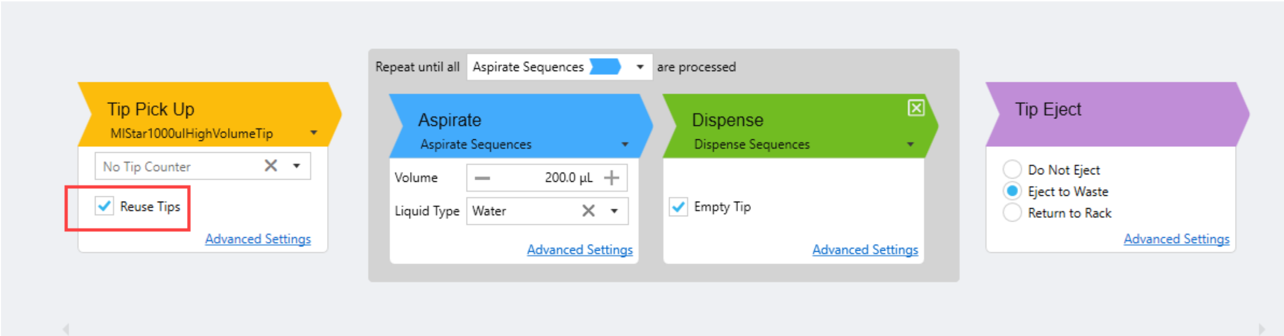
Mouse over any of the colored chevrons to blink its related positions on the deck. If multiple aspirate or dispense chevrons are displayed, they can be rearranged in a custom order for increased flexibility.

Tip Pick Up

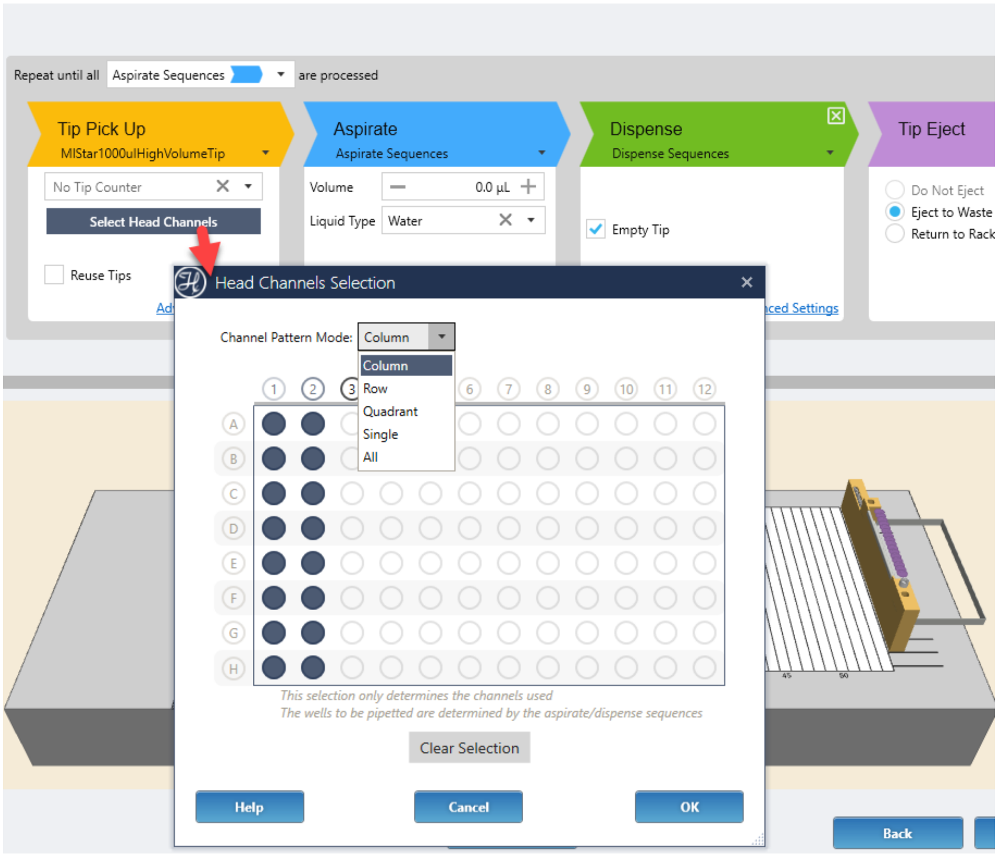
The Tip Pick Up chevron can optionally use a tip counter to consume tips remaining from a previously executed run without manipulating the tip sequence.

By default, it will use one tip per each source aspirated. Check ‘reuse tips’ in order to use the same tips for all pipetting in the entire pipetting of the Power Step and eject tips only once the Power Step is completed.





Note that when a Multi-Probe Head is the selected pipetting device, the individual channel pattern that will be used to aspirate and dispense liquid can be selected. Click the Select Head Channels button in the Tip Pick Up white box. The Head Channels Selection dialog appears.



- Use the Channel Pattern Mode to choose how to select the channels. For example, if Column is selected, the channels can only be edited by column.

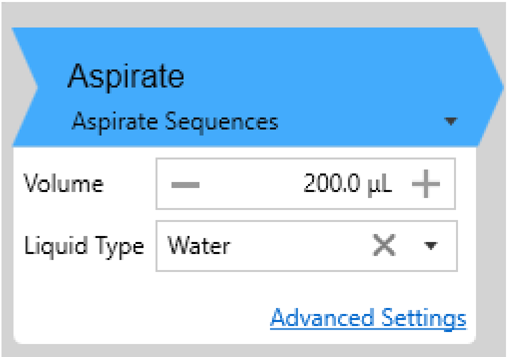
- The numbered or lettered circle surrounded in a bold blue or black ring is the only editable option at a time. Click the bold blue circle to remove a selection. Click the bold black circle to add a selection. Selected channels are always represented by a completely blue circle.

The Advanced settings section configures the advanced error handling options for the tip pick up steps. If enabled, it will override the general tip error handling defined on the Error screen.

Aspirate

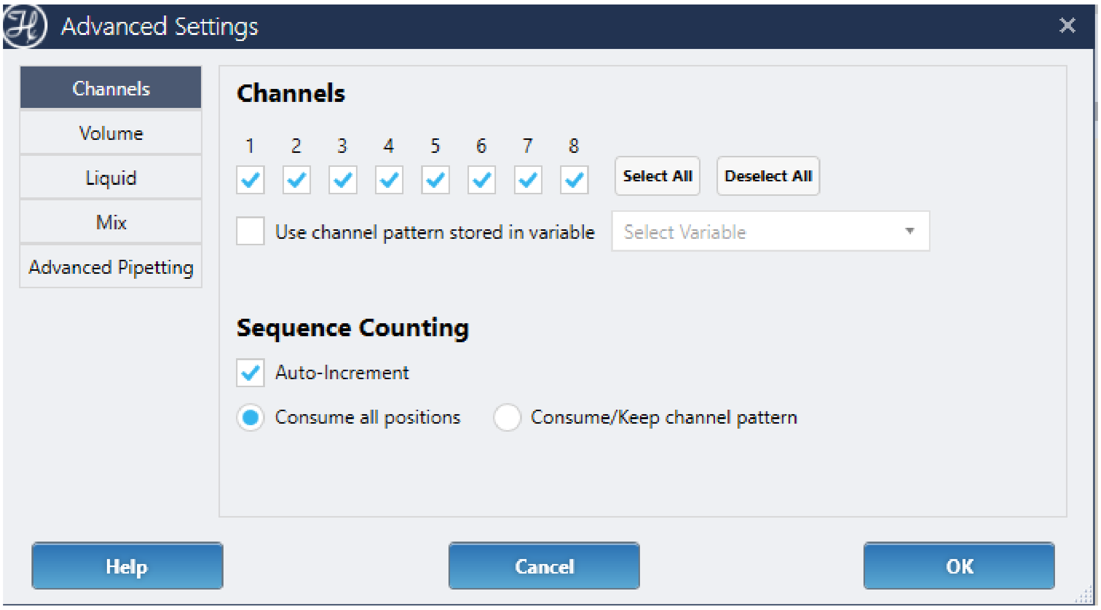
This represents the aspirations performed in every loop of the Power Step. One or more aspirate chevrons will be displayed, depending on the advanced settings set for the 'Aspirate from' in the Labware screen.

Enter the volume to be transferred and the liquid type. Based on this information and the type used, the system will automatically select an appropriate liquid class to optimize the pipetting.



Many other aspirate parameters can be configured in the Advanced Settings section:

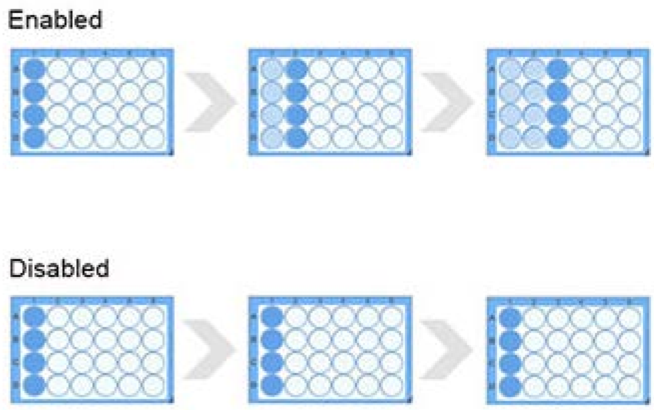
- **Channels**:
  Define the channel pattern to be used for the aspiration and how the sequence positions will be incremented after usage.



Check the sequence auto-increment box to indicate that the pipette will use different wells in every loop of the Power Step.

NOTE

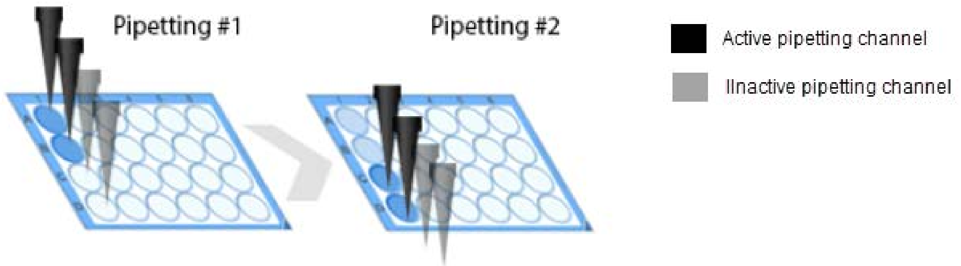
Disable auto-increment if pipetting from a reagent reservoir.



If auto-increment is checked, the sequence positions can be consumed (incremented after usage) in two ways:

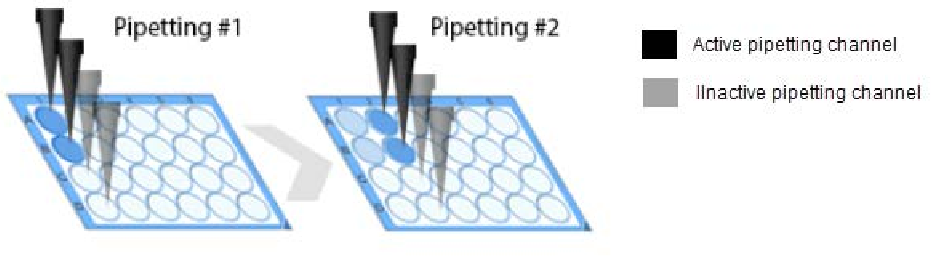
- "Consume all positions":

  All wells are used. No wells under inactive channels are left empty between aspirations or dispenses.

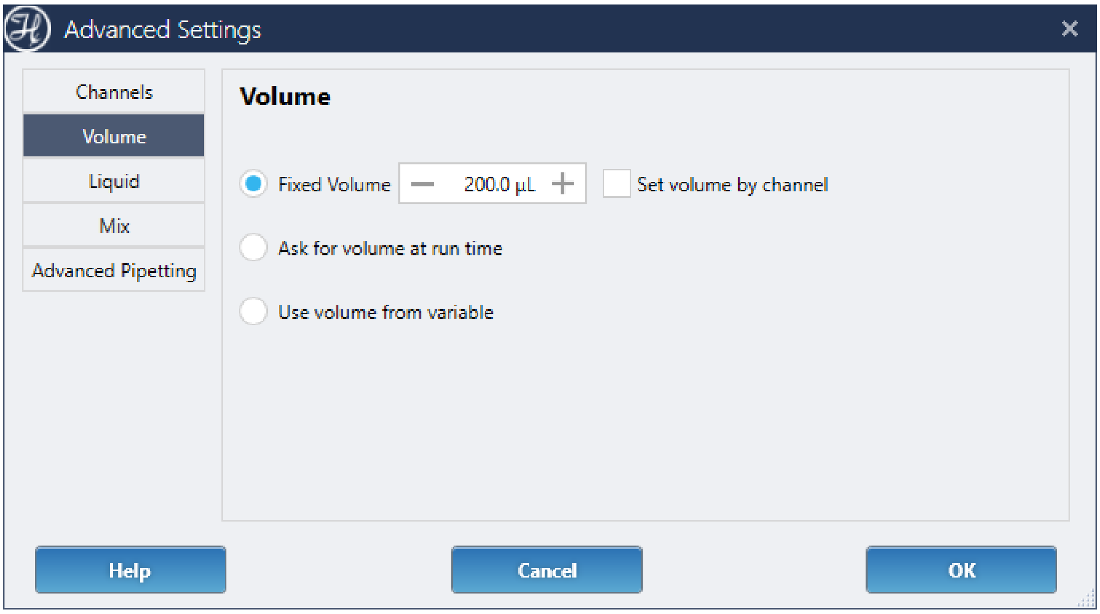


- "Keep channel pattern":

  Only the wells under active channels are used. The wells under inactive channels are left empty between aspirations or dispenses.



- **Volume**:
  Enter a fixed volume, retrieve the value from a variable or array in the method, or display a dialog at runtime to enter the volume.

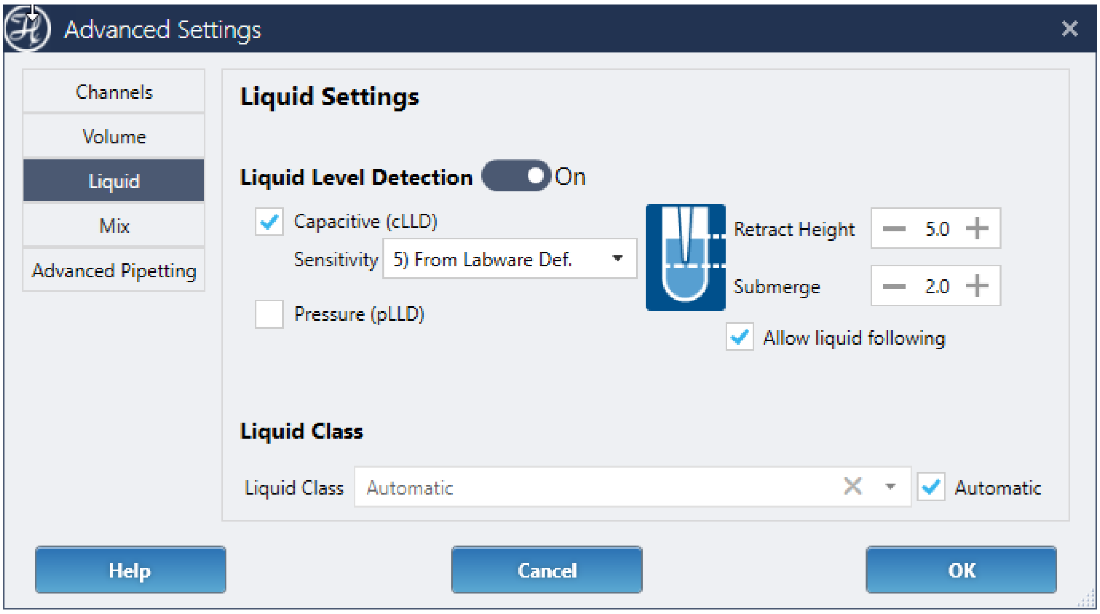


- **Liquid**:
  Define if liquid level detection will be used or if the pipetting will occur at a fixed height above the container bottom.

  The retract height defines the number of mm that the tip will move up in Z after completing the aspiration before aspirating any transport air.

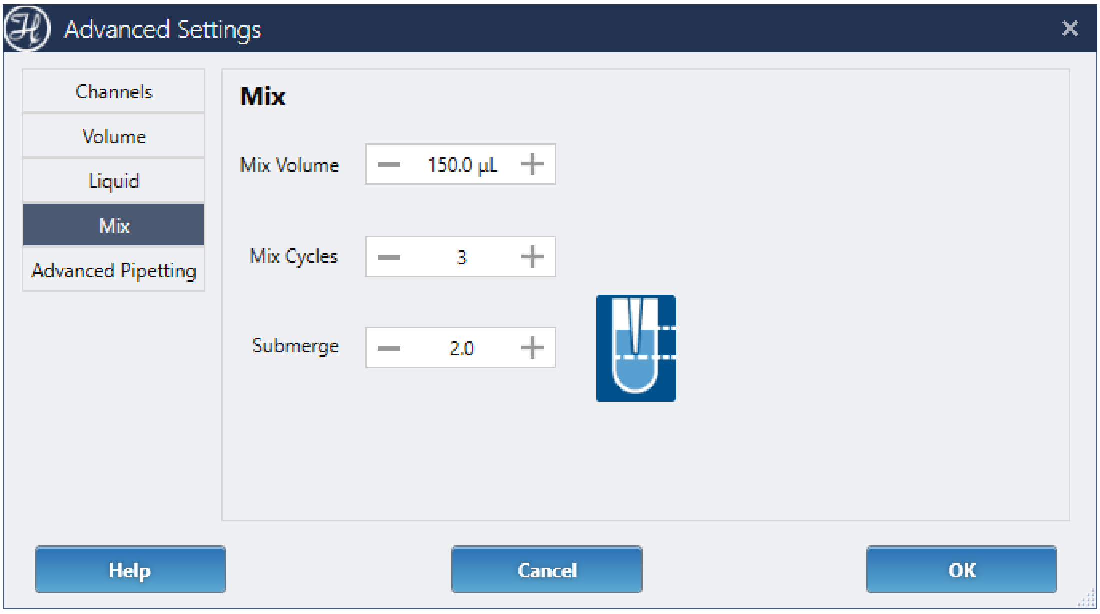
  Pay special attention to the 'Allow Liquid Following' checkbox. If enabled, the tip will travel down the container as the liquid is being aspirated. This is specially recommended when using liquid level detection to maintain the tip submerged in the liquid and avoid aspirating air.

  The liquid class is selected automatically based on the tip type and pipetting settings. A specific liquid class can be selected by unchecking the "Automatic" box.



- **Mix**:
  Define if any mixing will be executed before aspirating the target volume to be transferred. This can be done to resuspend or homogenize the liquid in the container or to condition the tip when pipetting volatile liquids.

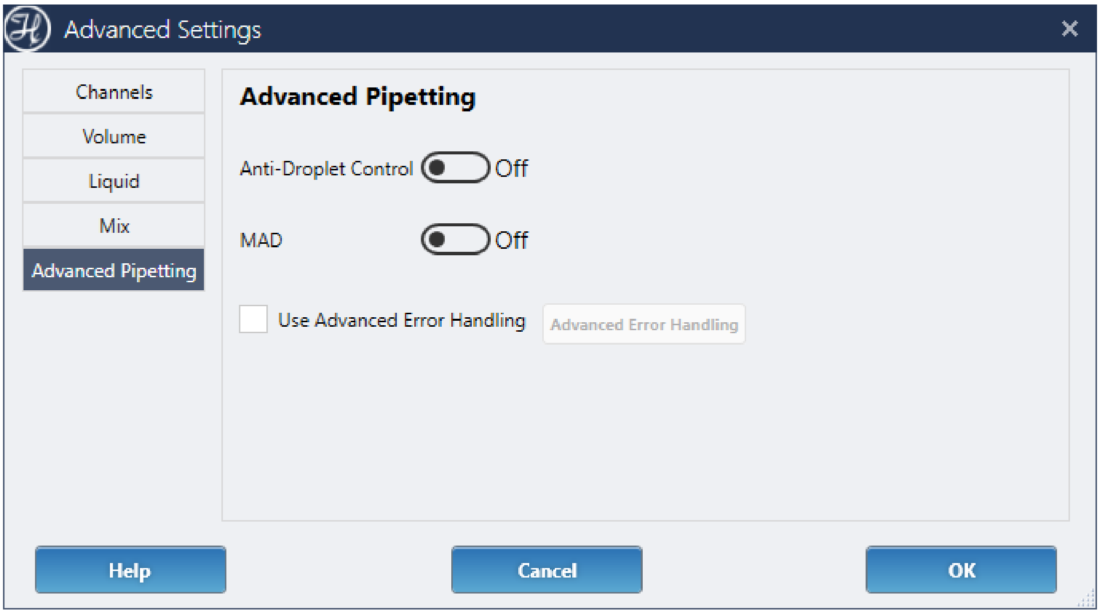
  A single mix cycle includes a full aspiration and dispense of the mix volume indicated. The 'submerge' value indicates the additional tip submerge where the mix will start, and its relationship to the aspiration height set in the Liquid section.



- **Advanced Pipetting**:
  Enable the Anti-Droplet Control feature to pipette volatile liquids without dripping by using the pipetting channel pressure sensor to monitor and compensate for pressure increases caused by evaporation inside the tip.

Enable the Monitored Air Displacement (MAD) feature to track the pressure inside the pipetting channels during the aspiration and detect any irregularities such as air bubbles, foam, clots or lack of liquid in the container.

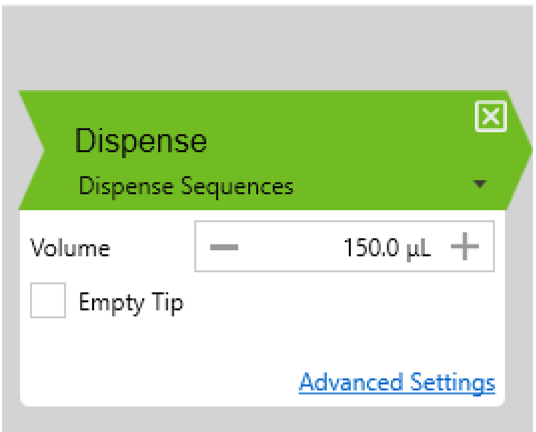
Optionally configure the advanced error handling for the aspiration. If enabled, it will override the general aspirate error handling defined in the Error screen.



Dispense

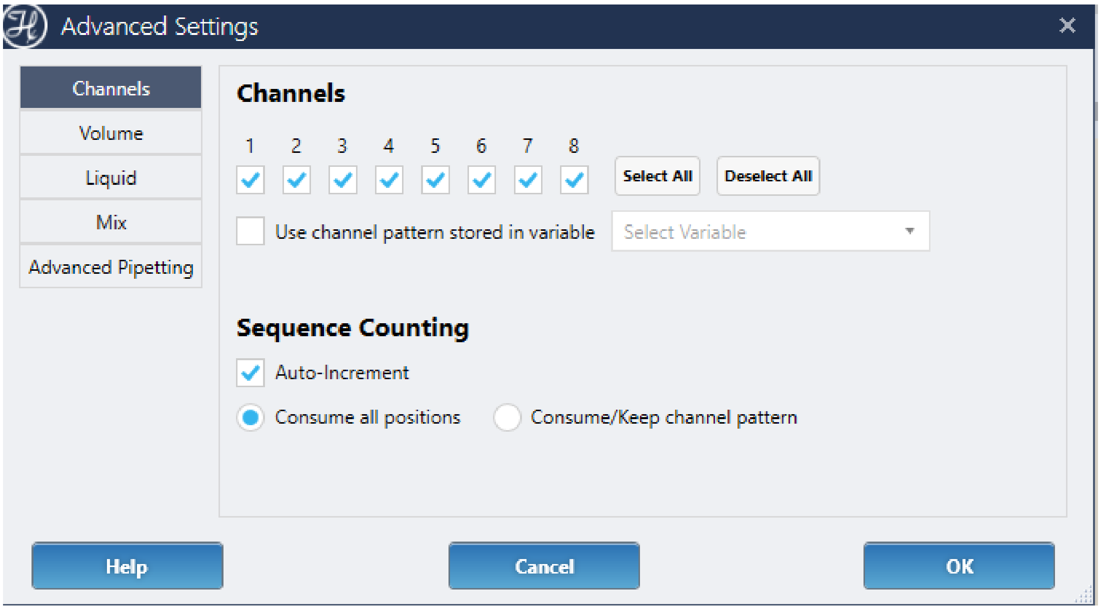
This represents the dispense steps performed in every loop of the Power Step. One or more dispense chevrons will be displayed, depending on the Advanced Settings set for the ‘Dispense To’ on the Labware screen.

By default, all the liquid volume present in the tip will be emptied in the target container, including any blowout air volume. Uncheck the ‘Empty Tip’ box to enter a different volume amount, less or equal to the volume present in the tip.

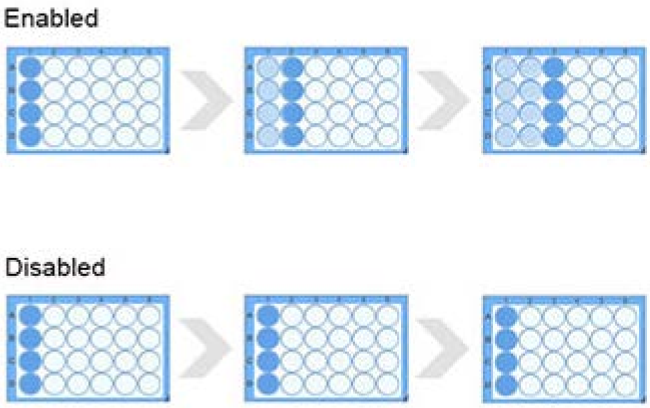


Many other dispense parameters can be configured in the Advanced Settings section:

- **Channels**:
  Define the channel pattern to be used for the dispense and how the sequence positions will be incremented after usage.



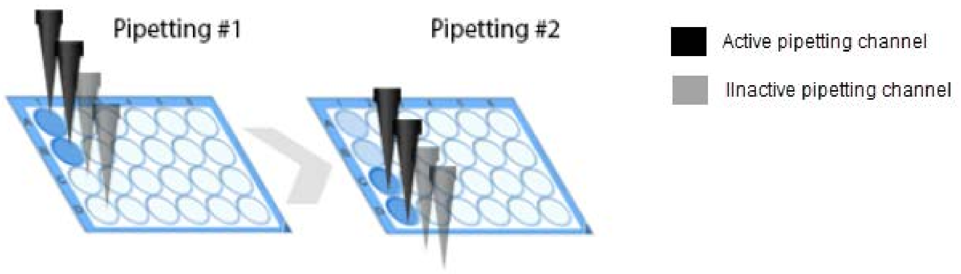
Check the sequence auto-increment box to indicate that the pipette will use different wells in every loop of the Power Step.



If auto-increment is checked, the sequence positions can be consumed (incremented after usage) in two ways:

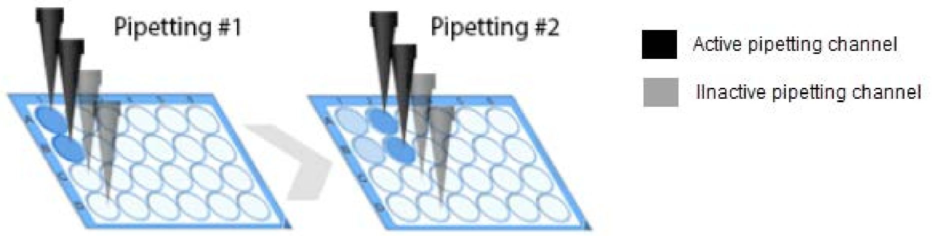
- "Consume all positions":

  All wells are used. No wells under inactive channels are left empty between aspirations or dispenses.

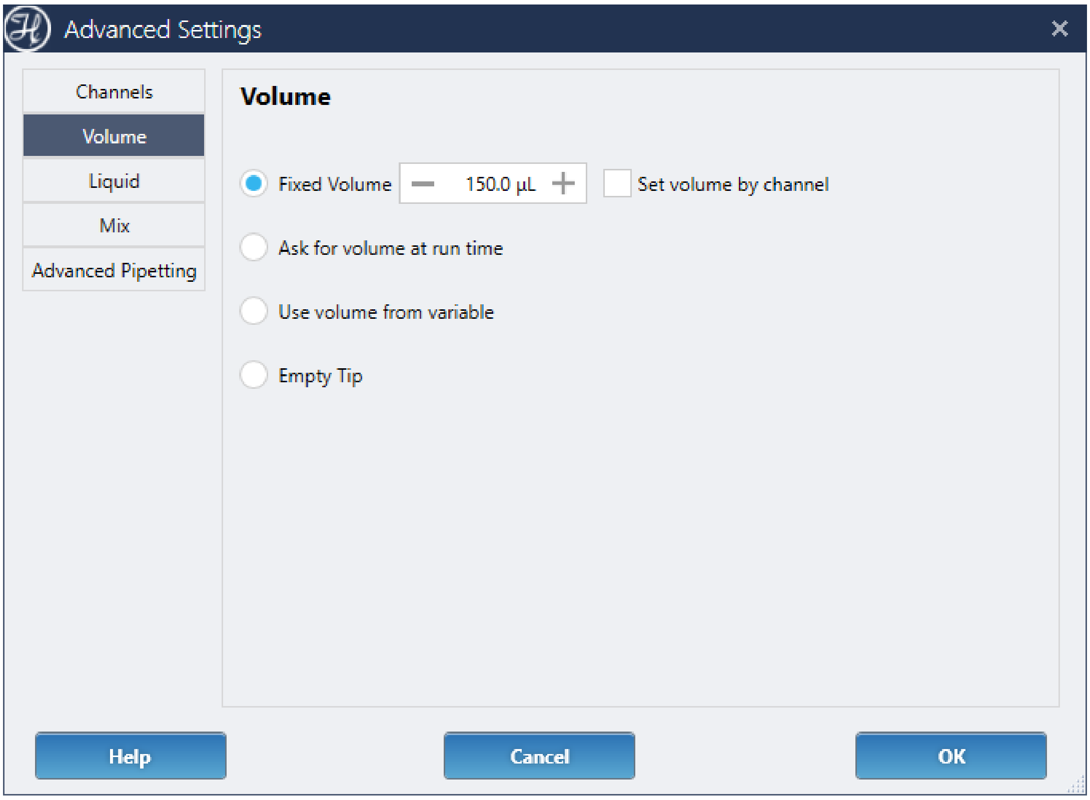


- "Keep channel pattern":

  Only the wells under active channels are used. The wells under inactive channels are left empty between aspirations or dispenses.



- **Volume**:
  Enter a fixed volume, retrieve the value from a variable or array in the method, display a dialog at runtime to enter the volume or empty the tip.



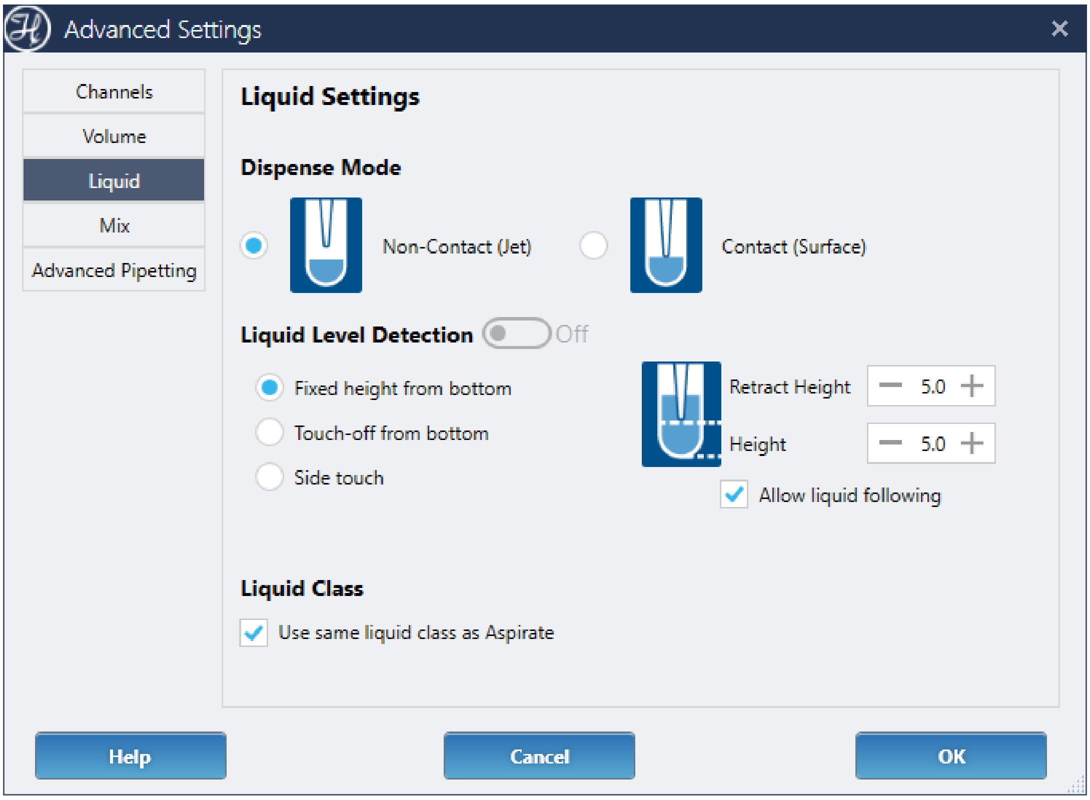
- **Liquid**:
  Define if dispense will be performed in Jet (without contact with the liquid surface) or in Surface mode.

If dispensing in Surface mode, define if liquid level detection will be used or if the pipetting will occur at a fixed height above the container bottom or on the side of the well (side touch). If selecting 'Side Touch', keep in mind that this height is defined within the container definition of the corresponding labware.

The retract height defines the number of mm that the tip will move up in Z after completing the dispense before aspirating any transport air.

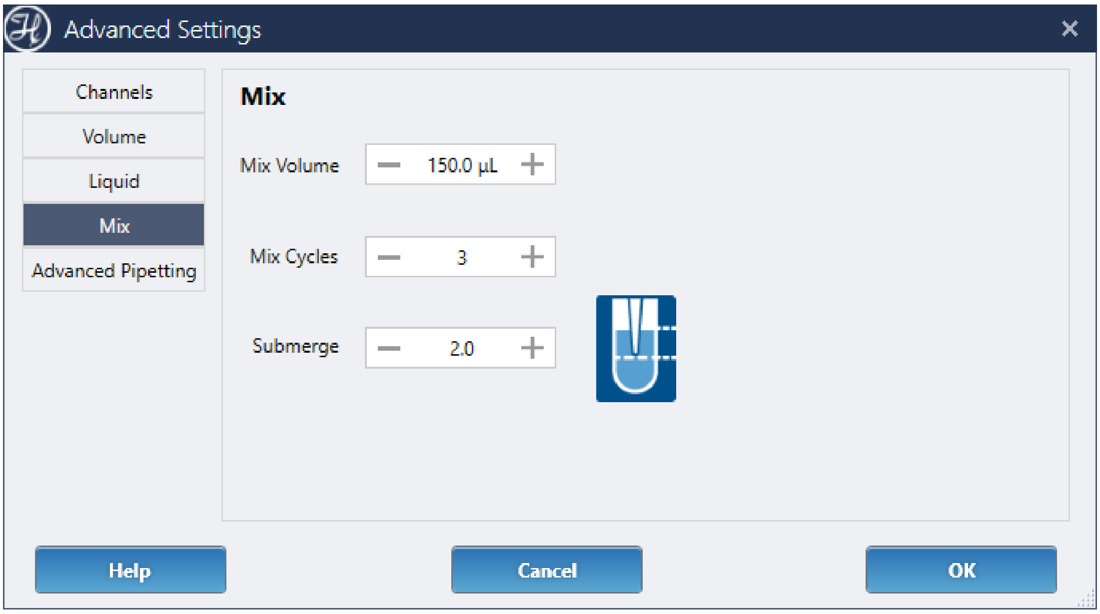
Pay special attention to the 'Allow liquid Following' checkbox. If enabled, the tip will travel up the container as the liquid is being dispensed.

The liquid class is selected automatically, based on the one used in the aspiration. If the process contains multiple aspiration chevrons, the dispense step will use the liquid class from the first aspiration chevron. A specific liquid class can be selected by unchecking the 'Automatic' box.

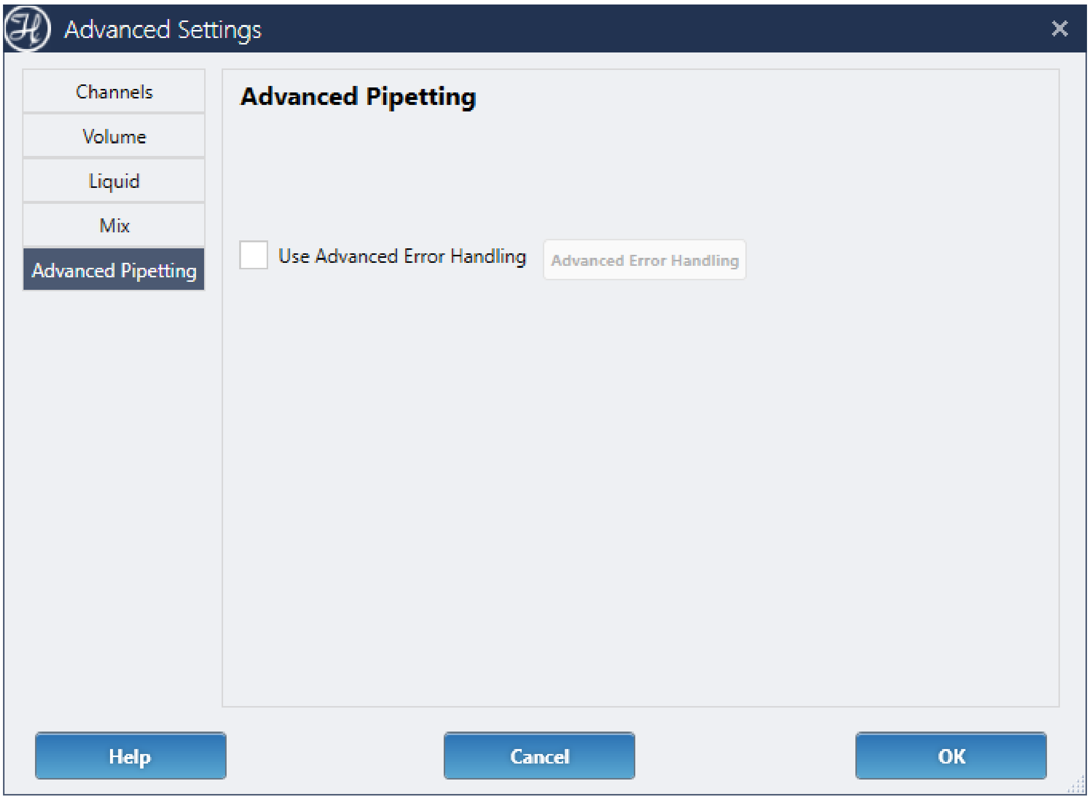


- **Mix**:
  Define if any mixing will be executed after a surface dispense is completed. The mix settings will be ignored by the system when dispensing in jet mode (non-contact).

  A single mix cycle includes a full aspiration and dispense of the mix volume indicated. The 'submerge' value indicates the additional tip submerge where the mix will start, and its relationship to the dispense height set in the Liquid section.



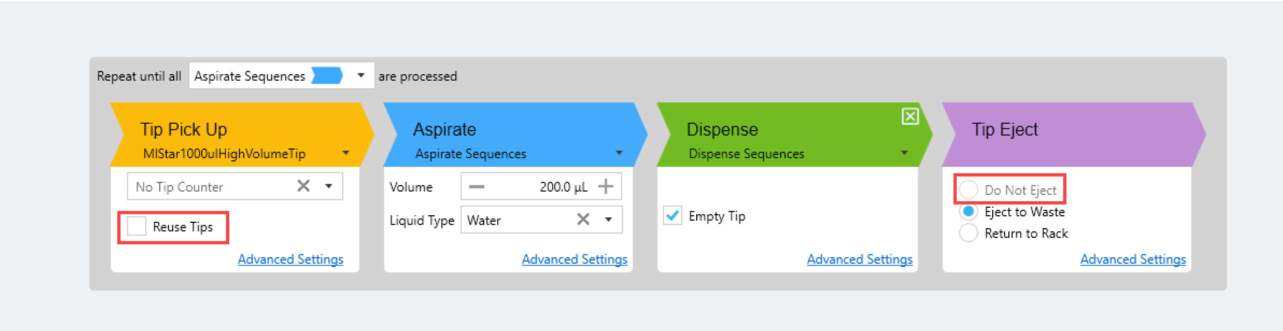
- **Advanced Pipetting**:
  Configure the advanced error handling for the dispense step. If enabled, it will override the general dispense error handling defined on the Error screen.

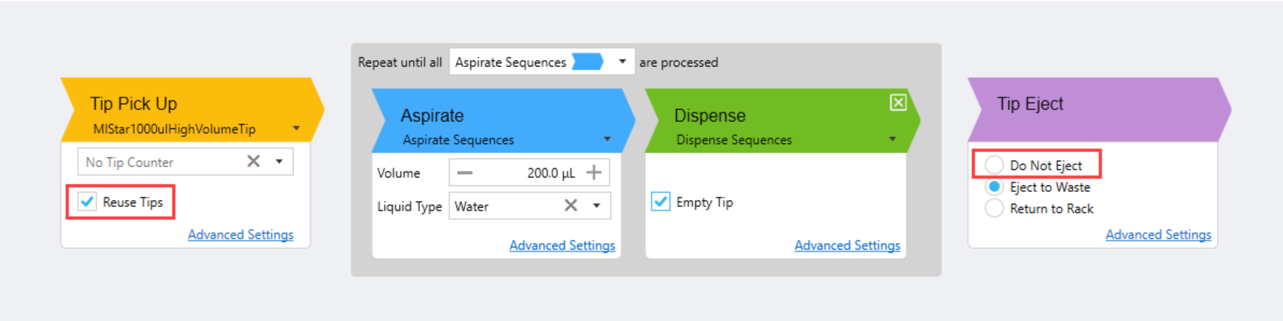


Tip Eject

Tips can be returned to their pickup position in the rack, ejected to the waste or left on the channels.

The ‘Do not eject’ option to leave the tips on the channels once the Power Step is completed is only available when reusing tips for the entire pipetting. This option is useful when reusing tips that have only been used to pipette non-contaminating liquids and will be used for the next step in the method, right after the Power Step.

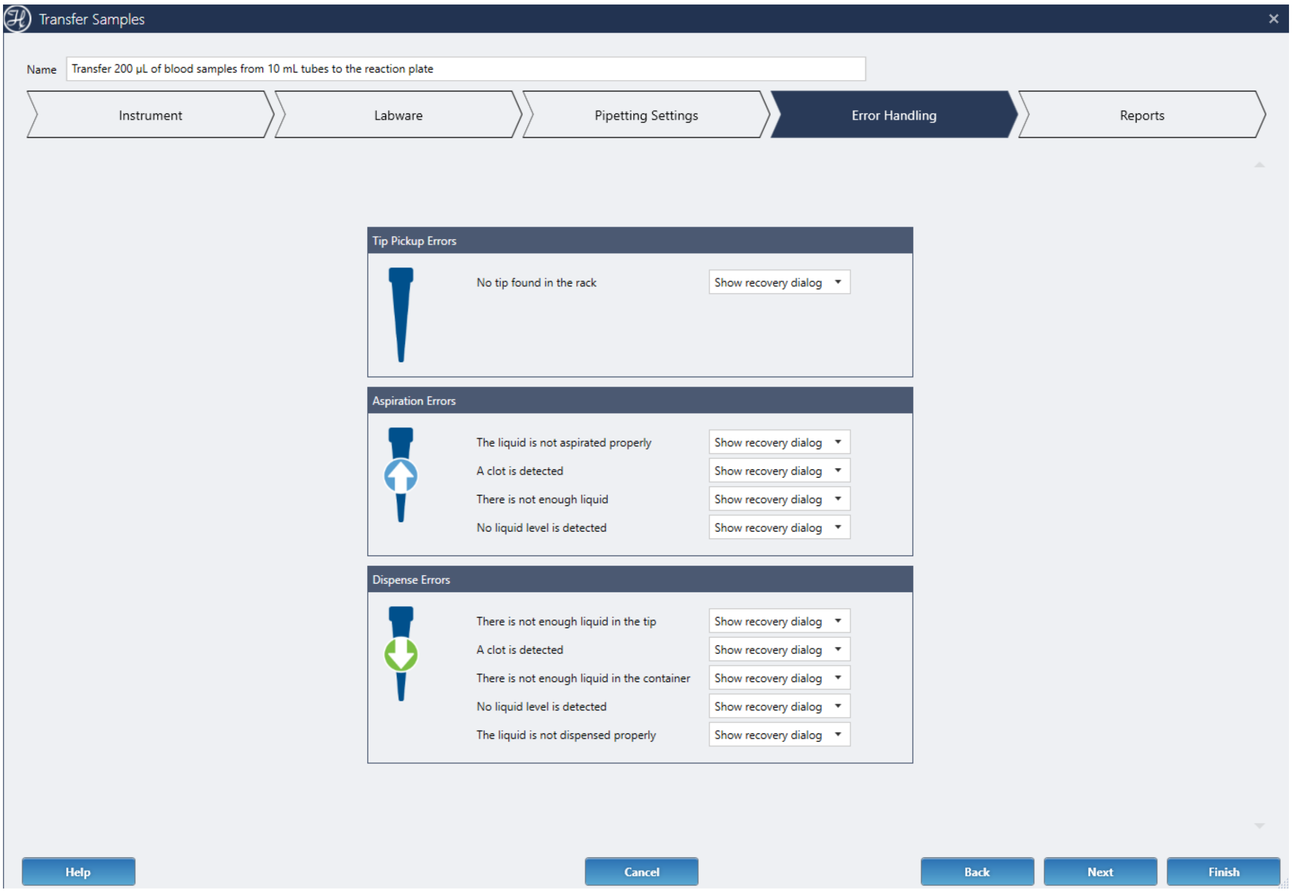




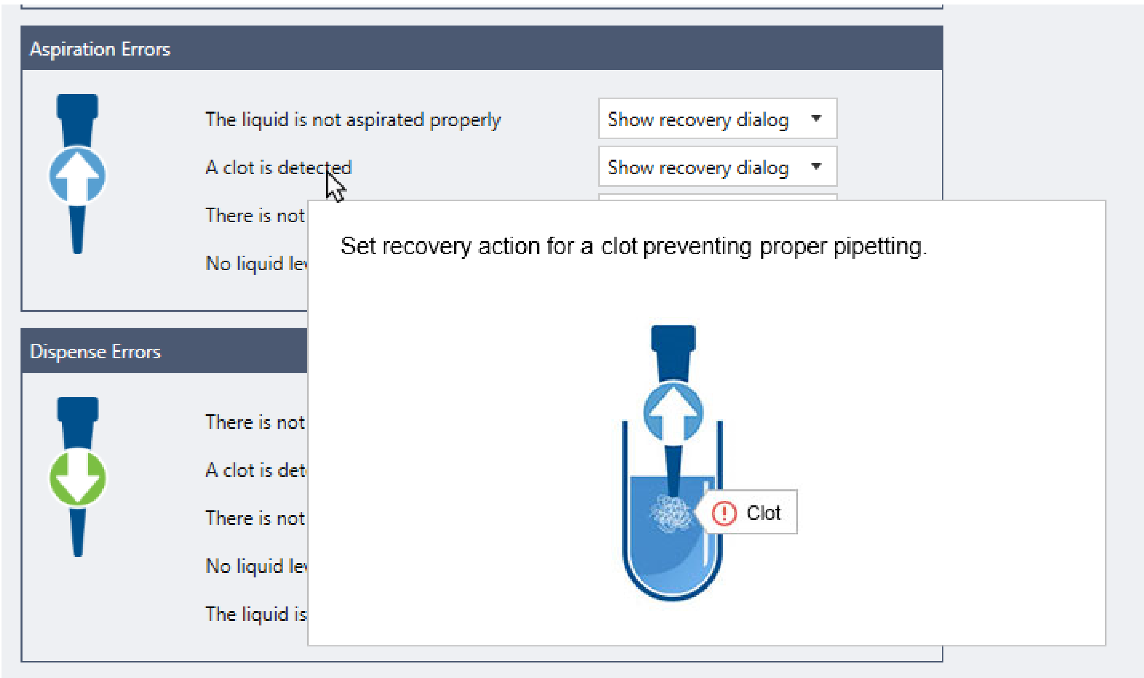
The Advanced Settings section configures the advanced error handling options for the tip eject steps. If enabled, it will override the general tip error handling defined on the Error screen.

13.7.2.4 Error Handling

The Error Handling screen displays a list of the most common errors that can occur during the pipetting process. Automated error recoveries can be set for each error.



Mouse over each error title or the recovery options in the drop-down list for additional details.

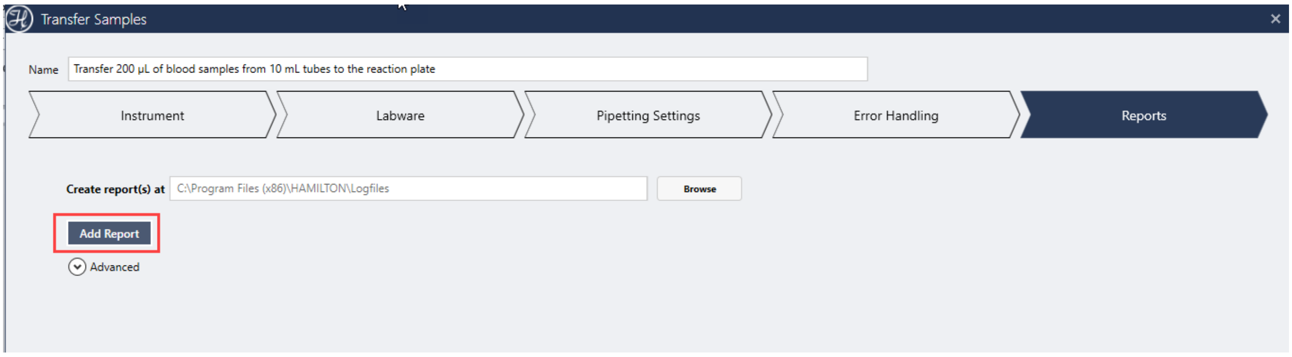


13.7.2.5 Reports

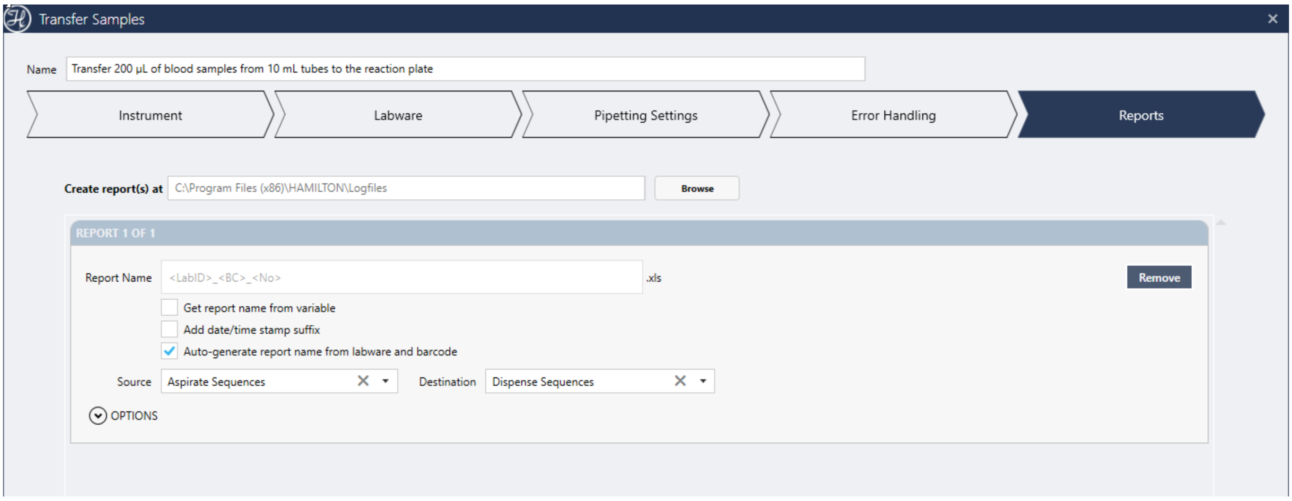
The Reports screen provides the option to generate a mapping report of the pipetting performed in the Power Step, including the sources and destinations information, barcodes, volumes, and pipetting errors.

	RecordId	TRackBC	TLabwareId	TPositionId	TPositionBC	TStatusSummary	TSumStateDescription	TVolume	SRackBC	SLabwareId	SPositionId	SPositionBC	ActionDateTime	UserName
2	1	X23000672	Plate_1	A1	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	1	S00001	2019-01-03 14:45:29	operator1
3	2	X23000672	Plate_1	B1	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	2	S00002	2019-01-03 14:45:29	operator1
4	3	X23000672	Plate_1	C1	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	3	S00003	2019-01-03 14:45:29	operator1
5	4	X23000672	Plate_1	D1	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	4	S00004	2019-01-03 14:45:29	operator1
6	5	X23000672	Plate_1	E1	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	5	S00005	2019-01-03 14:45:29	operator1
7	6	X23000672	Plate_1	F1	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	6	S00006	2019-01-03 14:45:29	operator1
8	7	X23000672	Plate_1	G1	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	7	S00007	2019-01-03 14:45:29	operator1
9	8	X23000672	Plate_1	H1	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	8	S00008	2019-01-03 14:45:29	operator1
10	9	X23000672	Plate_1	A2	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	9	S00009	2019-01-03 14:45:31	operator1
11	10	X23000672	Plate_1	B2	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	10	S00010	2019-01-03 14:45:31	operator1
12	11	X23000672	Plate_1	C2	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	11	S00011	2019-01-03 14:45:31	operator1
13	12	X23000672	Plate_1	D2	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	12	S00012	2019-01-03 14:45:31	operator1
14	13	X23000672	Plate_1	E2	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	13	S00013	2019-01-03 14:45:31	operator1
15	14	X23000672	Plate_1	F2	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	14	S00014	2019-01-03 14:45:31	operator1
16	15	X23000672	Plate_1	G2	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	15	S00015	2019-01-03 14:45:31	operator1
17	16	X23000672	Plate_1	H2	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	16	S00016	2019-01-03 14:45:31	operator1
18	17	X23000672	Plate_1	A3	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	17	S00017	2019-01-03 14:45:32	operator1
19	18	X23000672	Plate_1	B3	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	18	S00018	2019-01-03 14:45:32	operator1
20	19	X23000672	Plate_1	C3	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	19	S00019	2019-01-03 14:45:32	operator1
21	20	X23000672	Plate_1	D3	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	20	S00020	2019-01-03 14:45:32	operator1
22	21	X23000672	Plate_1	E3	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	21	S00021	2019-01-03 14:45:32	operator1
23	22	X23000672	Plate_1	F3	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	22	S00022	2019-01-03 14:45:32	operator1
24	23	X23000672	Plate_1	G3	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	23	S00023	2019-01-03 14:45:32	operator1
25	24	X23000672	Plate_1	H3	———	0	Correct pipetting	200	CarTubesS23	Sample_Tubes_1	24	S00024	2019-01-03 14:45:32	operator1

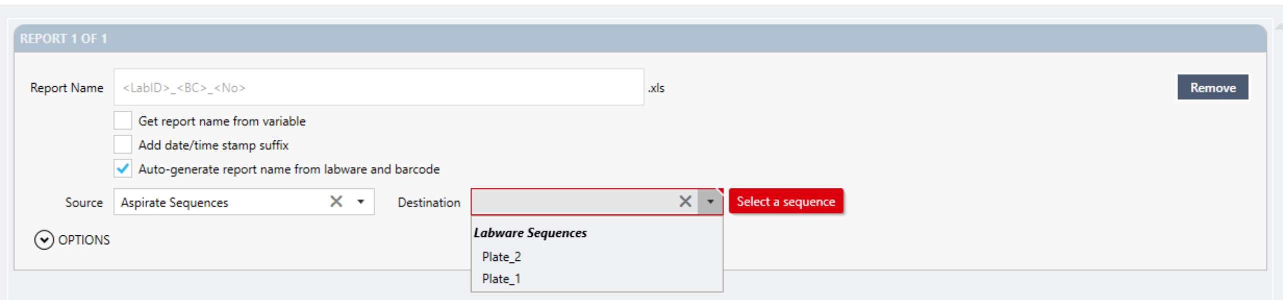
To generate a report at the end of the Power Step execution, click on 'Add Report'.



If there is only one Aspirate and one Dispense chevron in the Power Step, the source and destination fields of the report will be automatically populated.



If the Power Step has multiple aspirates or dispenses (as a result of the advanced settings in the Labware screen).



Report Name and Path

The reports are generated by default in the standard Hamilton\Logfiles folder (e.g. `C:\Program Files (x86)\HAMILTON\LogFiles`). A different path can be set by clicking “Browse”.

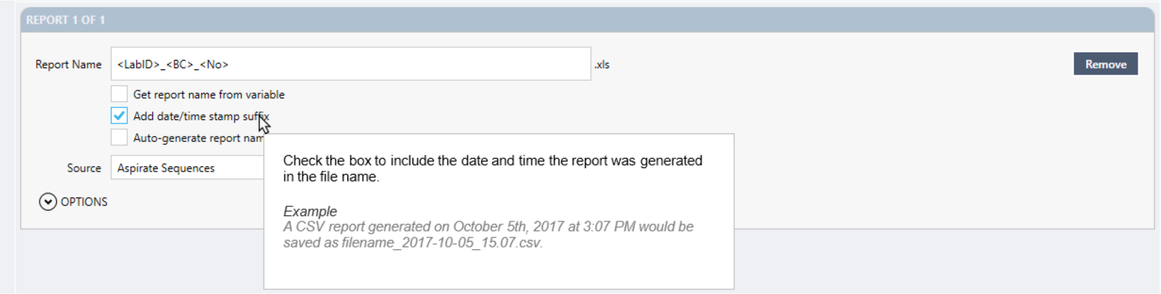
The report name can be assigned in three ways:

- Auto-generate the name, based on the labware ID and barcode of the destination labware. This is the default setting.

<LabID>	Replaced by the labware identification string (e.g. nun_96_l_0001) of the rack for which the report file is generated.
<BC>	Replaced by the barcode of associated labware.
<No>	Replaced by an auto-incrementing number starting with zero to make the file name unique within the specified directory.

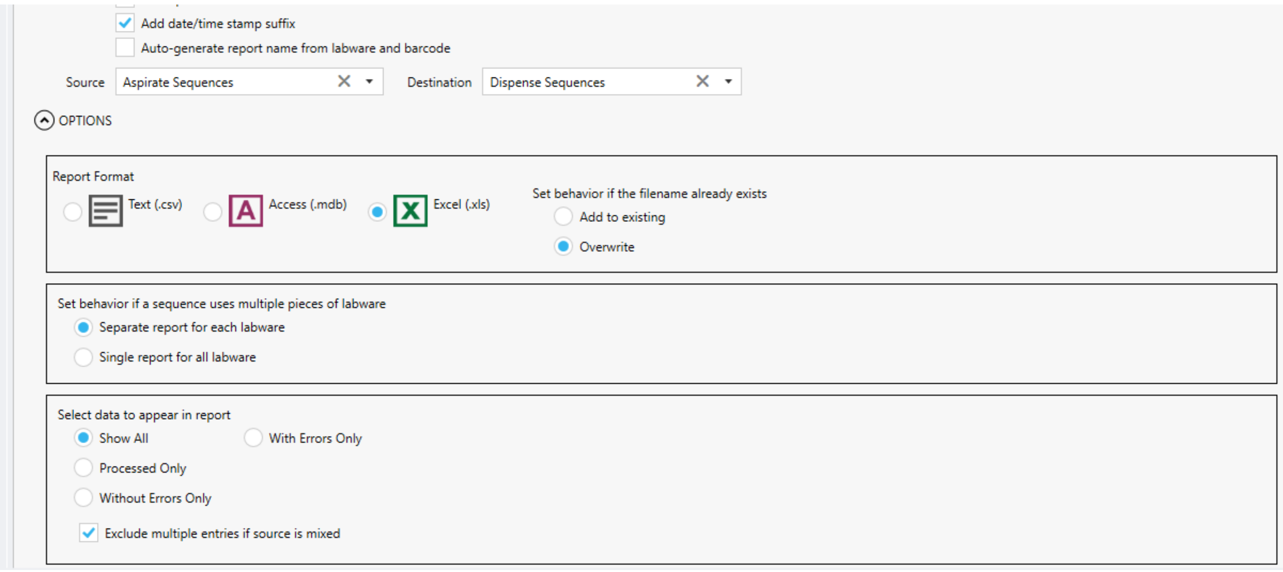
- Manual entry, by unchecking the ‘Auto-generate’ box and typing a new name in the ‘Report name’ field.
- Using a variable, by checking the “Get report name from variable”. This option will only be available if the method has variables defined.

  Optionally, a time stamp suffix can be automatically added to the report name when the report is generated at runtime.

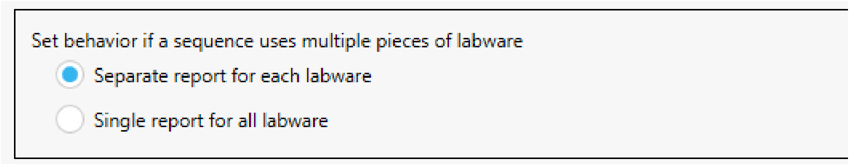


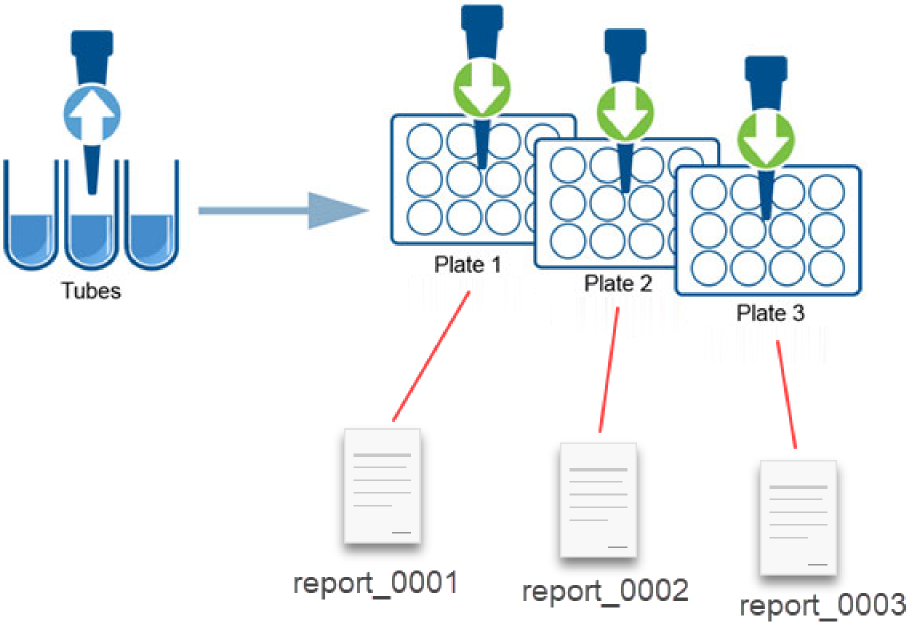
Report Options

The report is generated by default in Excel format (.xls). In the Options section this setting can be changed to ASCII text (.csv) or Access database (.mdb).



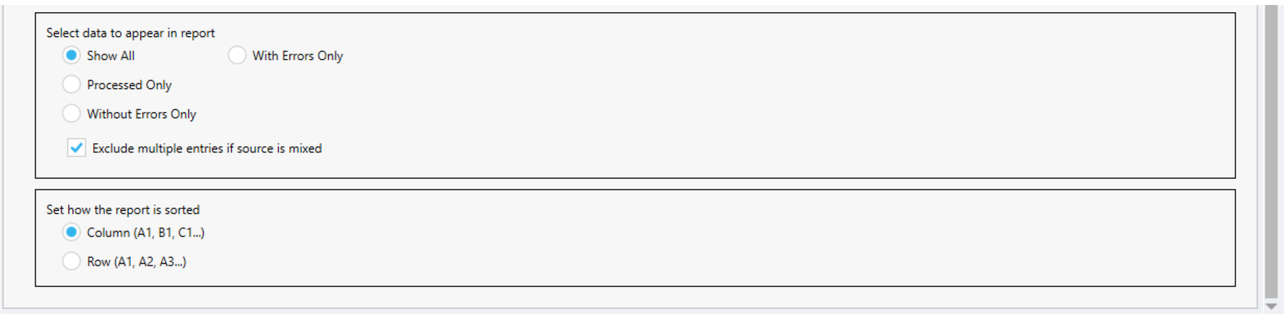
If the destination sequence involves more than one piece of labware, individual reports can be generated for each target labware. If this option is selected, each report will have an additional suffix indicating the labware index within the sequence.



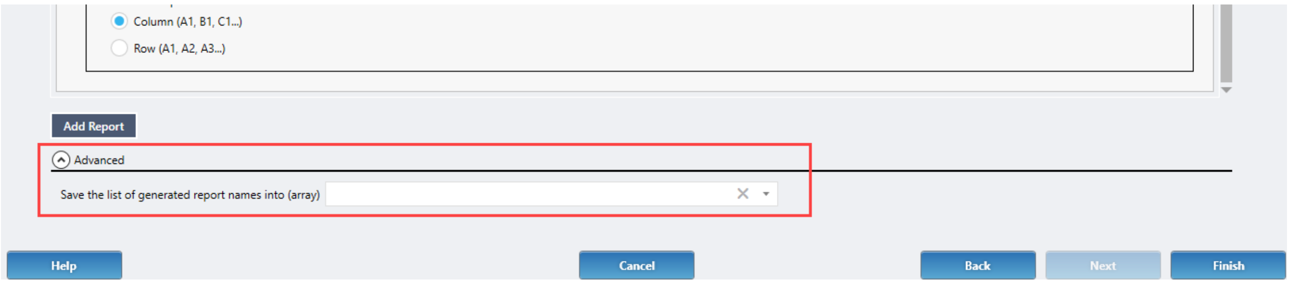


The report by default includes all the positions present in the destination labware. Some of these wells can be filtered out, depending on their state (e.g., processed, empty, with error, etc.).

The list of entries in the report is sorted by the container/well ID of the destination labware. By default, it is sorted by column (A1, B1, C1) in plates or sequentially (1, 2, 3) in tube racks. This setting can be changed to sort the entries by row.



At the bottom of the screen there is an Advanced section to optionally store the generated report filenames (path + name) into an array for later use in the method.



Report Contents

For an ASCII text file, the list-separator as set in the Regional settings will be used as column delimiter, and text is delimited by double quotes (" ").

The table name, in a Microsoft Excel sheet as well as in a Microsoft Access database, is set to ReportMapping.

Command	Icon	Action Performed
RecordId	Long	Unique record identifier.
TRackBC	String *	The barcode of the processed labware; may be empty.
TLabwareId	String	The labware identification string of the target labware.
TPositionId	String	The labware position (container) identification string of the target labware (an alphanumeric or a numeric index e.g. A1).
TPositionBC	String *	The barcode of the target position (container), e.g. barcode of sample tube; may be empty.
TStatusSummary***	Long	A bit-coded overall state of the target container (TargetPositionId).
TSumStateDescription	String	A human-readable description of the target container state, e.g. No error, Well not used, Barcode warning AND Pipetting error, etc.
TVolume	Double **	The current volume (volume at file creation time) of the target container [µl].
SRackBC	String *	The barcode of the source rack (e.g. sample carrier); may be empty.
SLabwareId	String *	The labware identification string of the source labware; may be empty.
SPositionId	String *	The labware position (container) identification string of the source labware (an alphanumeric or a numeric index e.g. A1); may be empty.
SPositionBC	String *	The barcode of the source position (container), e.g. barcode of sample tube; may be empty.
ActionDateTime	String *	Action timestamp (format: YYYY-MM-DD hh:mm:ss) of aspirate action from source; may be empty.

Command	Icon	Action Performed
UserName	String	Name of the user currently logged into the system.

* If no data is available, null fields in columns of type string are converted to the string '----------'

** If there was never a volume move, the value 1E+9 is set

*** TStatusSummary

The status summary is a bit-coded value. If several errors occurred, the value represents the bitwise addition of the individual error codes.

No errors

Column Name	Type	Description
0	0x0000000	No error.
1	0x0000001	Well not used, i.e. no action executed to this well.

Warnings

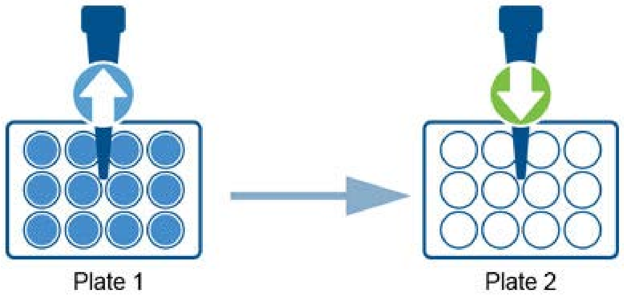
Column Name	Type	Description
2	0x0000002	Barcode warning. User entered barcode manually or user-defined error-handling executed during identification of labware.
4	0x0000004	Pipetting warning. User-defined error-handling executed during volume move step.
8	0x0000008	Transport warning. User-defined error-handling executed during element move step.
16	0x0000010	Load warning. User-defined error-handling executed during element load step.
32	0x0000020	Wash warning. User-defined error-handling executed during wash step.
2048	0x0000800	Previous run warning. Labware is affected with warning(s) from previous run(s).
4096	0x0001000	Unspecified warning.

Errors

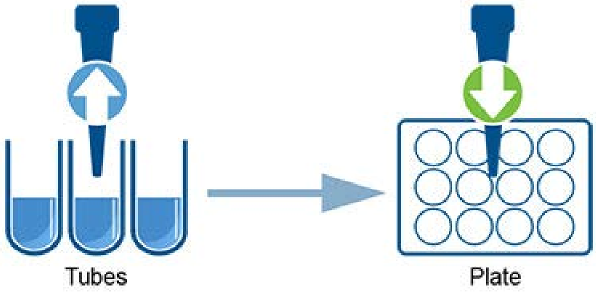
Column Name	Type	Description
8192	0x0002000	Barcode error.
16384	0x0004000	Pipetting error.
32768	0x0008000	Transport error.
65536	0x0010000	Load error.
131072	0x0020000	Wash error.
8388608	0x0800000	Previous run error. Labware is affected with error(s) from previous run(s).
16777216	0x1000000	Unspecified error.

13.7.2.6 Usage Examples

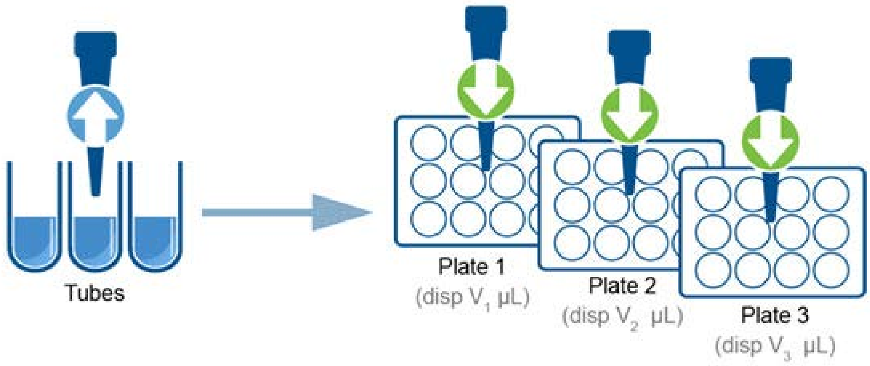
Example 1: Transfer samples between plates



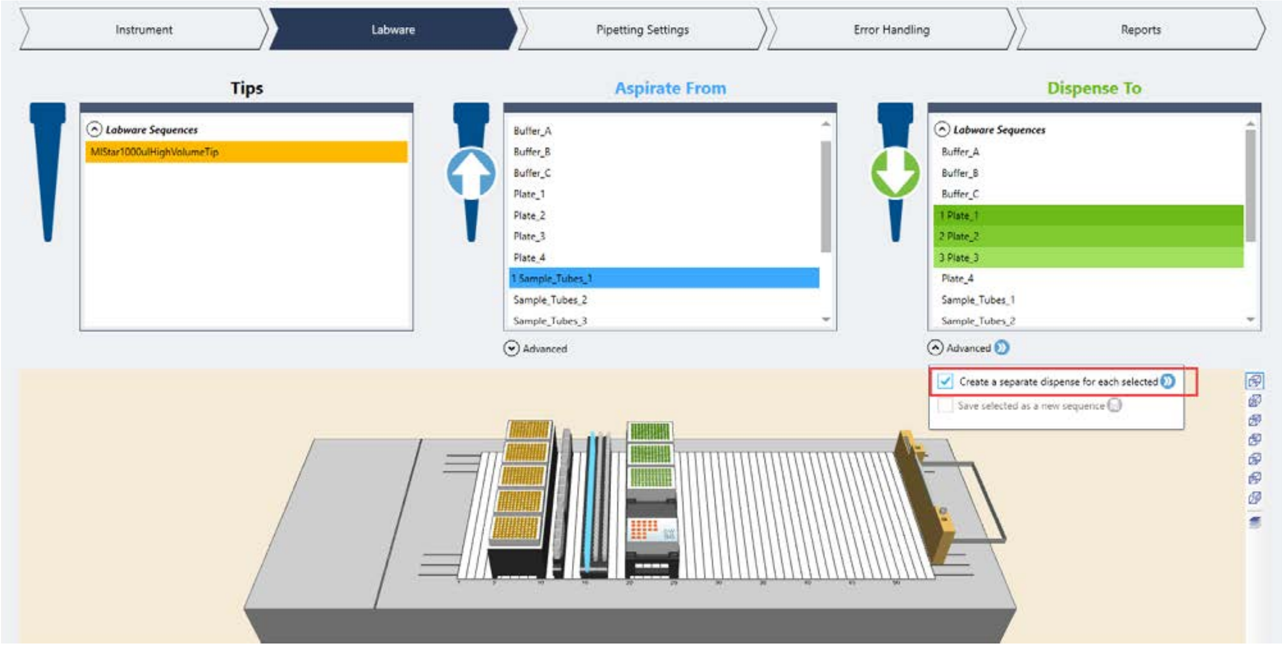
Example 2: Transfer samples from tubes to a plate

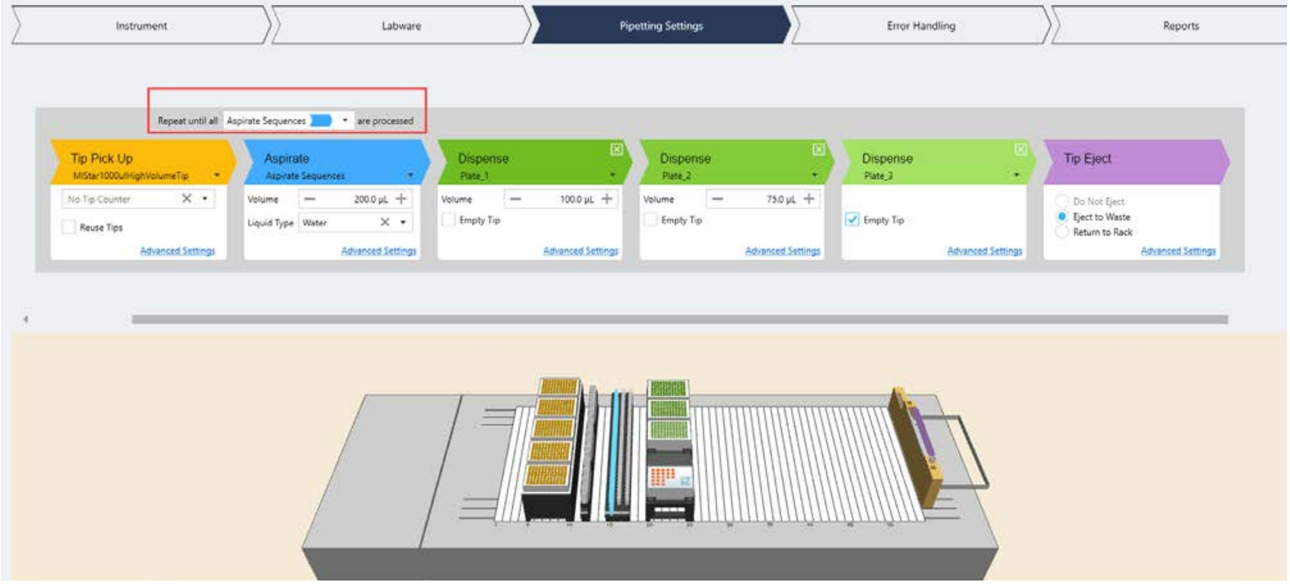


Example 3: Transfer samples from tubes to 3 different plates with different aliquot volumes

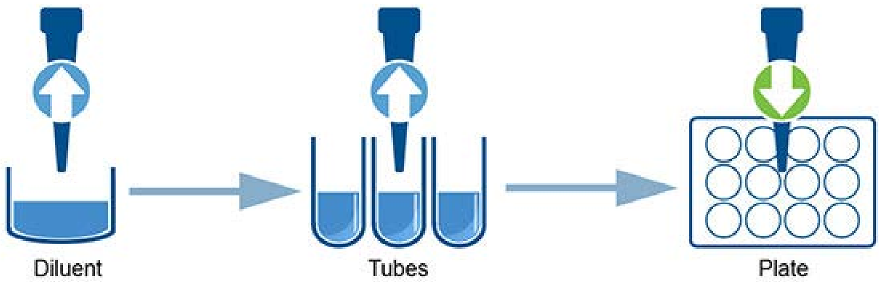


Remember to select “Create separate dispense for each selected” on the labware screen and that in the Pipetting Settings screen the process is repeated until all sample tubes (the Aspirate Sequences) are processed.

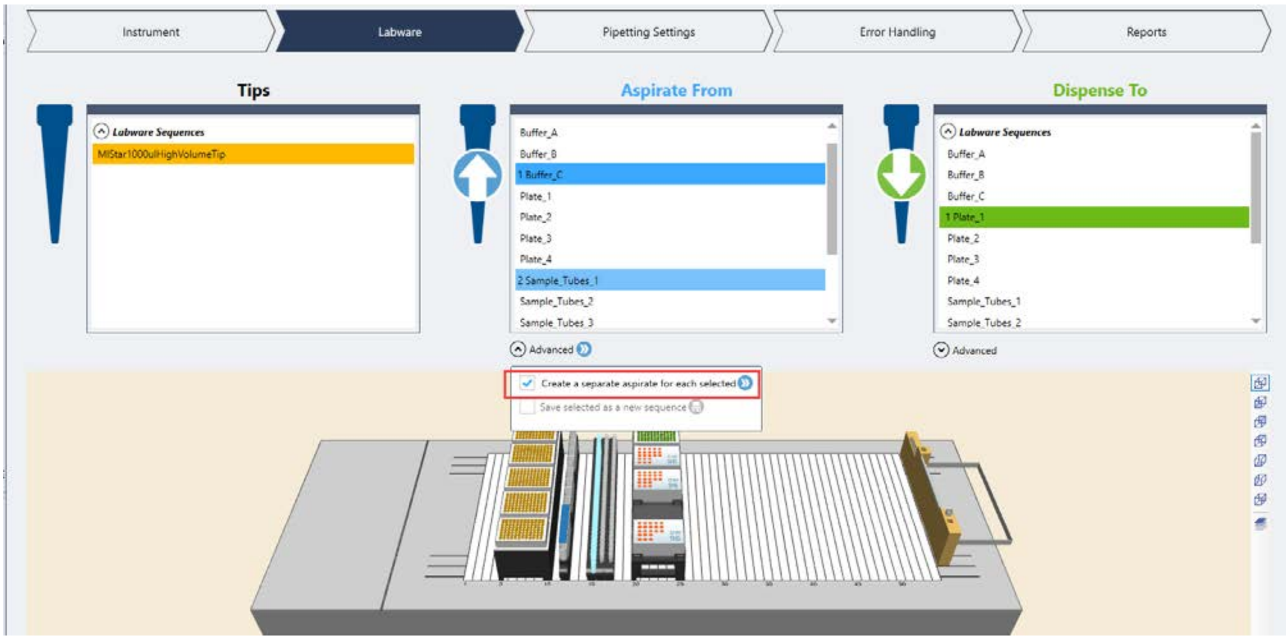


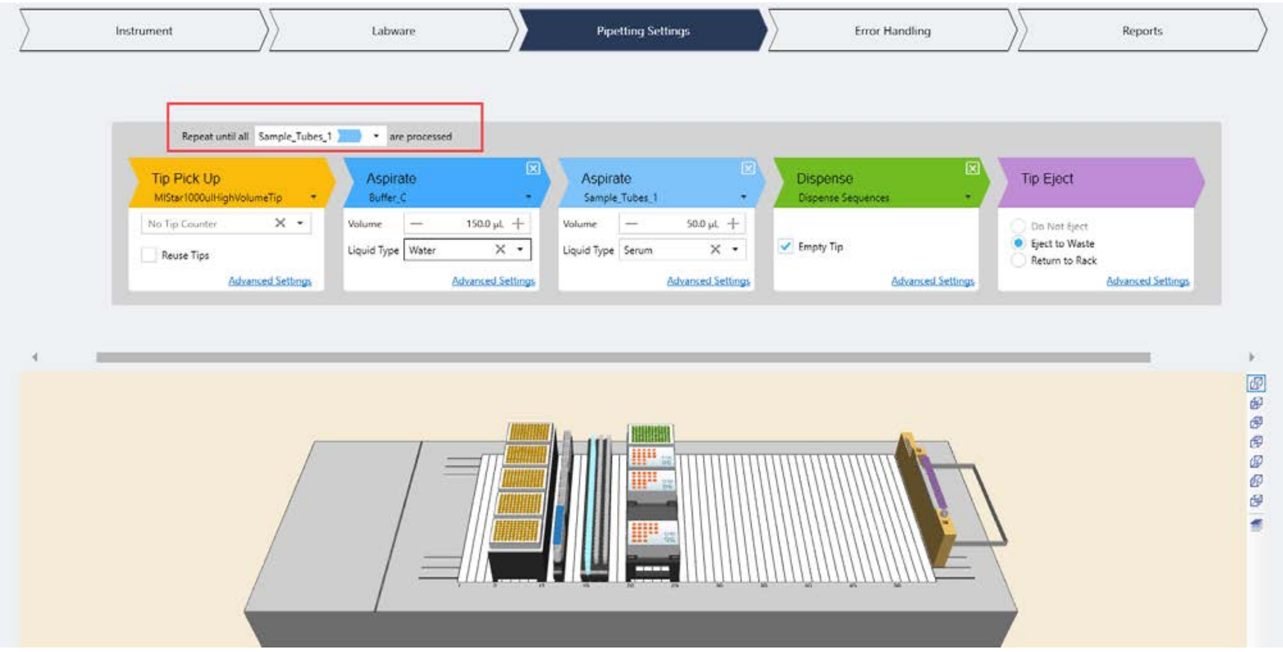


Example 4: Aspirate diluent, then aspirate samples and transfer to a plate

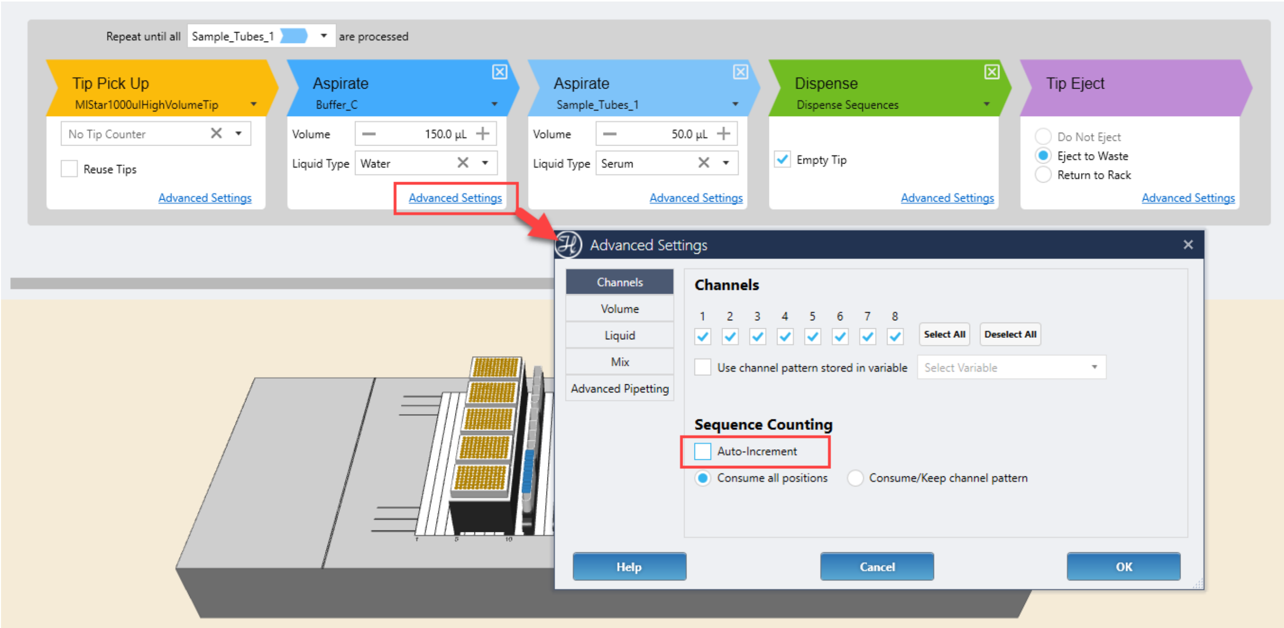


Remember to select “Create separate aspirate for each selected” in the Labware screen and that in the Pipetting Settings screen the process is repeated until all sample tubes (the samples Aspirate sequence) are processed.





Disable the auto-increment of the diluent aspirate to ensure it always comes back to the same reservoir positions in each loop of the Power Step.



13.7.3 Add Reagent

The Add Reagent Power Step sets up the distribution of a reagent to a set of targets over one (or multiple) pieces of labware.

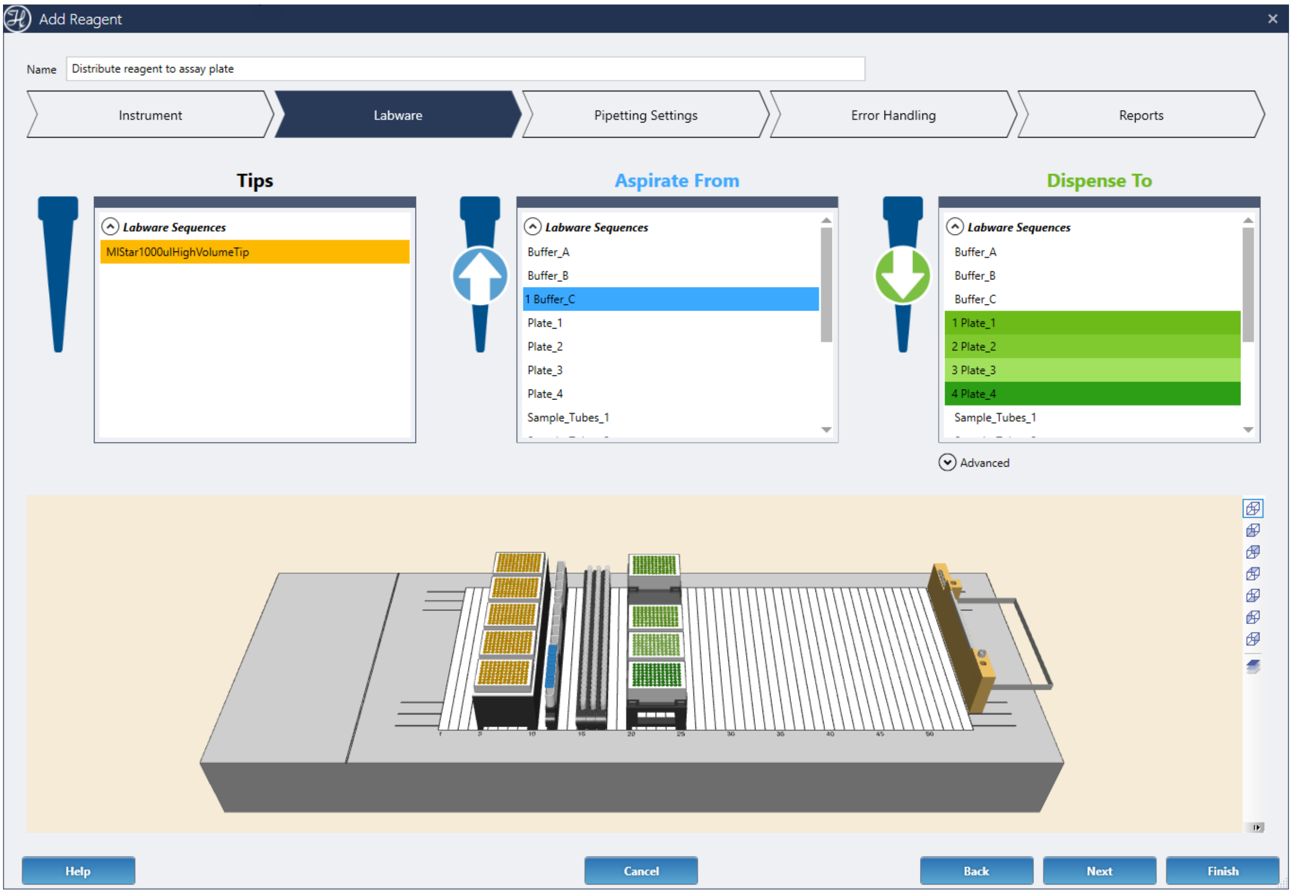
13.7.3.1 Instrument

This screen has the same settings and options as the Instrument screen of the Transfer Samples Power Step. Refer to the Transfer Samples > Instrument section of this manual for details.

13.7.3.2 Labware

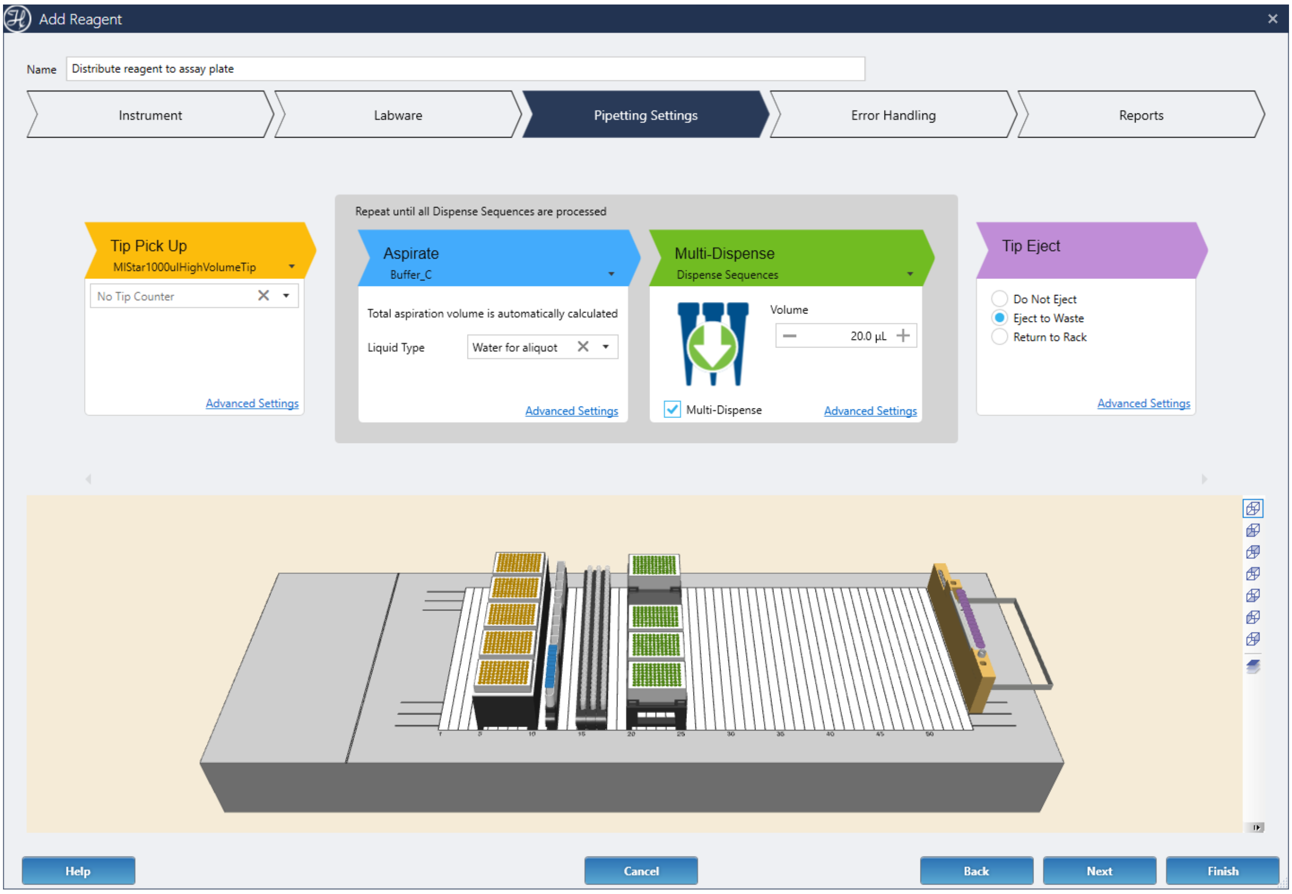
This screen has the same settings and options as the Labware screen of the Transfer Samples Power Step. Refer to the Transfer Samples > Labware section of this manual for details.

Note that the 'Aspirate From' in the Add Reagent step is limited to select only one sequence and that is not possible to create separate dispenses for each 'Dispense To' sequence selected.


Add Reagent
Name
Distribute reagent to assay plate
Instrument
Labware
Pipetting Settings
Error Handling
Reports
Tips
Aspirate From
Dispense To
Labware Sequences
MlStar1000ulHighVolumeTip
Labware Sequences
Buffer_A
Buffer_B
1 Buffer_C
Plate_1
Plate_2
Plate_3
Plate_4
Sample_Tubes_1
Labware Sequences
Buffer_A
Buffer_B
Buffer_C
1 Plate_1
2 Plate_2
3 Plate_3
4 Plate_4
Sample_Tubes_1
Advanced
Help
Cancel
Back
Next
Finish

13.7.3.3 Pipetting Settings

The Pipetting Settings screen displays a diagram of the pipetting procedure and steps.



The top gray block represents the steps that will be repeated in a loop until all the Dispense positions are processed.

Mouse over any of the colored chevrons to blink their related positions on the deck.

Tip Pick Up

This step is the same as the as the Tip Pick Up step of the Transfer Samples Power Step. Refer to the Transfer Samples > Pipetting Settings > Tip Pick Up section of this manual for details.

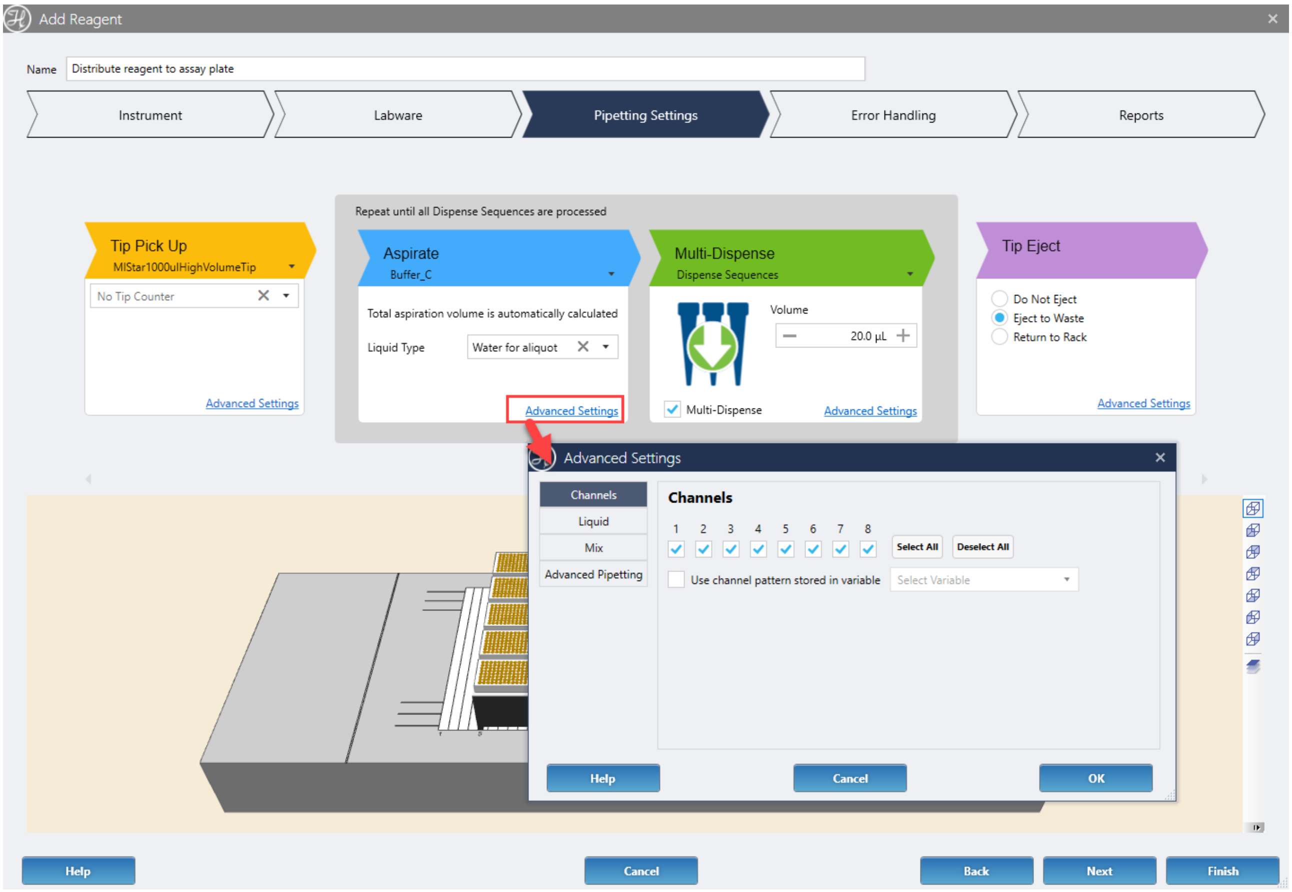
Note that the Tip Pick Up is limited to use the same tips for the entire pipetting. If new tips need to be used before each aspirate use the Transfer Samples step instead.

Aspirate

This step is the same as the as the Aspirate step of the Transfer Samples Power Step. Refer to the Transfer Samples > Pipetting Settings > Aspirate section of this manual for details.

Note that the Aspirate volume is automatically calculated in this step, based on the tip size, the dispense mode, and the number of target dispenses required.

If the pipetting is performed with multi-dispenses (1 large aspiration □ N dispenses), the software automatically calculates the aspirate volume to dispense as many targets as possible without coming back to the reagent source. Due to this automatic calculation, the 'volume' section on the Aspirate advanced settings is not available.

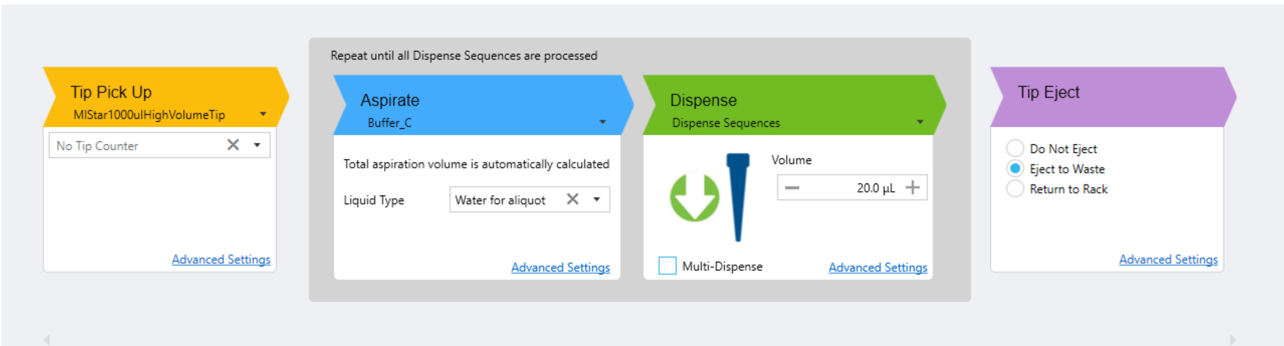


Dispense

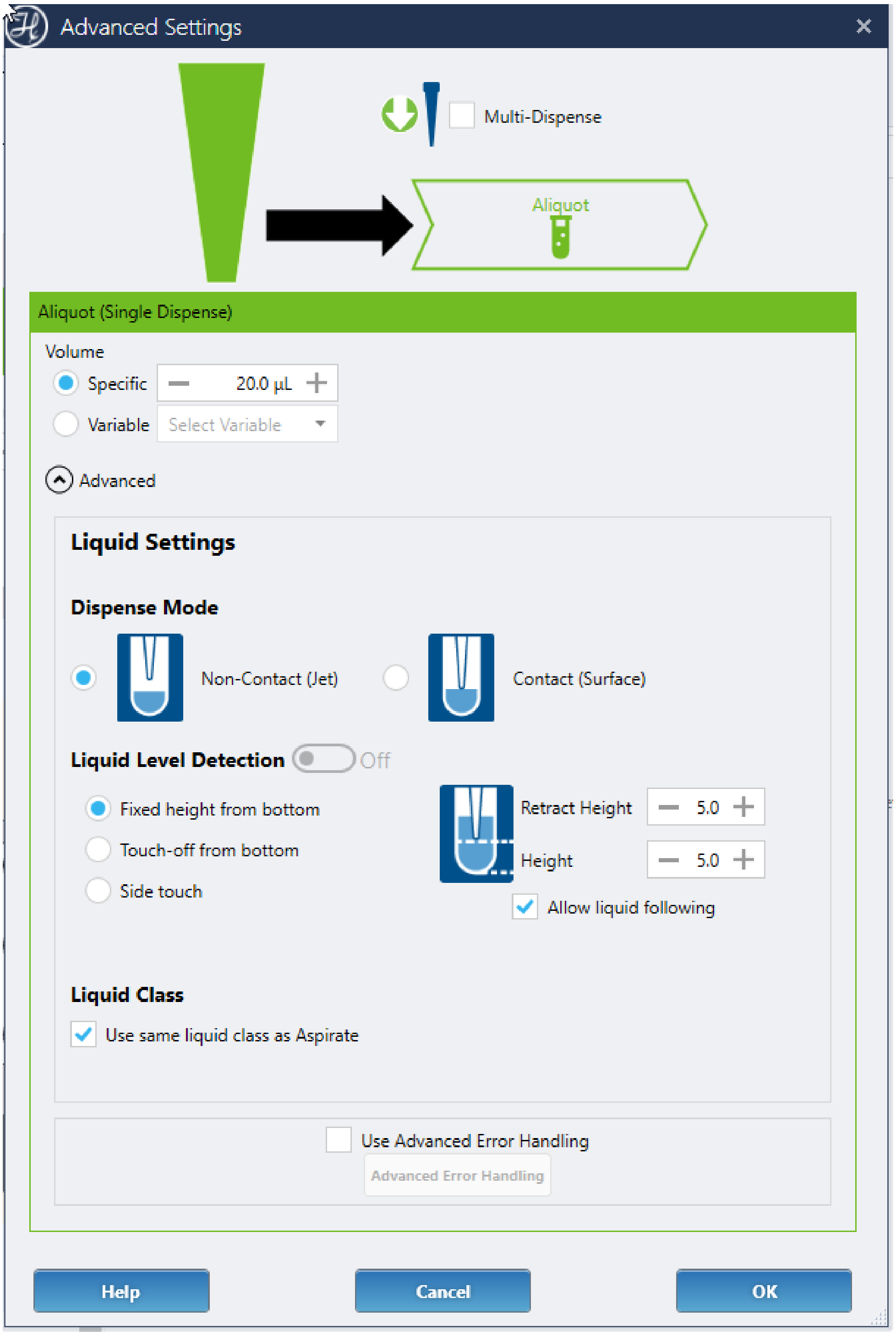
The dispense step can be configured in single dispense or multiple dispense mode. The volume set in the dispense step determines the volume of reagent to be dispensed in each target well, independent of the dispense mode (single/multi) selected.

- **Single dispense**: one aspiration followed by one dispense with the same tip

  Aspirated volume = volume per target well

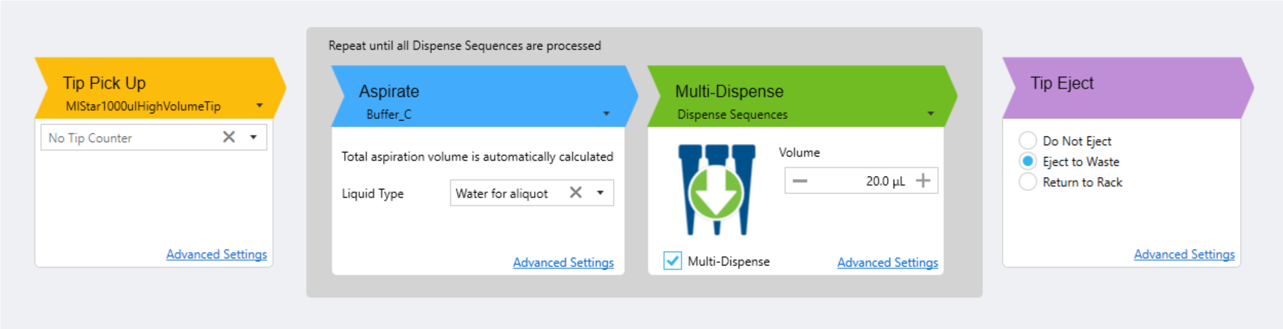


The advanced settings for single dispense are the same as the advanced settings liquid section of the Transfer Sample step. Refer to Transfer Samples > Pipetting Settings > Dispense > Advance settings – Liquid Section for more details.

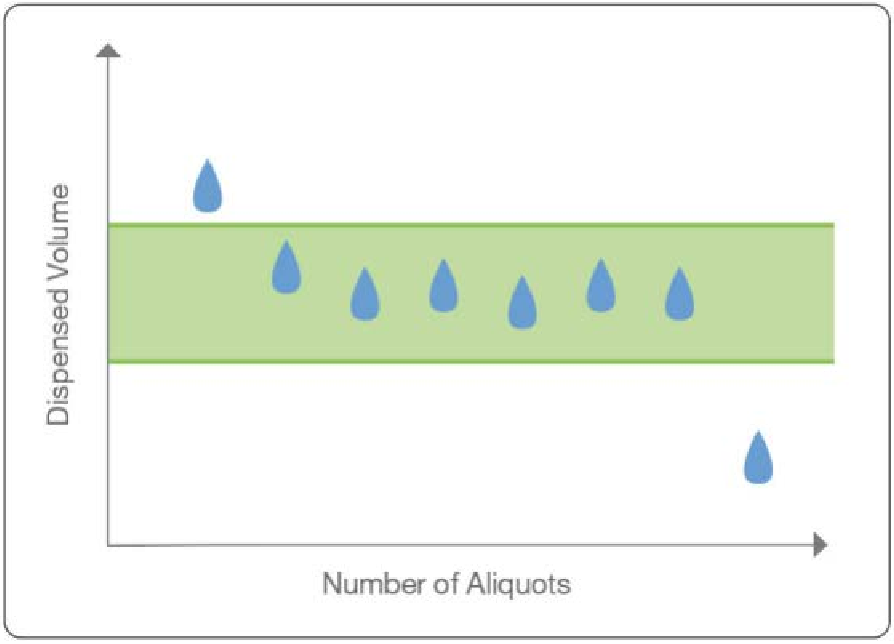


- **Multi-dispense**: also known as aliquoting, is the act of aspirating enough total volume to dispense more than once from the same tip.

  Aspirated volume = pre-aliquot volume + n * volume per target well + post-aliquot volume whereas n depends on the tip size.



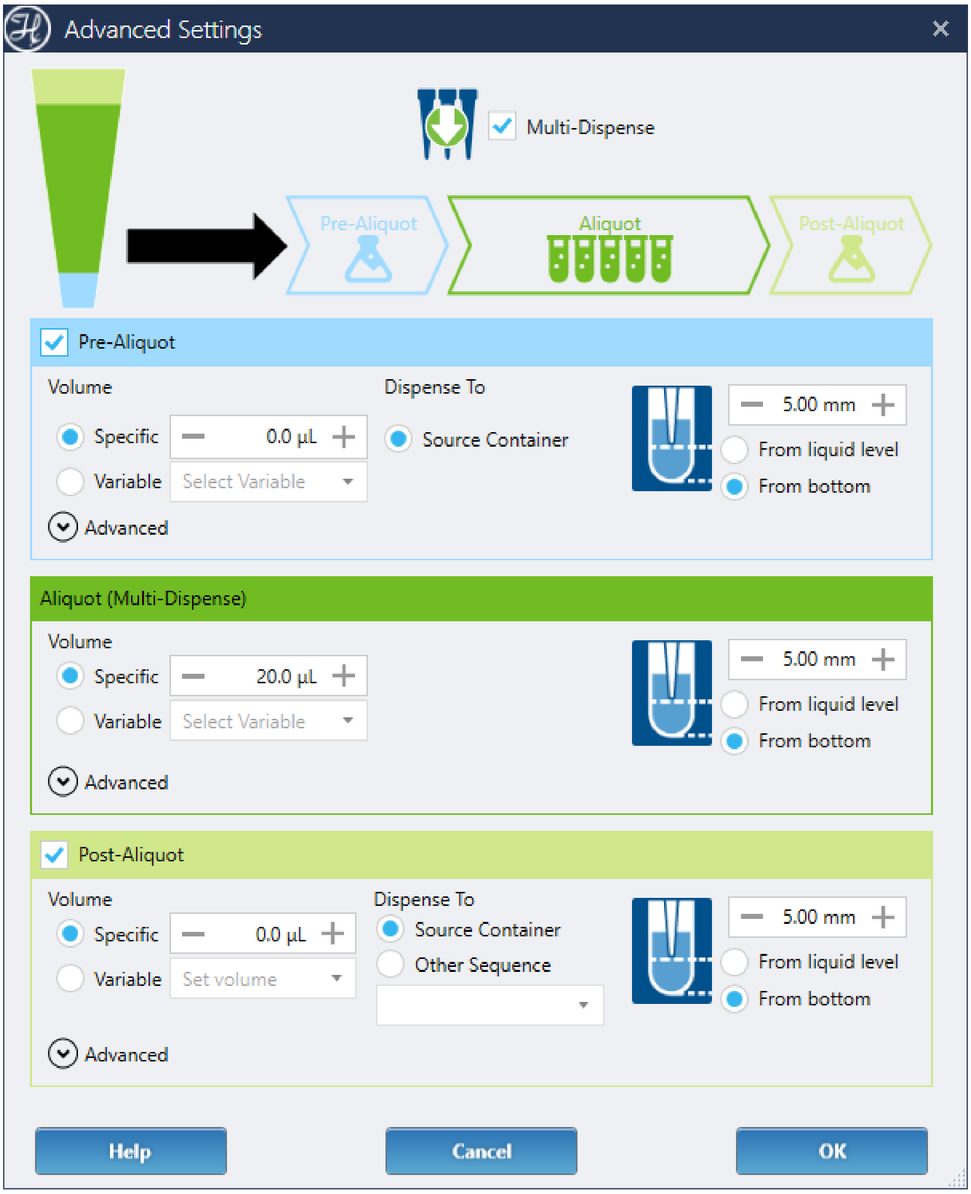
When multi-dispensing, the first and last aliquots are often inaccurate and should be discarded in order to maintain good precision overall amongst the transfers. Some liquids may require 2 pre-dispenses. See the image below for an example of a pre- and post-aliquot transfer that are discarded, ensuring that the transfer in-between is within an acceptable range.



The **pre-aliquot** is aspirated at the end of the aspiration, it is the first aliquot to be dispensed, and it is always dispensed back to the source container.

The **post-aliquot** is aspirated at the beginning of the aspiration, and it is the last aliquot to be dispensed.

The advanced settings for multi-dispense are the same as for the single dispense, with additional options to define pre- and post-aliquots.



Note that the pre-aliquot is always dispensed back to the source container, while the post-aliquot can be dispensed to any deck sequence.

Tip Eject

This step is the same as the Tip Eject step of the Transfer Samples Power Step. Refer to the Transfer Samples > Pipetting Settings > Tip Eject section of this manual for details.

13.7.3.4 Error Handling

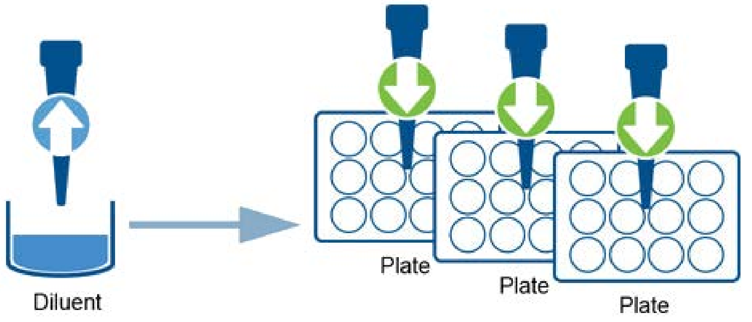
This screen has the same settings and options as the Errors screen of the Transfer Samples Power Step. Refer to the Transfer Samples > Error Handling section of this manual for details.

13.7.3.5 Reports

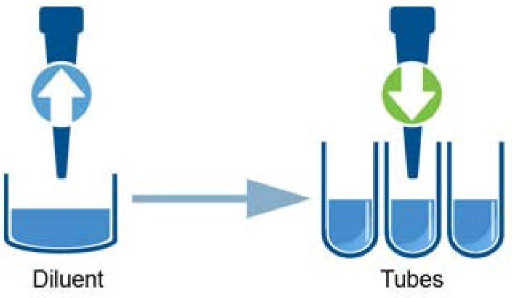
This screen has the same settings and options as the Reports screen of the Transfer Samples Power Step. Refer to the Transfer Samples > Reports section of this manual for details.

13.7.3.6 Usage Examples

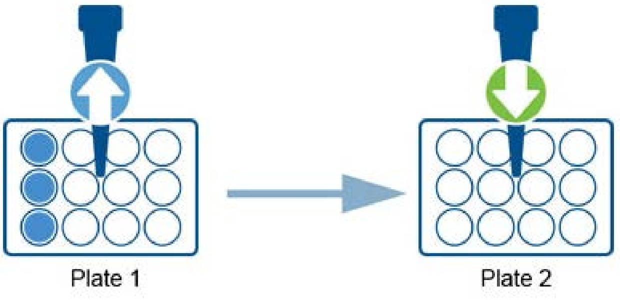
Example 1: Add reagent to 3 plates



Example 2: Add reagent to sample tubes



Example 3: Add reagent from a plate column



13.7.4 Serial Dilution

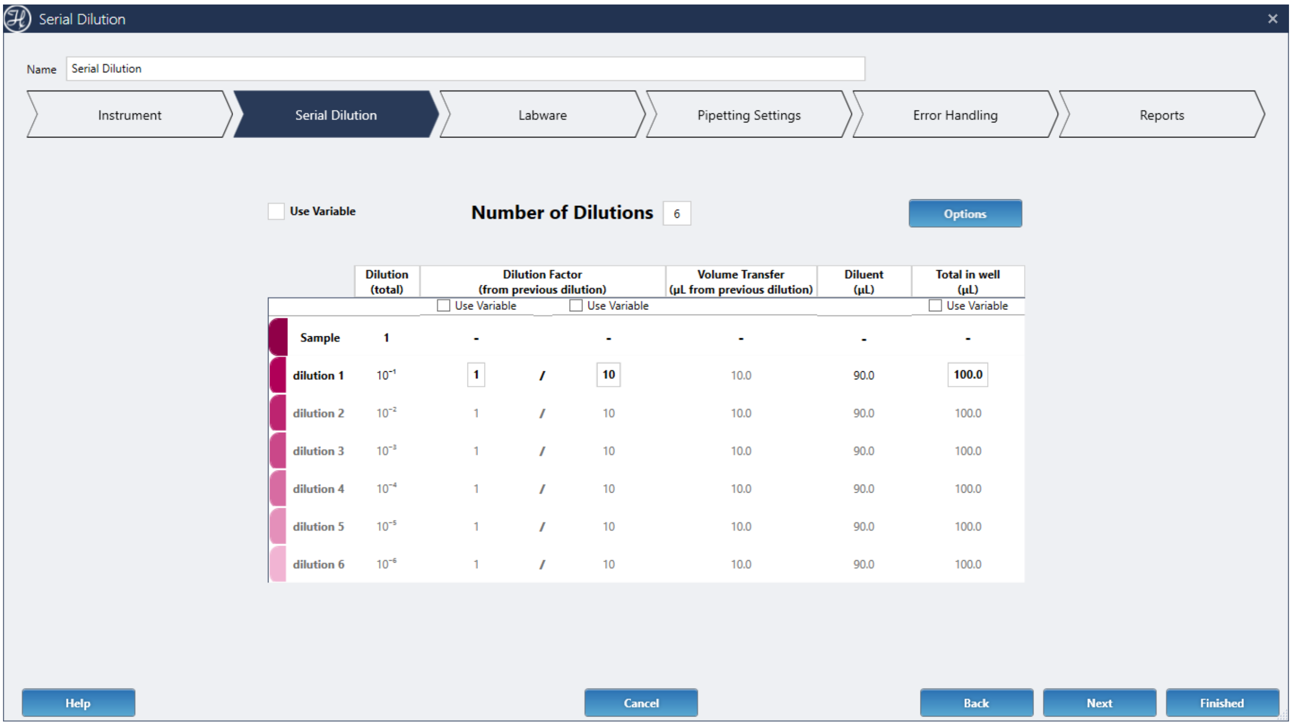
The Serial Dilution Power Step sets up the distribution of a reagent, samples addition and serial dilution of a set of samples. The serial dilution can be performed in tubes, plate rows, or plate columns.

13.7.4.1 Instrument

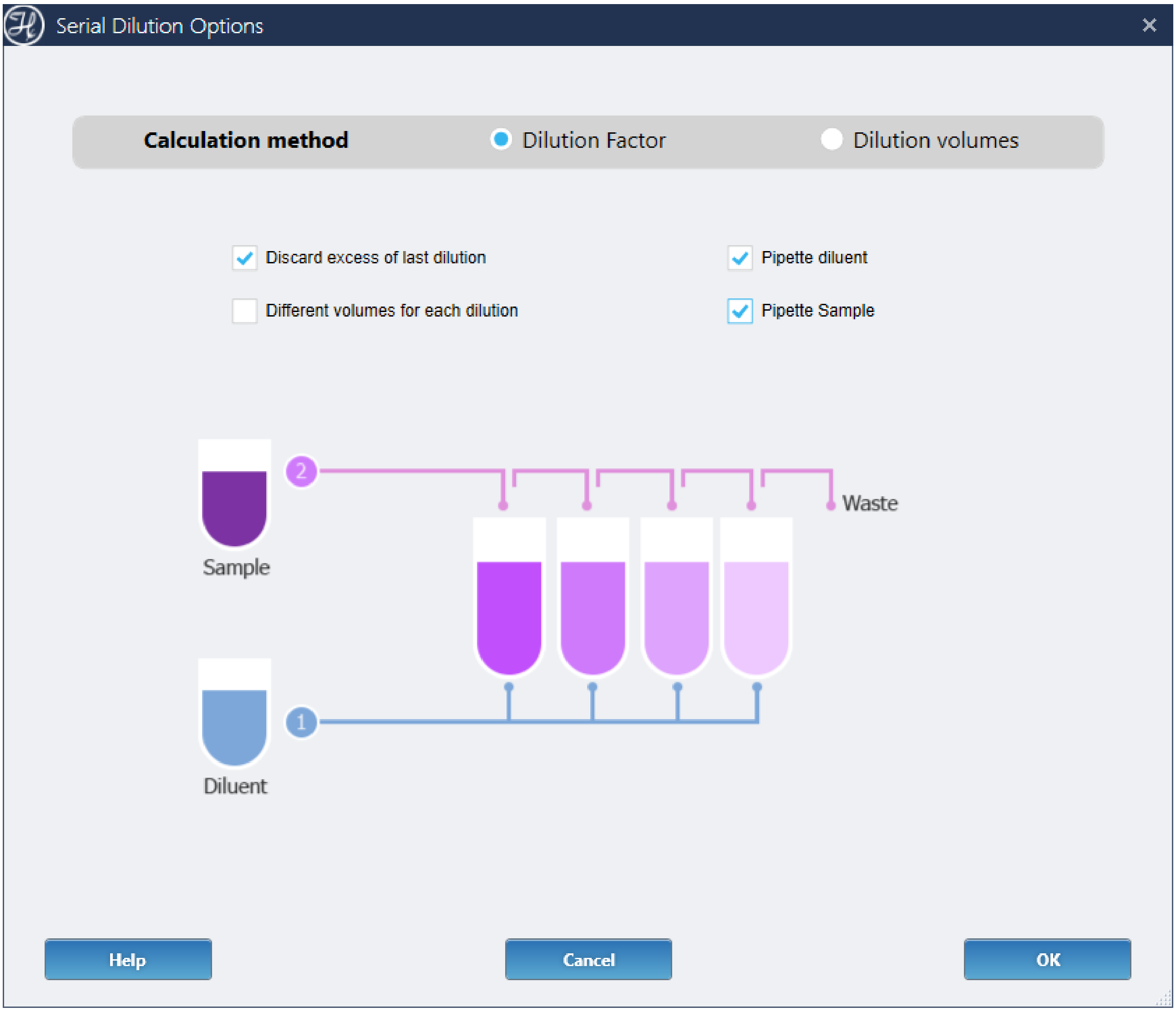
This screen has the same settings and options as the Instrument screen of the Transfer Samples Power Step. Refer to the Transfer Samples > Instrument section of this manual for details.

13.7.4.2 Serial Dilution

This screen configures the number of dilutions, dilution factors, and diluent and sample volumes.



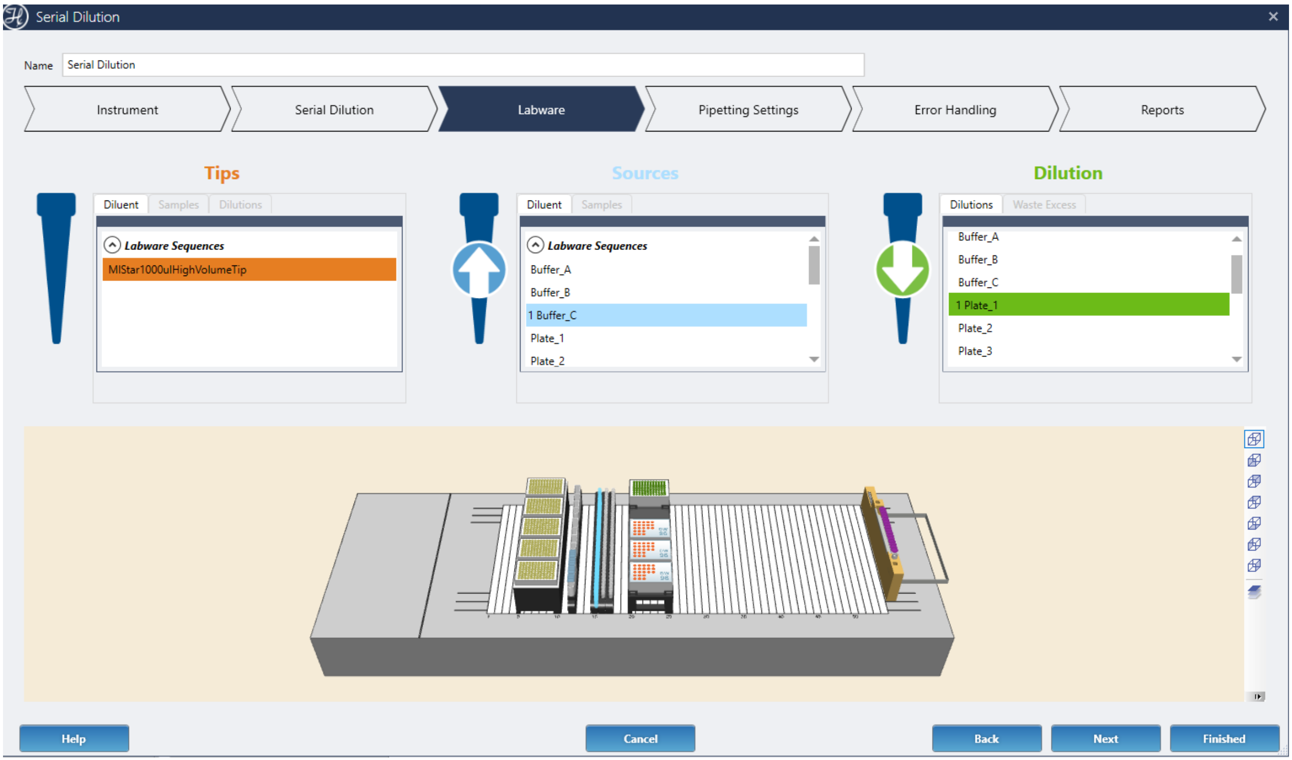
Click the Options button to access all the serial dilution settings.



- **Calculation method**:
  based on dilution factors or the volume to transfer between dilutions.
- **Discard excess of last dilution**:
  to the waste or another container.
- **Different volumes for each dilution**:
  enable this option to specify different dilution factors or volumes for each dilution.
- **Pipette diluent**:
  pipette the required diluent volumes on the target containers before pipetting the samples and performing the serial dilution.
- **Pipette samples**:
  pipette the samples from a different source to the first target container before performing the serial dilution. Enable this option if the dilution plate or tube(s) do not already contain the source sample(s).

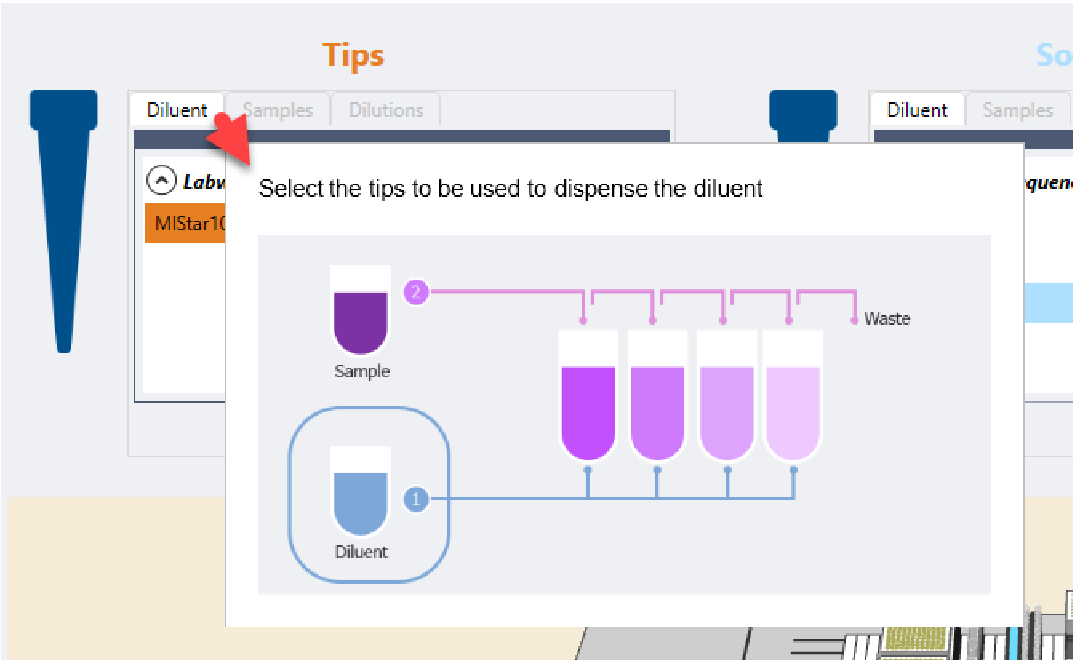
13.7.4.3 Labware

The Labware screen displays the deck layout and the tips, aspirate and dispense sequences used to perform the serial dilution.



Depending on the serial dilution options selected in the Serial Dilution screen, different tabs will be available under each section.

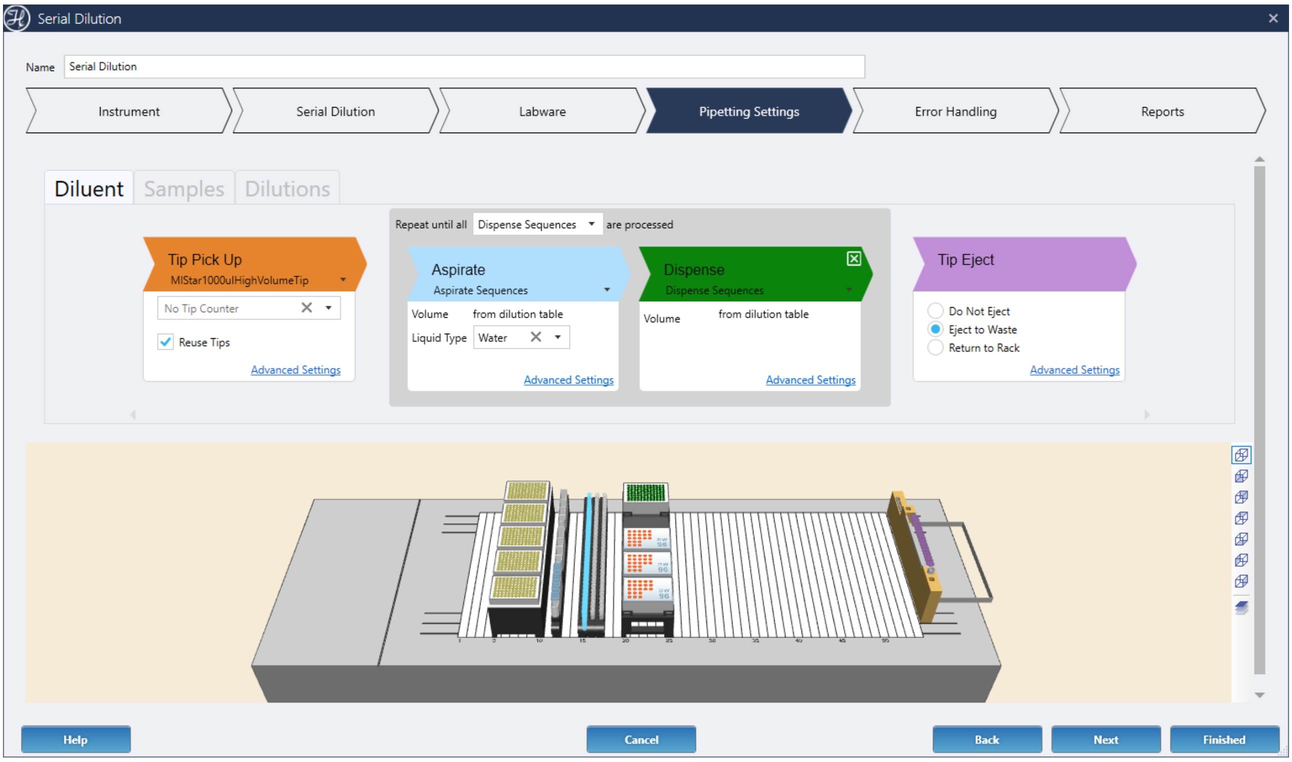
Mouse over any tab under each section for additional information.

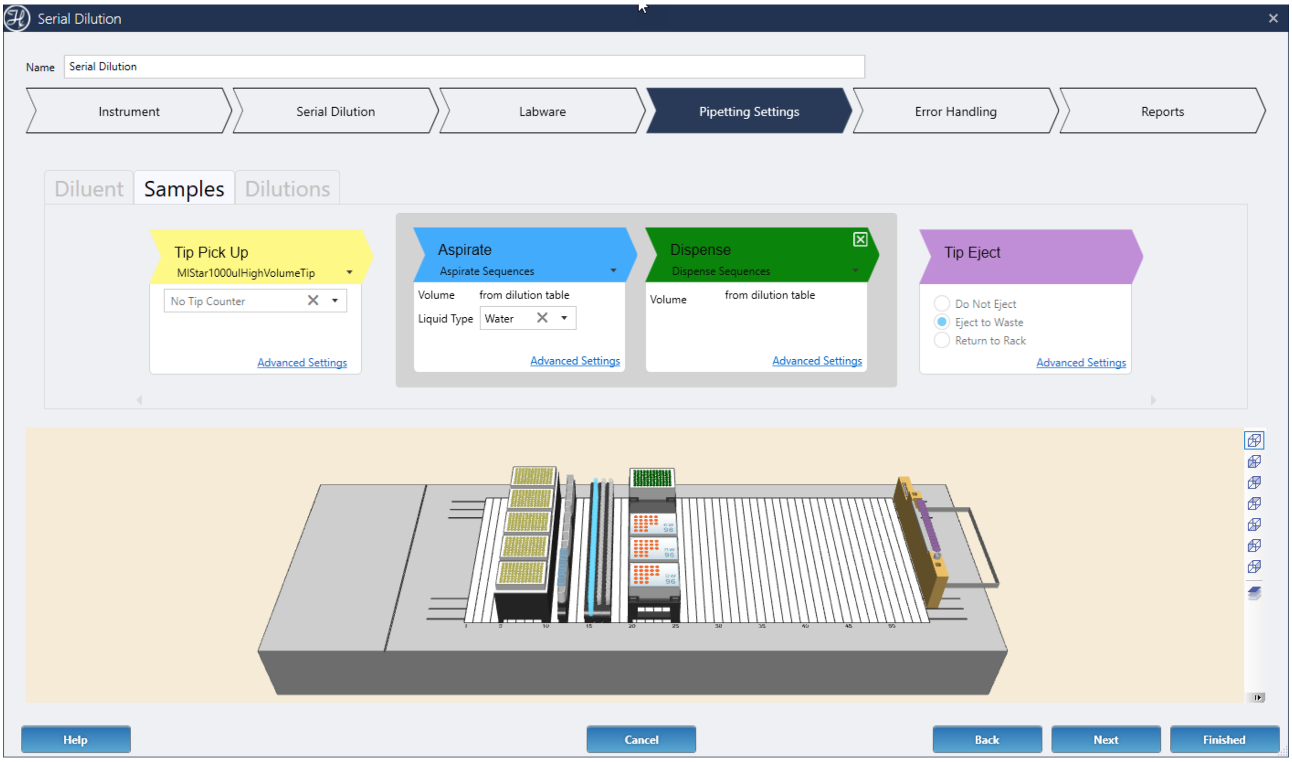


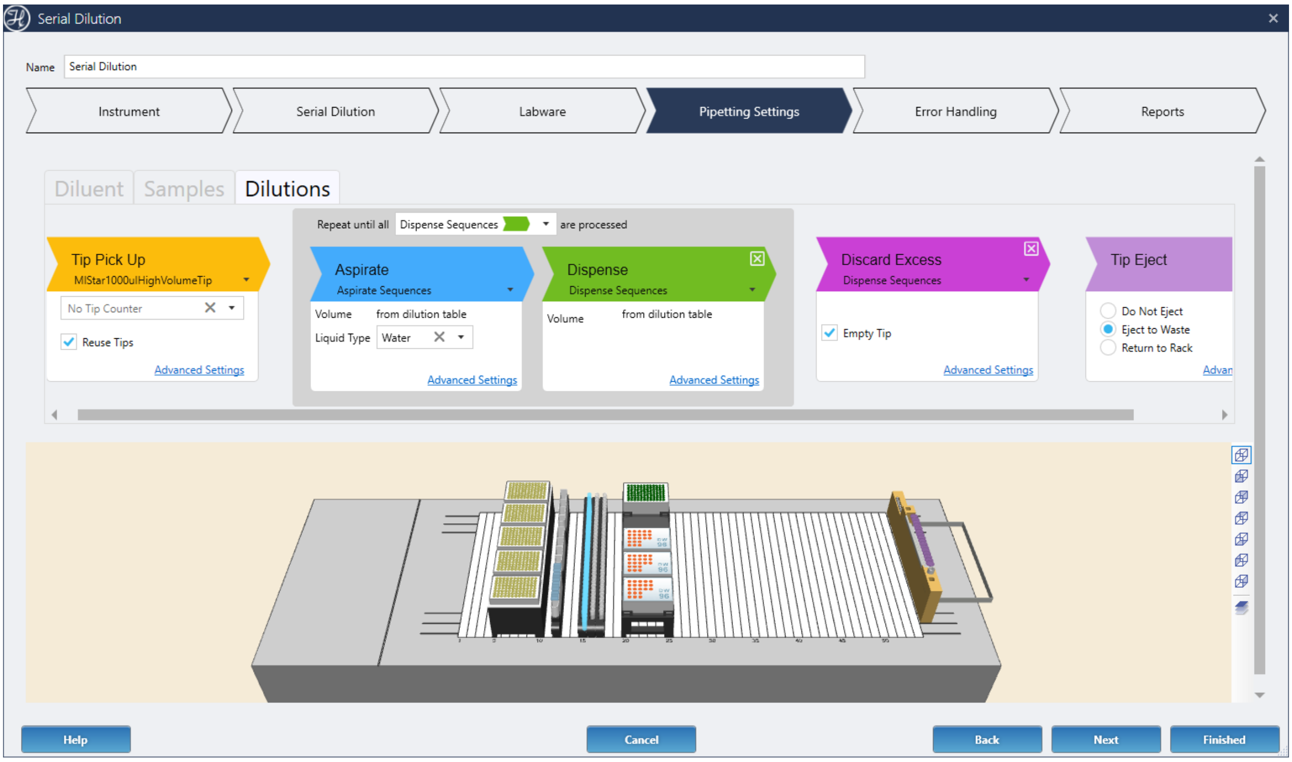
13.7.4.4 Pipetting Settings

The Pipetting Settings screen displays a diagram of the pipetting procedure and steps.

Depending on the serial dilution options selected in the Serial Dilution screen, different tabs will be available. The advanced settings of each step are similar to the Transfer Samples step. Refer to the Transfer Samples > Pipetting Settings section for more details.







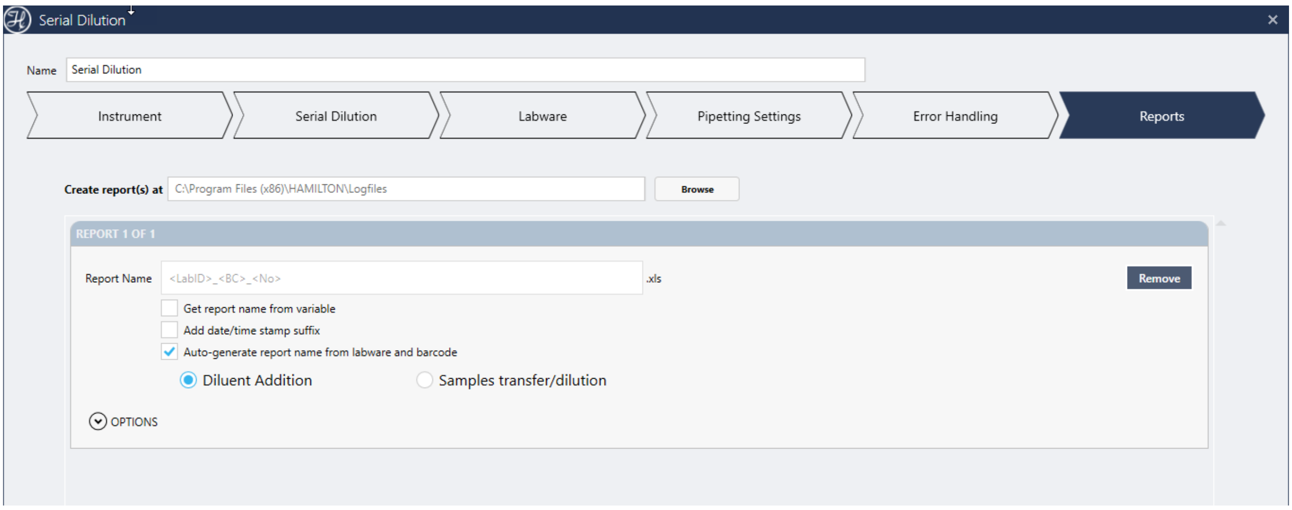
13.7.4.5 Error Handling

This screen has the same settings and options as the Errors screen of the Transfer Samples Power Step. Refer to the Transfer Samples > Error Handling section of this manual for details.

13.7.4.6 Reports

This screen has the same settings and options as the Reports screen of the Transfer Samples Power Step. Refer to the Transfer Samples > Reports section of this manual for details.

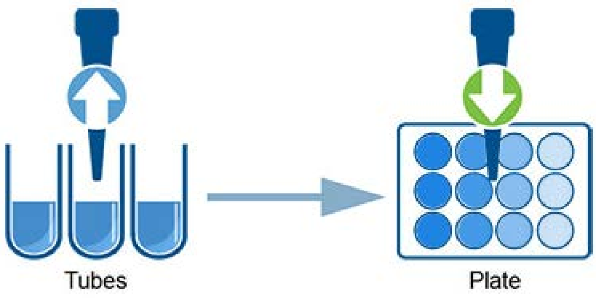
Note that instead of selecting aspirate and dispense sequences to generate a report, the available options are 'Diluent addition' and 'Sample transfer/Dilutions'. The 'Diluent addition' will be grayed-out if the 'Pipette diluent' option is disabled on the Serial Dilution screen.



13.7.4.7 Usage Examples

Example 1: Serial dilution of a sample in tubes

Example 2: Serial dilution of tube samples in a plate



13.7.5 Replicates

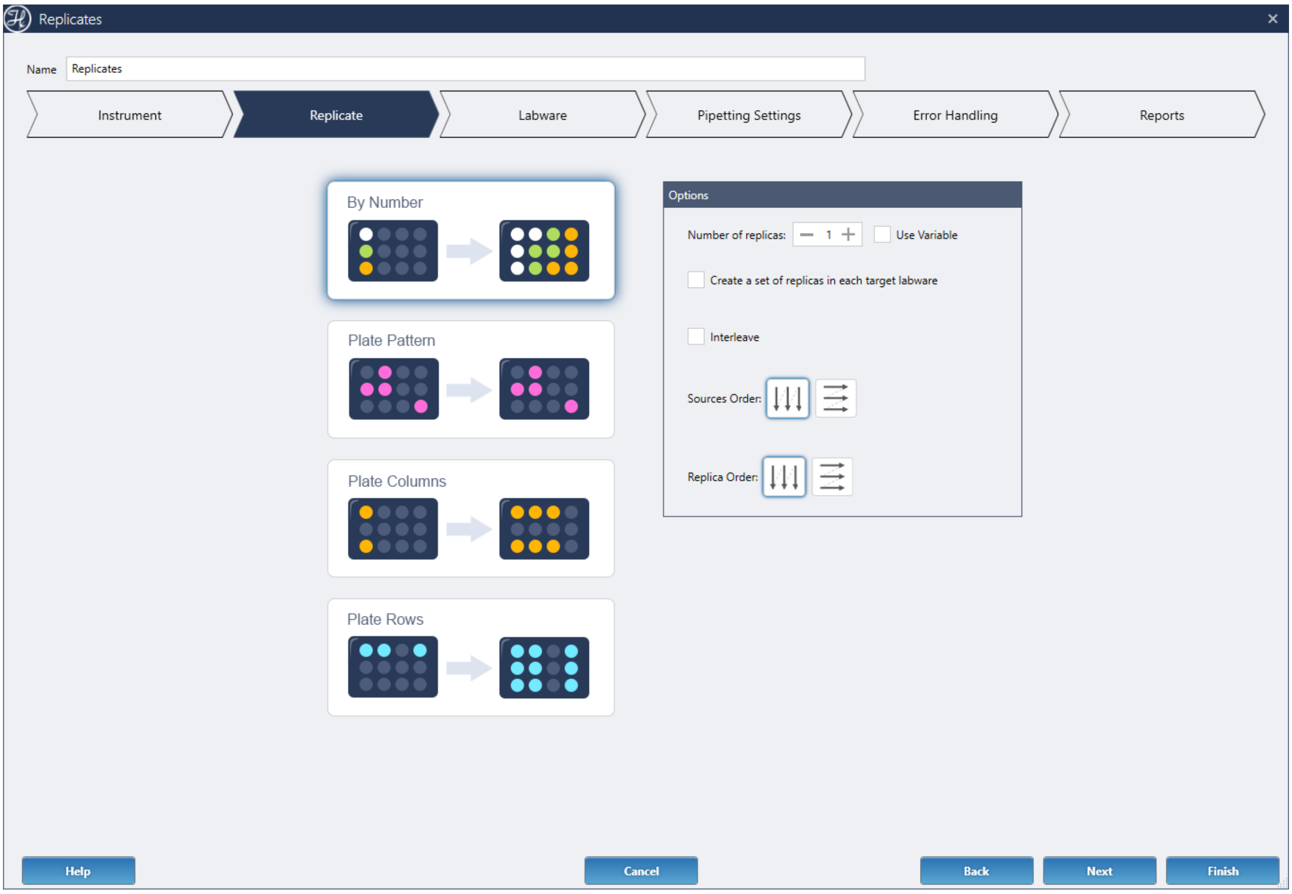
The Replicates Power Step simplifies the process of creating copies of a set of samples or a full pattern of a plate.

13.7.5.1 Instrument

This screen has the same settings and options as the Instrument screen of the Transfer Samples Power Step. Refer to the Transfer Samples > Instrument section of this manual for details.

13.7.5.2 Replicate

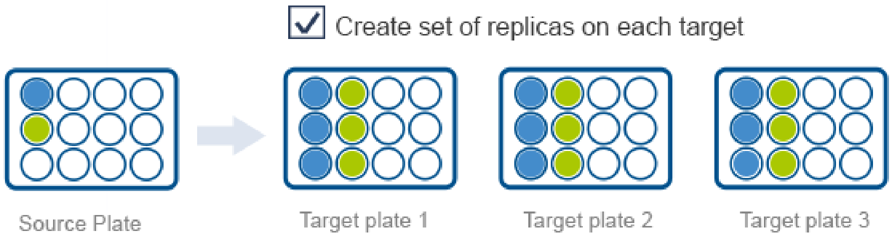
The Replicate screen specifies the different replicate type and settings to perform the samples copy or replication.

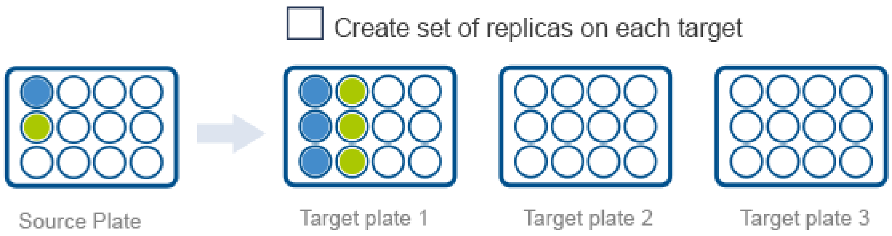


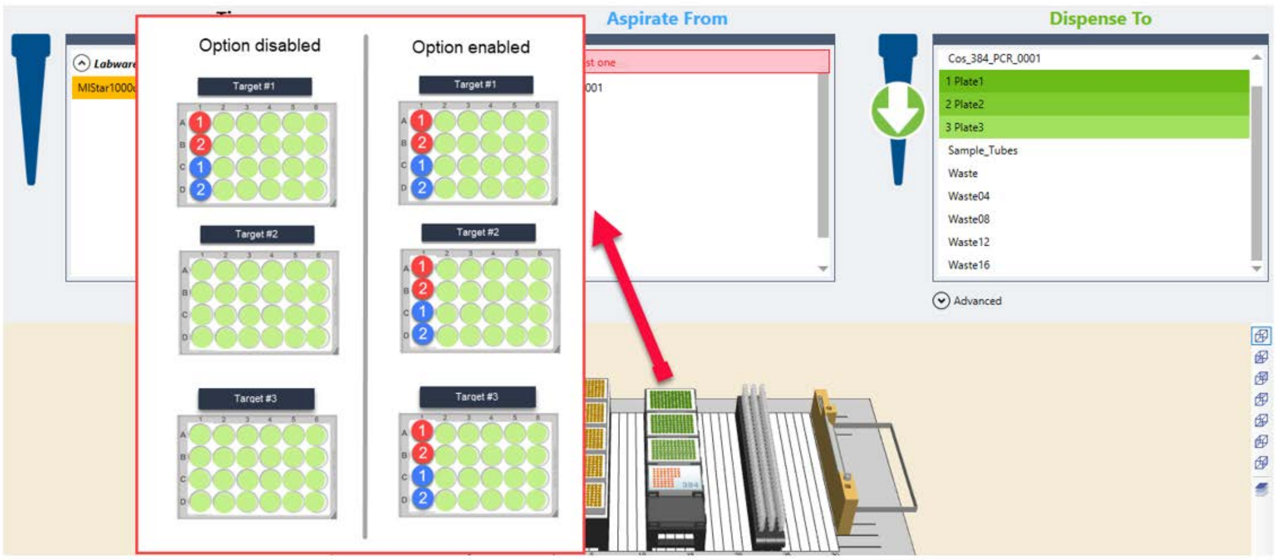
- **By Number**:
  creates n-replicas of each of the wells selected without maintaining the source pattern, with these options:
  - **Number of replicas**
  - **Create a set of replicas on each target labware:**
    Creates a set of n-replicas on each labware included in the 'Dispense To' section of the Labware screen.

    This value will only be considered when using an existing target sequence (Dispense To list) in the Labware screen. Using a custom selection for the targets overrides this setting.

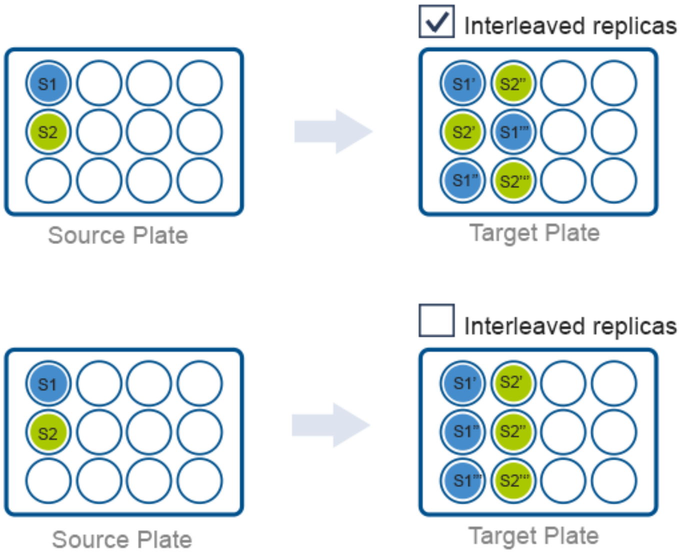
For example, create 3 replicas of each source sample over a sequence with 3 plates.







- **Interleave**:
  if enabled, alternate all replicas of each sample. If disabled, transfer all replicas of each same source altogether as a block.



- **Sources order**:
  determines if the sources defined in 'Aspirate from' of the Labware screen should be processed by columns or rows
- **Replicas order**:
  determines if the targets defined in 'Dispense To' of the Labware screen should be processed by columns or rows

- **Plate Pattern**:
  creates a copy of the source plate pattern into one or more target plates.
  - **Number of replicas**:
    This value will only be considered when using an existing target sequence (Dispense To) in the Labware screen.

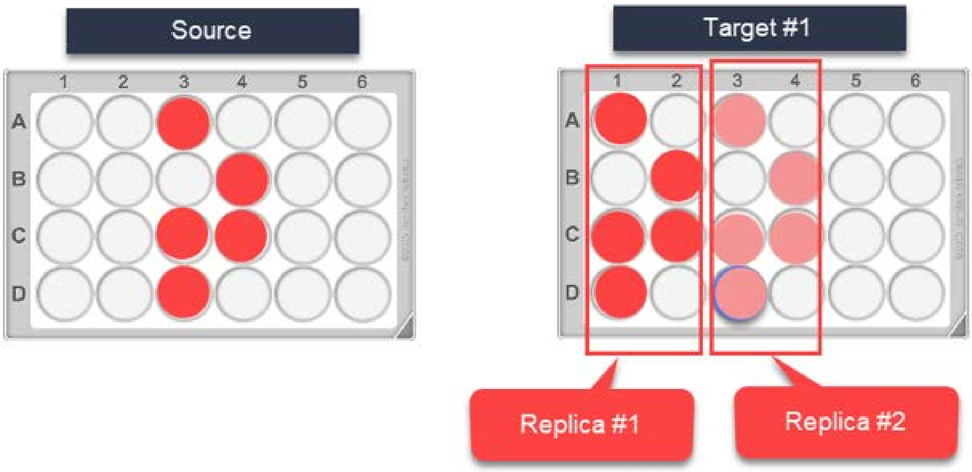
    When using custom selection for the targets (Dispense To) on the Labware screen, the number of replicas will be determined by the manual selection, depending on the different labware selected / highlighted. (1 target labware selected= 1 replica, 2 target labware selected= 2 replicas, etc.).
  - **Create a set of replicas on each target abware**:
    same behavior as in 'By Number' described above.

- **Plate Columns**:
  creates n-replicas of the pattern of one or more plate columns from a single source plate.

  When using an existing target sequence from the Dispense To list (no custom selection), the replicas with be pipetted in order in the columns available in the sequence, starting in the first column available.

  When using custom selection for target sequence the replicas with be pipetted in the columns selected, which may or may not be contiguous.

  If the source sequence to be replicated contains two or more columns, these columns will be treated as a block.



If the target labware does not contain position IDs that match the sources, these non-matching source wells will be skipped/ignored in this particular target labware.

- **Number of replicas**:
  This value will only be considered when using an existing target sequence (Dispense To) on the Labware screen.

  When using custom selection for the targets (Dispense To) on the Labware screen, the number of replicas will be determined by the manual selection, depending on the different labware selected / highlighted.

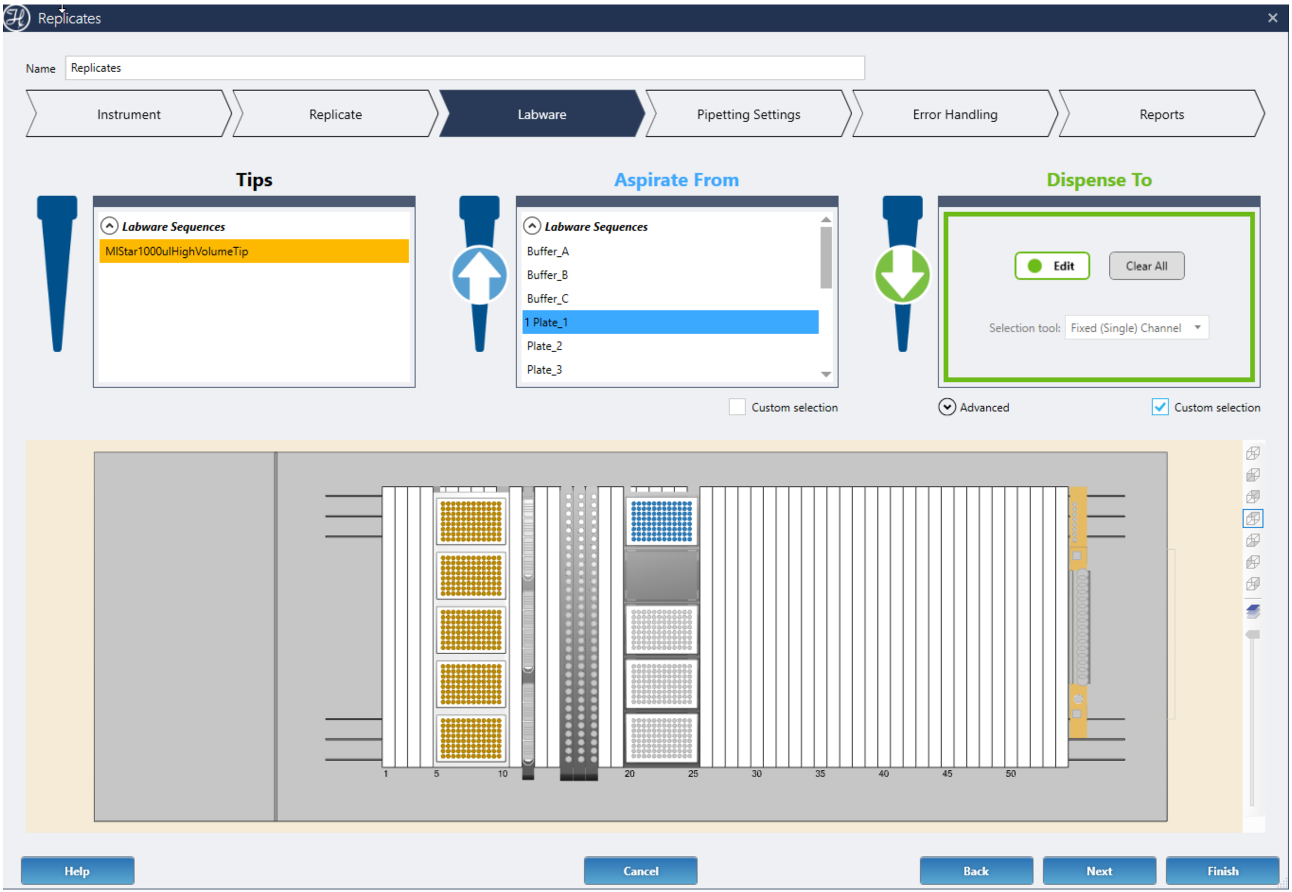
- **Copy plate pattern to each target labware**:
  same behavior as in ‘By Number’ described above.

- **Plate Rows**:
  creates n-replicas of the pattern of one or more plate rows from a single source plate.

  The options and behavior of this step are the same as the ‘Plate Columns’ described above but based on plate rows selections.

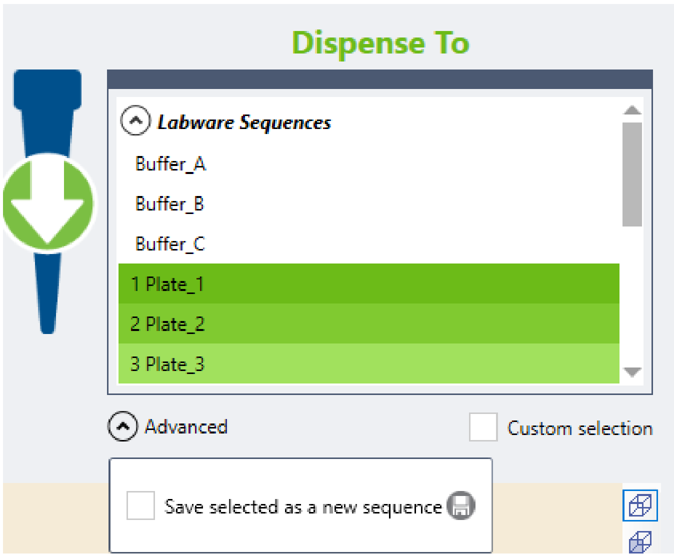
13.7.5.3 Labware

The Labware screen displays the deck layout and the tips, aspirate and dispense sequences used to pipette the replicates.

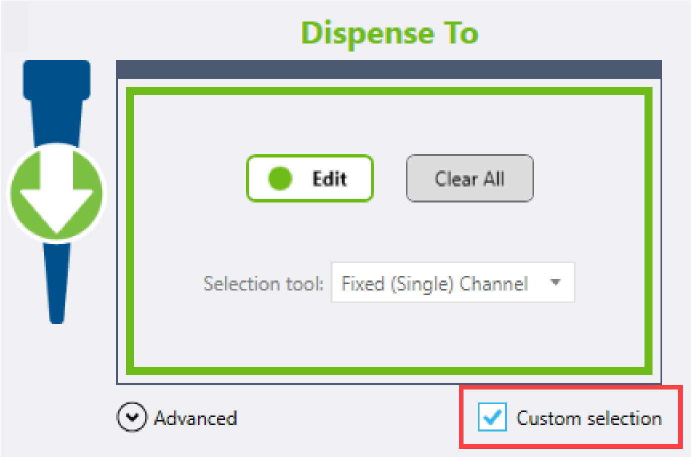


Only one sequence can be selected in the ‘Aspirate From’ section, which will contain the samples to be replicated.

Multiple sequences can be selected in the ‘Dispense To’ section. Optionally, under the Advanced dialog, the selected sequences can be saved into a single new sequence that can be used in the method later on.

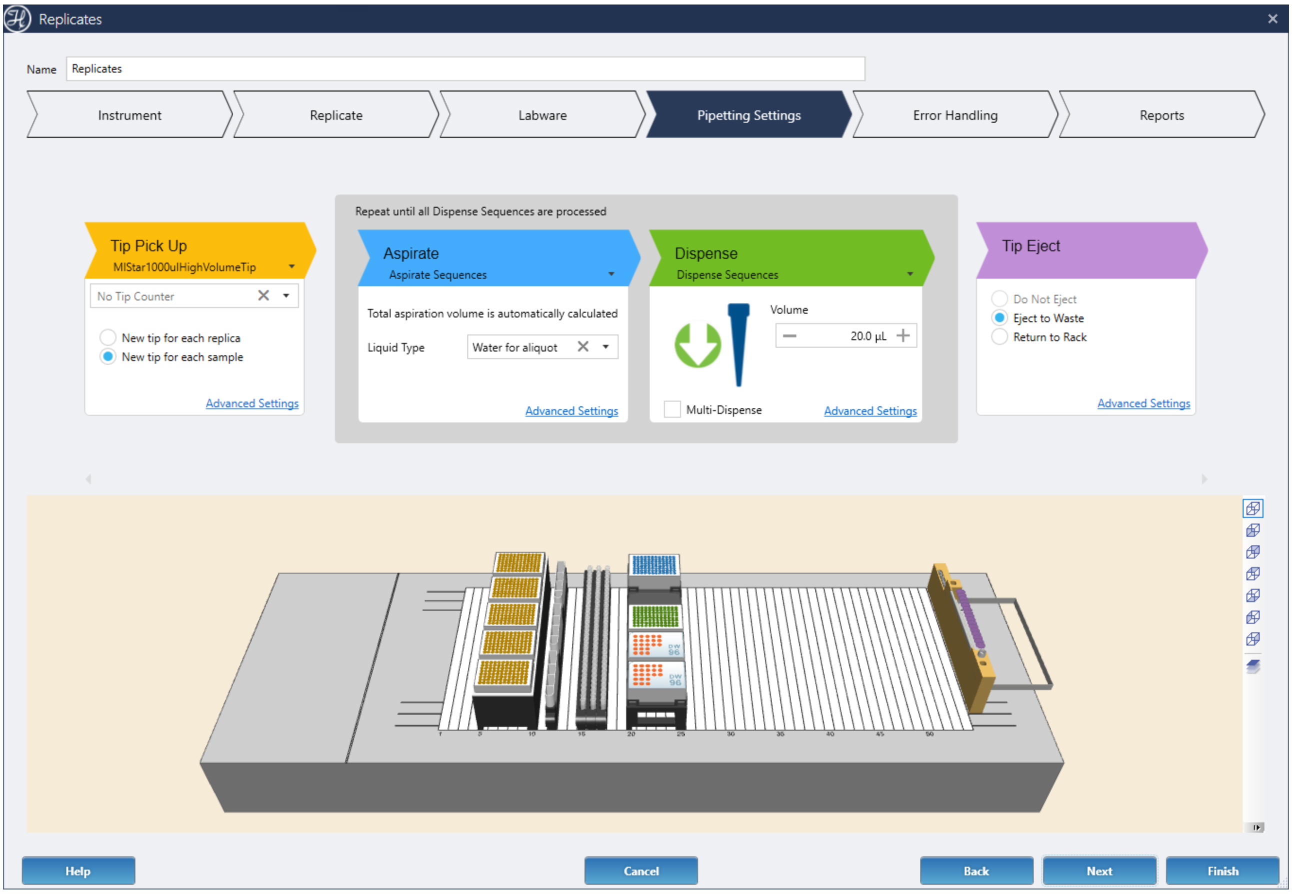


Alternatively, the sources and targets can be selected directly from the deck layout below by enabling the ‘Custom selection’ checkbox.



13.7.5.4 Pipetting Settings

The Pipetting Settings screen displays a diagram of the pipetting procedure and steps.



Tip Pick Up

This step is the same as the Tip Pick Up step of the Transfer Samples Power Step. Refer to the Transfer Samples > Pipetting Settings > Tip Pick Up section of this manual for details.

Note that the Replicate Tip Pick Up only offers two options:

- **New tip for each replica**: use a new tip for each transfer
- **New tip for each sample**: reuse the same tip for all the transfers of the same sample in the Power Step

Aspirate

This step is the same as the as the Aspirate step of the Add Reagent Power Step. Refer to the Add Reagent > Pipetting Settings > Aspirate section of this manual for details.

Dispense

This step is the same as the as the Dispense step of the Add Reagent Power Step. Refer to the Add Reagent > Pipetting Settings > Dispense section of this manual for details.

Tip Eject

This step is the same as the Tip Eject step of the Transfer Samples Power Step. Refer to the Transfer Samples > Pipetting Settings > Tip Eject section of this manual for details.

13.7.5.5 Error Handling

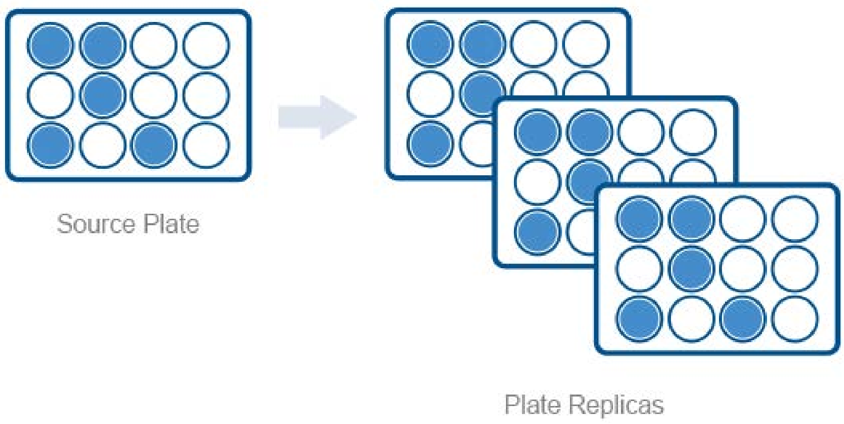
This screen has the same settings and options as the Errors screen of the Transfer Samples Power Step. Refer to the Transfer Samples > Error Handling section of this manual for details.

13.7.5.6 Reports

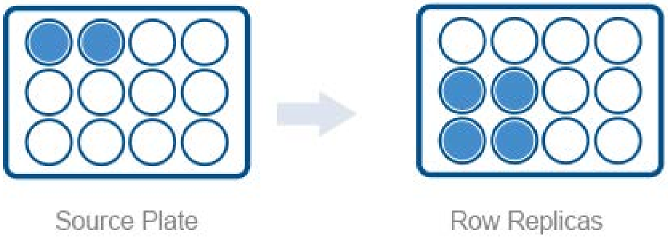
This screen has the same settings and options as the Reports screen of the Transfer Samples Power Step. Refer to the Transfer Samples > Reports section of this manual for details.

13.7.5.7 Usage Examples

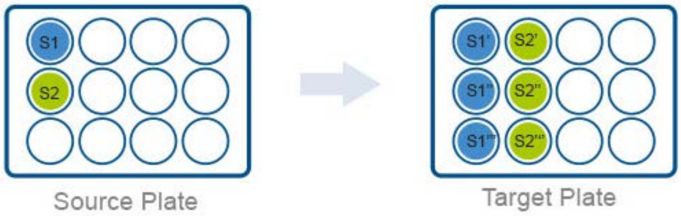
Example 1: Create 3 copies of a plate, keeping the wells pattern



Example 2: Create 2 copies of a row in a plate into another 2 rows of another plate



Example 3: Create 3 replicas of 2 samples into another plate



Remember to disable the ‘Interleave’ option on the Replicates screen to keep all replicas of the same sample in contiguous wells of the target plate.

13.7.6 Hit Picking

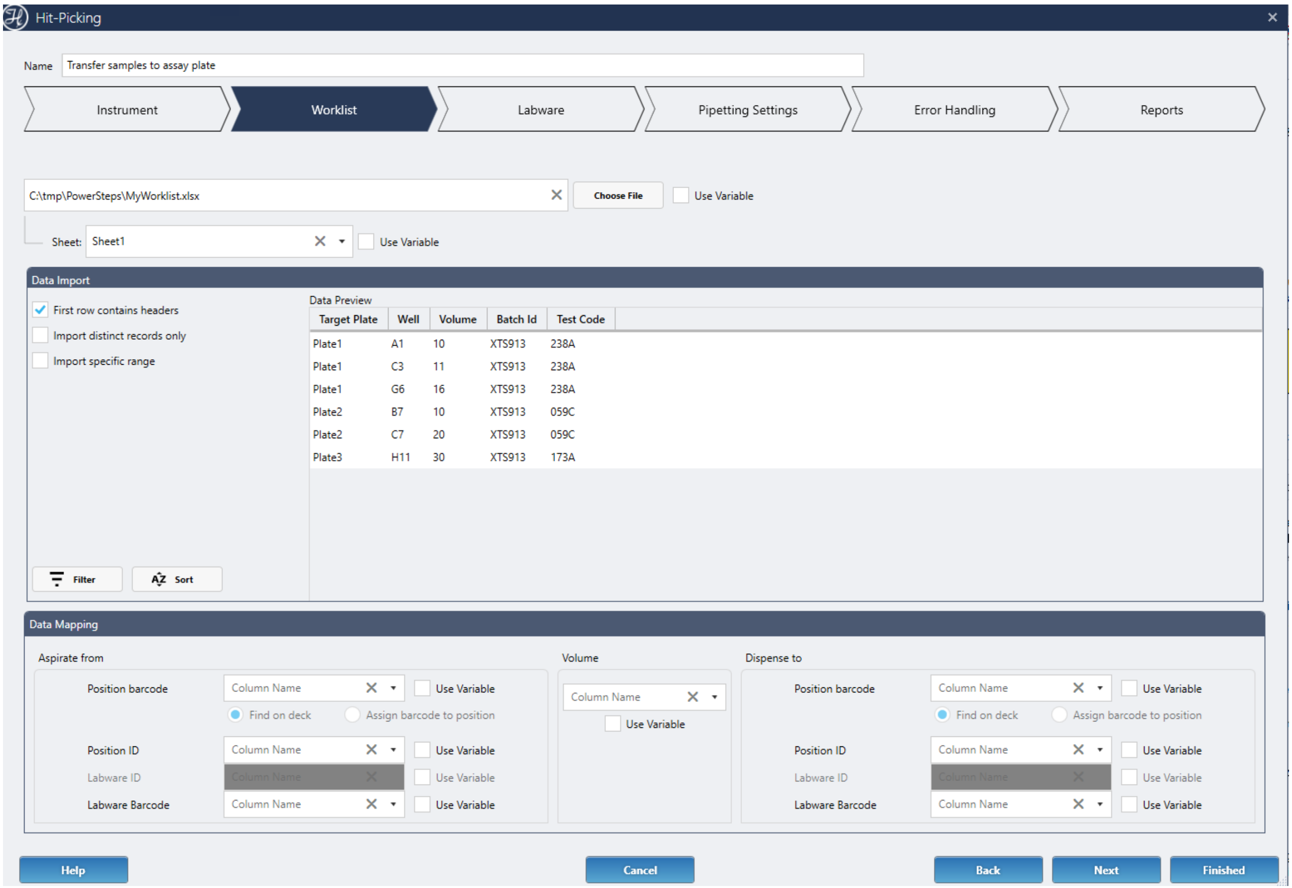
The Hit Picking step is very similar to the Transfer Samples step, with the option of getting the sources, destinations and/or transfer volume information from a file (worklist).

13.7.6.1 Instrument

This screen has the same settings and options as the Instrument screen of the Transfer Samples Power Step. Refer to the Transfer Samples > Instrument section of this manual for details.

13.7.6.2 Worklist

The Worklist screen will define where the data is coming from, as well as what data is being provided (e.g., sources, destinations, barcodes, volumes).



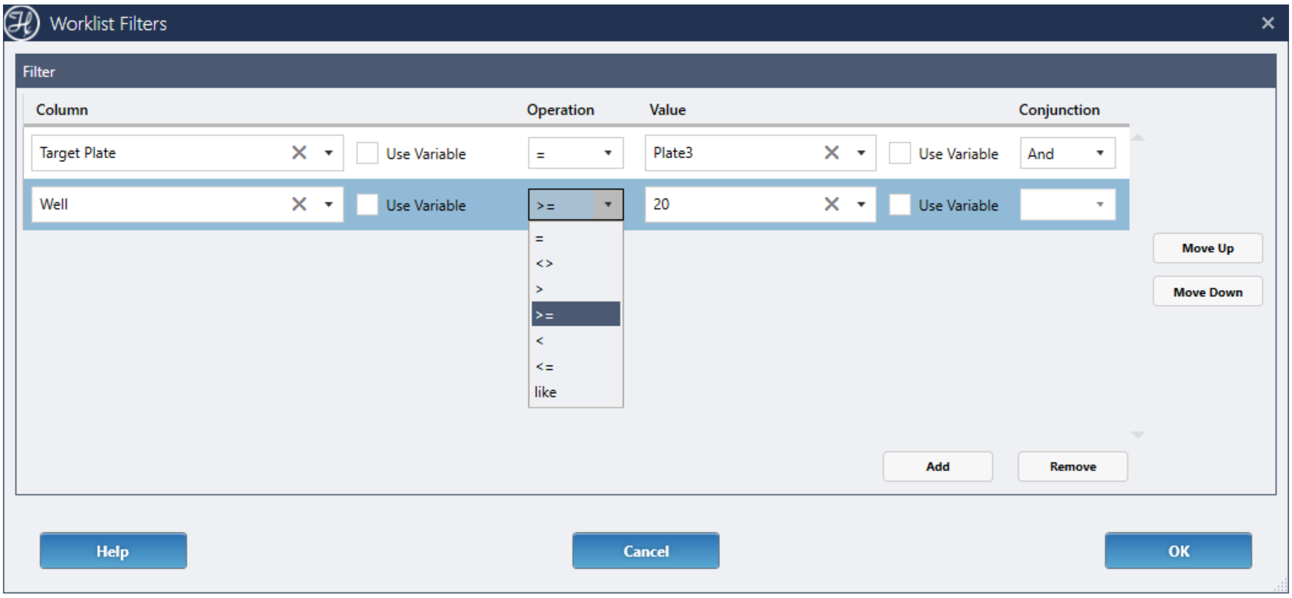
The Worklist step supports Excel (.xls, .xlsx) and structured ASCII text files (.csv). The worklist filename can be optionally retrieved from a variable in the method.

The 'Data Import' section contains different filtering and sorting options to restrict the records to be processed in the step. It also provides a preview of the records that will be processed.

- **First row contains headers**:
  Select this option if the first row of the worklist contains the labels for each column
- **Import distinct records only**:
  Eliminates duplicate rows and avoids processing the same row twice

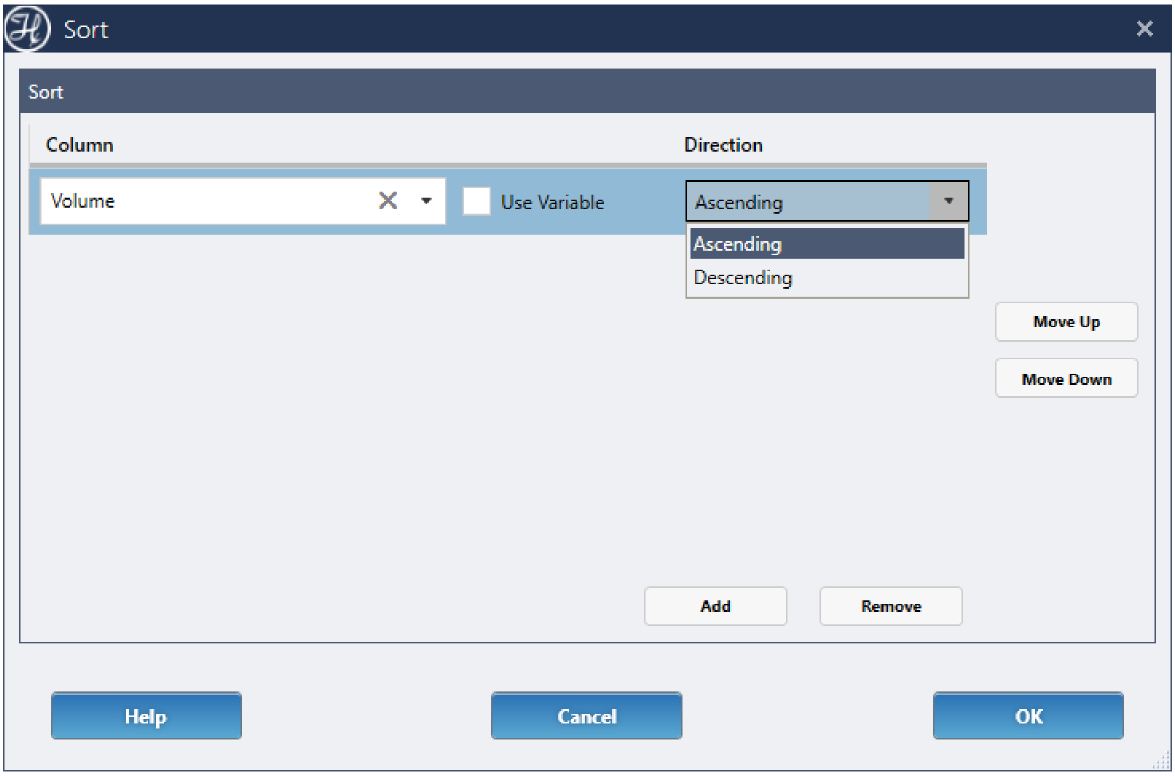
- **Import specific range**:
  Determines the start and end row of the worklist to be processed. Any rows outside this range will be ignored
- **Filter**:
  Adds custom rules to filter the worklist data. Multiple filters can be combined for complex filtering

  The step will only process the entries of the worklist that satisfy all the filters set.



- **Sort**:
  Adds custom rules to sort the worklist data. Multiple sorting rules can be combined for complex sorting

  The step will process the entries in the sorted order of the worklist.



The 'Data Mapping' section defines what the data in each column means. This information will determine what will be executed in the step. All fields in this section are optional.

- Aspirate from / Dispense To
  - **Position Barcode:**
    the barcode that identifies a sample well or tube.

    Find on deck, if you read or assigned the barcode to a sample in a previous step. The software will automatically locate it on the deck layout.

    Assign barcode to position, the software will assign the barcode to the positions used in the Aspirate From or Dispense To. These positions can be defined by mapped Position Id and Labware Id data in the worklist, or by sequences selected in the Labware screen.
  - **Position ID**:
    the position in a plate (A1, B1) or in a tube rack (1, 2, 3). If this field is blank, the data will be retrieved from the sequences selected on the Labware screen.
  - **Labware** ID:
    the unique labware ID generated by the software when labware is placed on the deck layout editor. If the worklist contains a column with labware IDs, select the column from the drop-down menu.

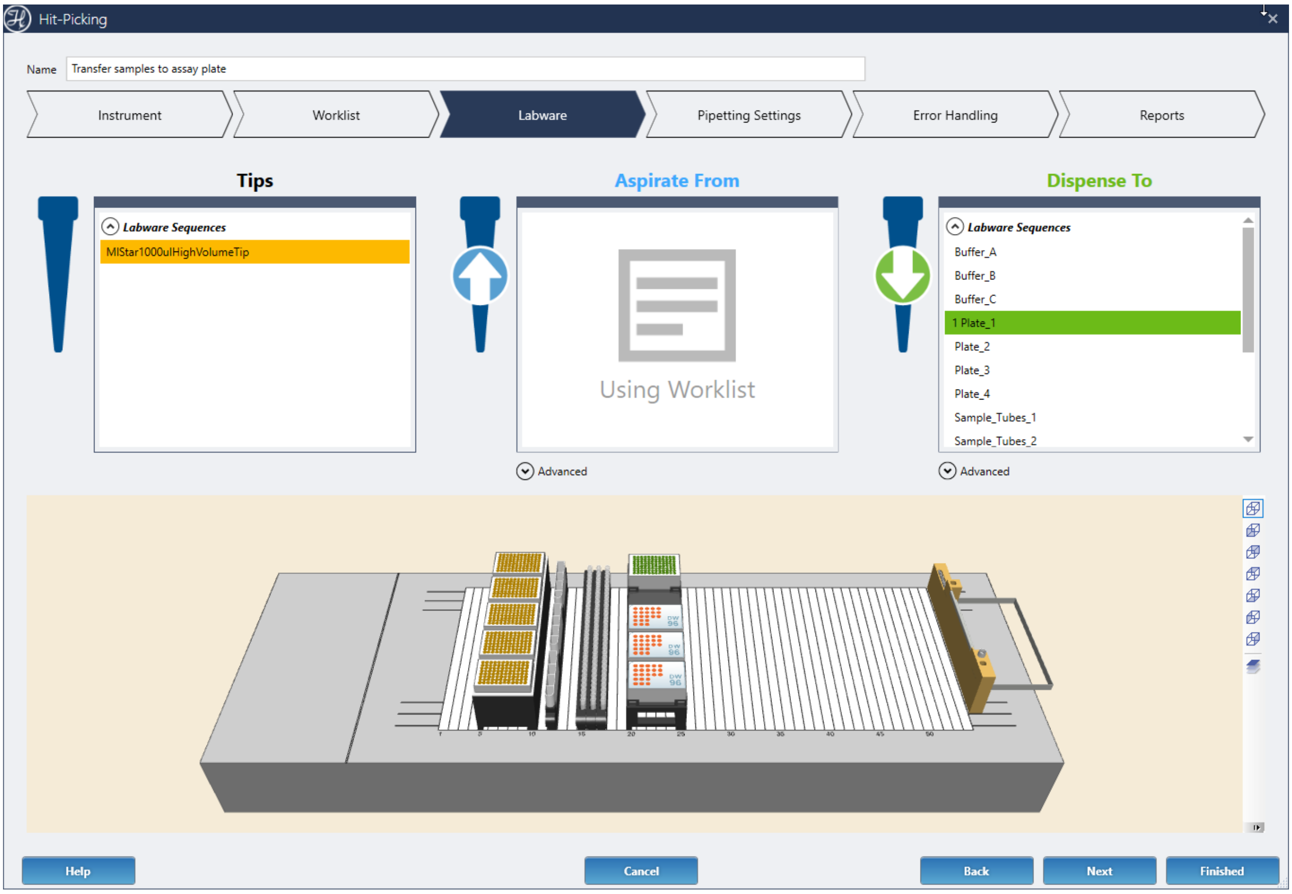
    If this field is blank, this data will be retrieved from the selection on the Labware screen.
  - **Labware Barcode**:
    the barcode that identifies a tube rack, plate, or carrier.

- **Volume**:
  the column with the volumes to be transferred in the step. Each volume given in the worklist will be used in order for each well/tube transfer

  If the worklist does not provide the volume, enter the volume to pipette on the Pipetting Settings page.

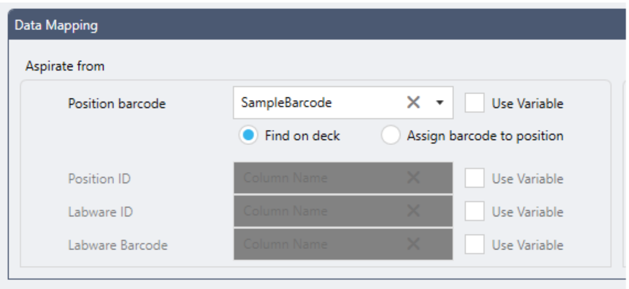
13.7.6.3 Labware

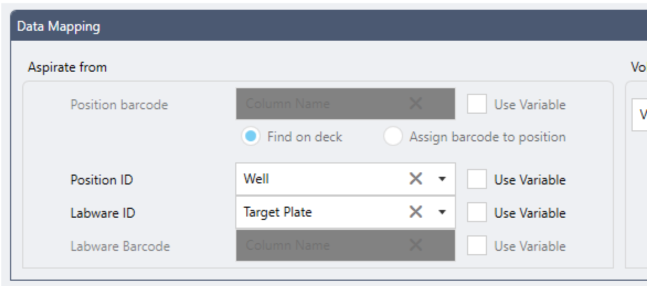
Like in the Transfer Samples Power Step, the Labware screen displays the deck layout and the tips, aspirate, and dispense sequences used to perform the pipetting.



The ‘Aspirate From’ or ‘Dispense To’ will replace the sequence list with ‘Using worklist’ if either of these combinations of data have been mapped on the Worklist screen:

- Position Barcode + Find on deck
- Position ID + Labware ID



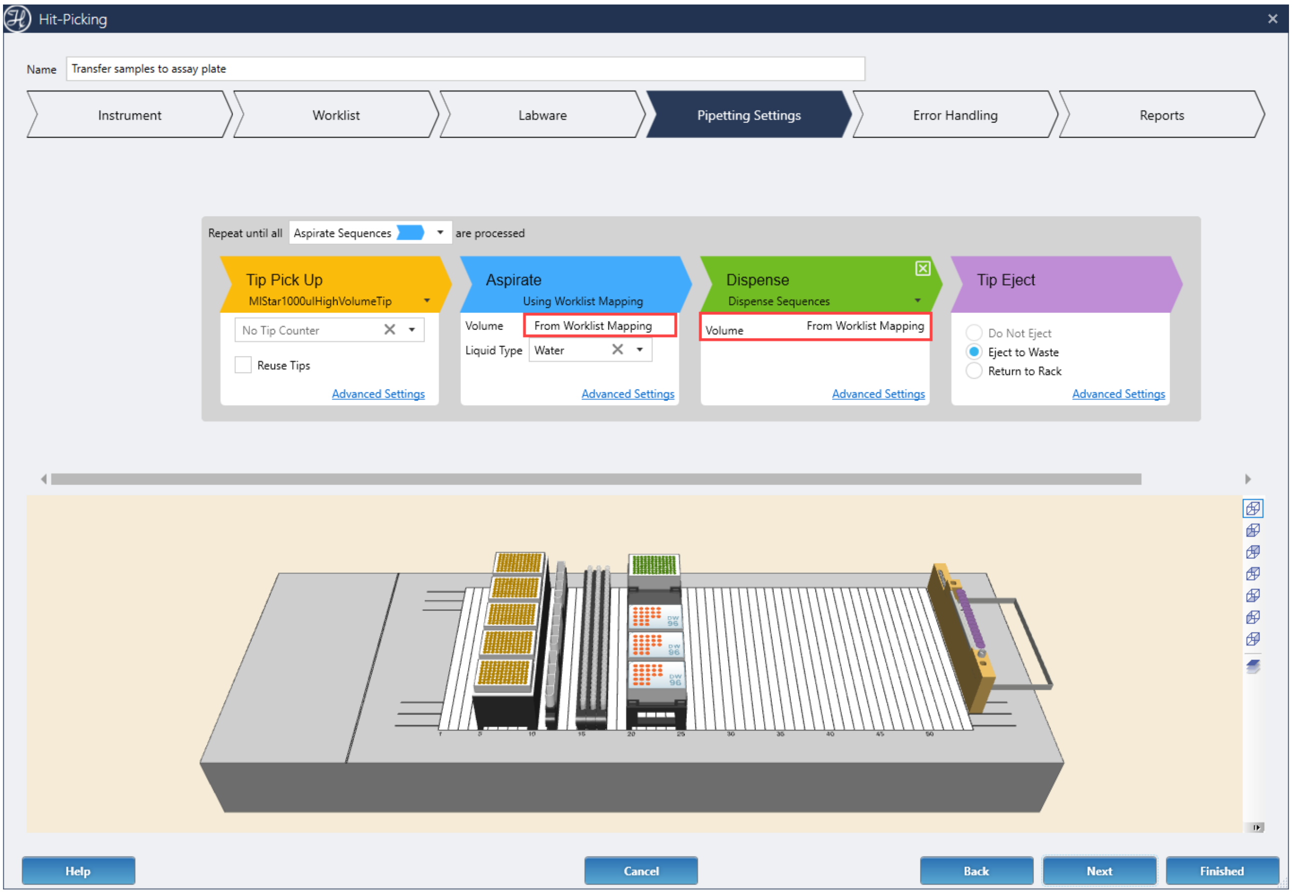


All the other settings and Advanced options are the same as in the Transfer Samples Power Step. Refer to the Transfer Samples > Labware section of this manual for details.

13.7.6.4 Pipetting Settings

This screen has the same settings and options as the Pipetting Settings screen of the Transfer Samples Power Step. Refer to the Transfer Samples > Pipetting Settings section of this manual for details.

If the volume has been mapped on the Worklist screen, it will be indicated in the Aspirate and Dispense chevrons.



13.7.6.5 Error Handling

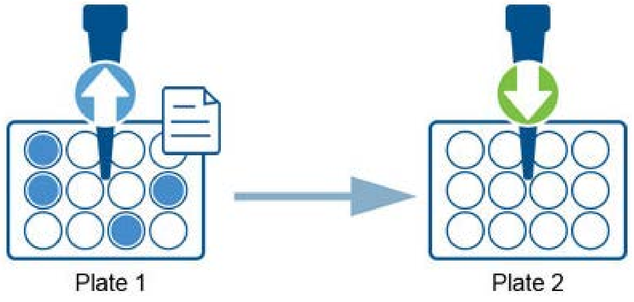
This screen has the same settings and options as the Errors screen of the Transfer Samples Power Step. Refer to the Transfer Samples > Error Handling section of this manual for details.

13.7.6.6 Reports

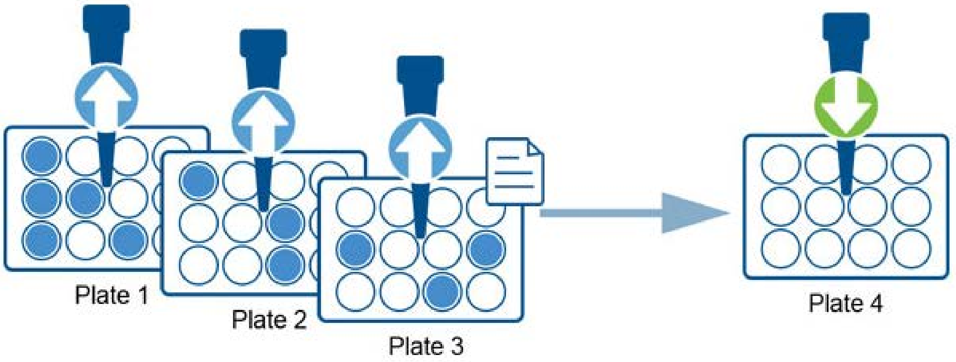
This screen has the same settings and options as the Reports screen of the Transfer Samples Power Step. Refer to the Transfer Samples > Reports section of this manual for details.

13.7.6.7 Usage Examples

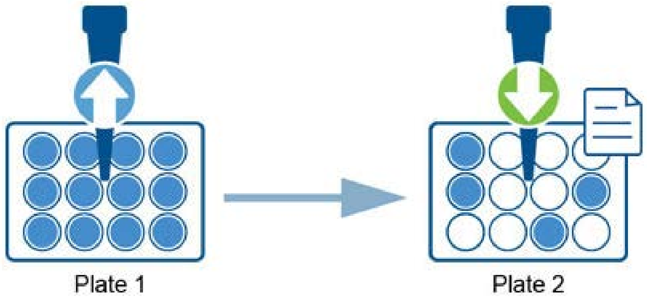
Example 1: Transfer hits from a single plate using a worklist



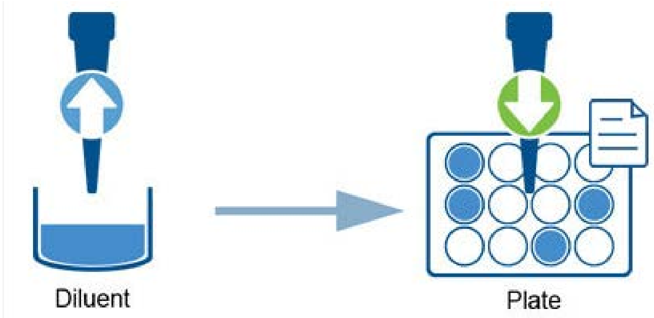
Example 2: Transfer hits from 3 plates using a worklist



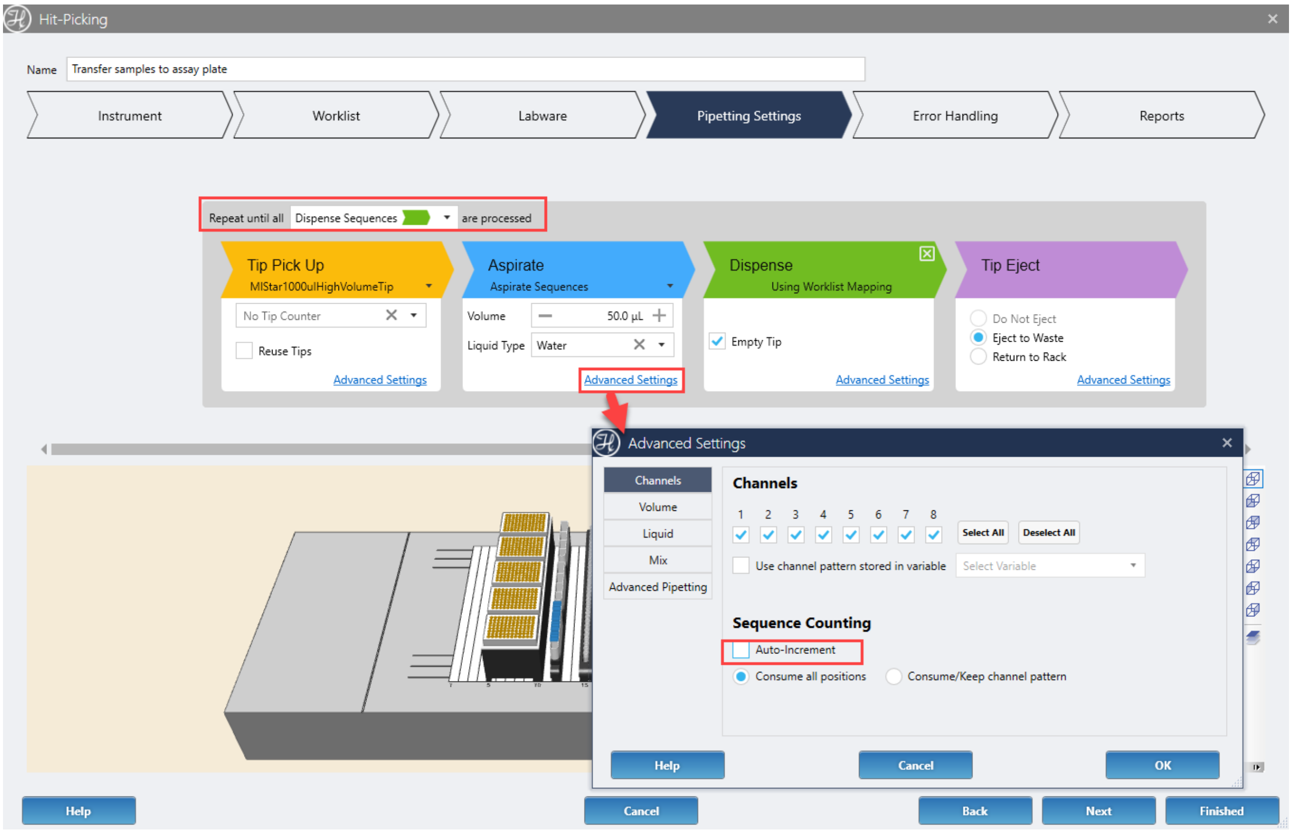
Example 3: Transfer samples to specific target wells defined in a worklist



Example 4: Add diluent to specific target wells defined in a worklist

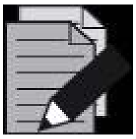


Remember to set the process to repeat until all target wells (the samples Dispense sequence) are processed. Also disable the auto-increment of the diluent aspirate to ensure that it always comes back to the same reservoir positions in each loop of the Power Step.



13.7.7 Load and Match

The Load and Match Power Step reads a worklist with tube barcodes to be pipetted in later steps in the method and determines if those barcodes are present on the deck (=Match) or not. If the system has an automated loading and scanning device (STAR Autoload, VANTAGE IDL), the step can optionally Load and scan the carriers selected by the user within the Power Step, prior looking for matches from the given barcodes worklist.



NOTE

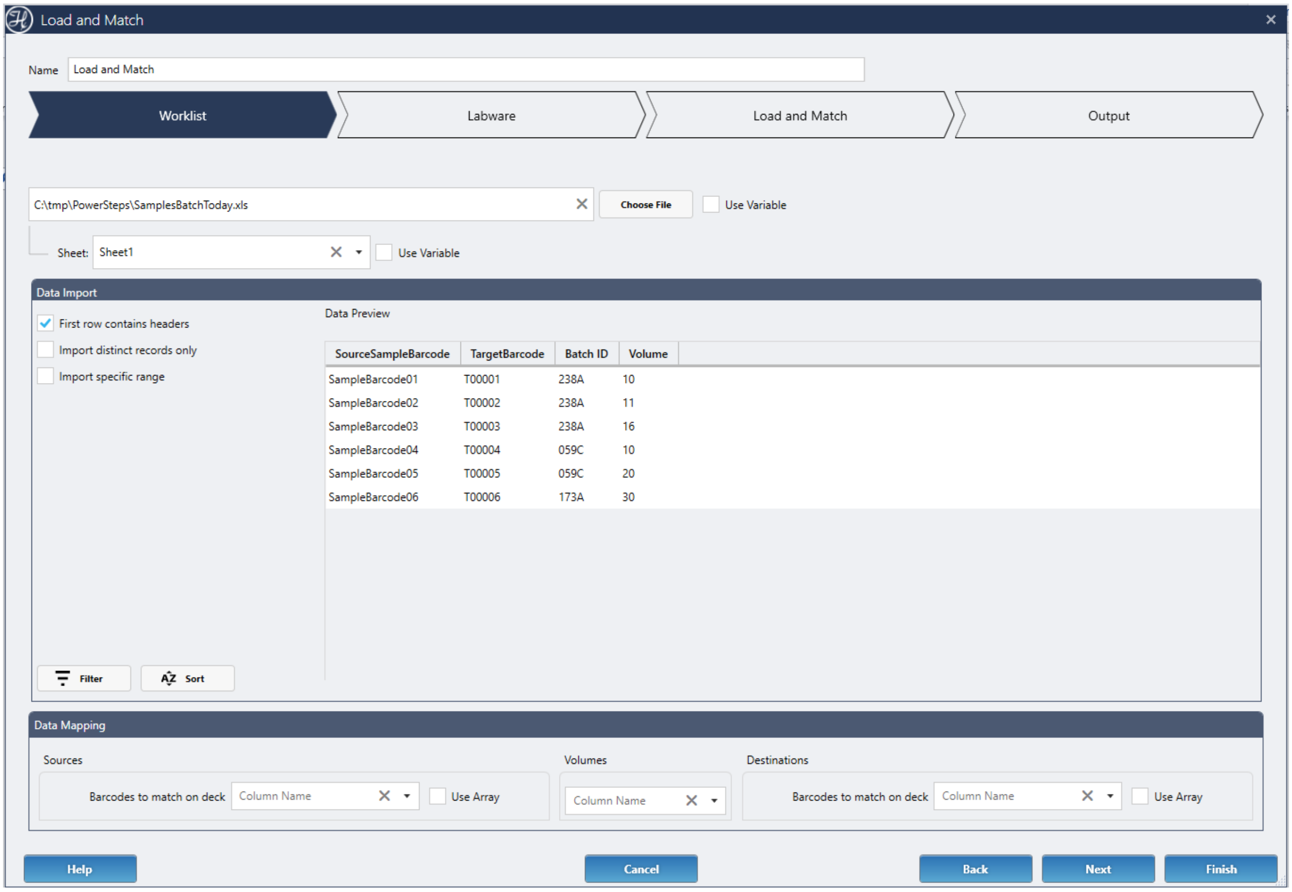
<u>This step does not perform any pipetting</u>. At the end of the Load and Match execution, the step will return a sequence of the deck positions with a match in the worklist, ready to be used in other pipetting or Power Steps within the method.

13.7.7.1 Instrument

This screen has the same settings and options as the Instrument screen of the Transfer Samples Power Step. Refer to the Transfer Samples >Instrument section of this manual for details.

13.7.7.2 Worklist

The Worklist screen defines the data source as well as what type of information is being provided (source barcodes, destination barcodes, and/or volumes).

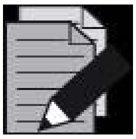


The Worklist screen supports Excel (.xls, .xlsx) and structured ASCII text files (.csv). The worklist filename can be optionally retrieved from a variable in the method.

The '**Data Import'** section contains different filtering and sorting options to restrict the records to be processed in the step. It also provides a preview of the records that will be processed.

- **First row contain headers**: Select this option if the first row of the worklist contains the labels for each column
- **Import distinct records only**: Eliminate duplicated rows and avoid processing the same row twice
- **Import specific range**: Determine the start and end row of the worklist to be processed. Any rows outside this range will be ignored
- **Filte**r: Add custom rules to filter the worklist data. Multiple filters can be combined for complex filtering. The step will only process the entries of the worklist that satisfy all the filters set.
- **Sort**: Add custom rules to sort the worklist data. Multiple sorting rules can be combined for complex sorting. The step will process the entries in the sorted order of the worklist.

The '**Data Mapping'** section defines what column in the worklist contains the source barcodes, destination barcodes or volumes.

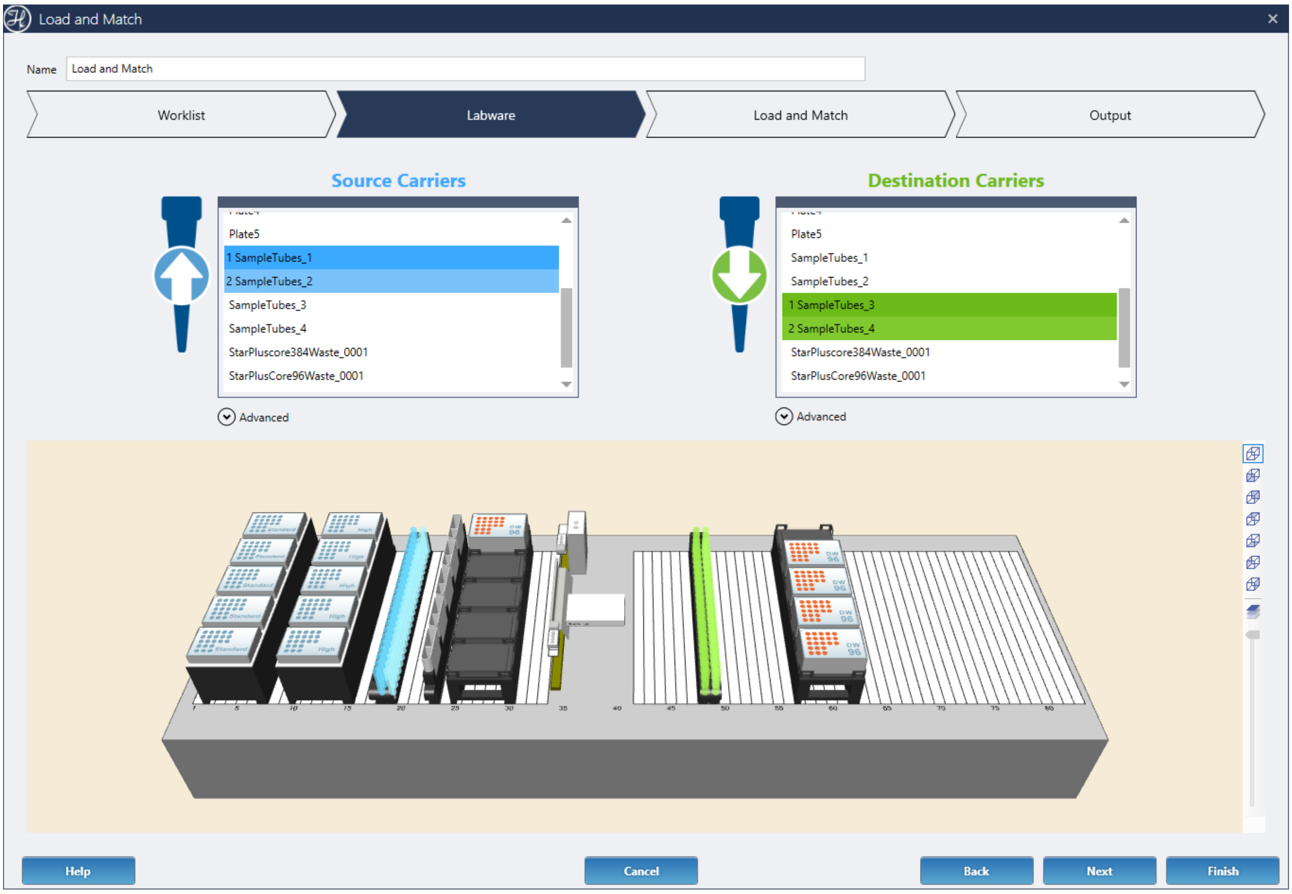


NOTE

At least the source or destination barcodes must be mapped.

13.7.7.3 Labware

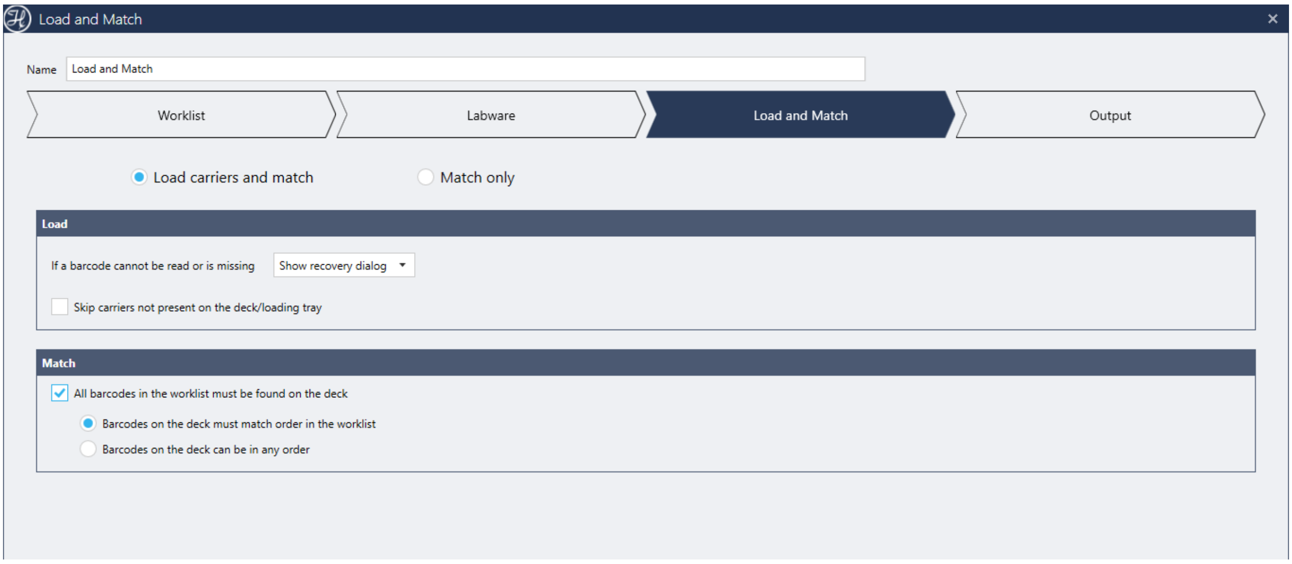
This screen defines in what tube carriers on the deck shall the software look at runtime for tube barcode matches from the given worklist. The option to select source and/or destination carriers depends on the data mapped in the previous worklist screen.



13.7.7.4 Load and Match

In order to look for barcode matches from the worklist on the system deck at runtime, the barcoded tubes present on the deck must be either previously scanned or assigned manually.

The Load and Match screen provides the option to automatically load and scan the carriers selected in the labware screen, whenever an automated device is available (STAR Autoload, VANTAGE IDL).



- **Load carriers and Match**: The step will load and scan the carriers selected in the labware screen and then look for barcode matches from the given worklist on those carriers.
- **Match only**: The system will only look for barcode matches from the given worklist on the deck. Select this option if the tube barcodes have been already loaded, scanned, or manually assigned in previous steps in the method.

LOAD options:

- <u>Skip carriers not present on deck/loading tray:</u> If this option is checked, the system will only load the carriers selected in the labware screen that are physically detected on the deck/loading tray.
  If unchecked, the system will request to load all carriers selected in the labware screen.

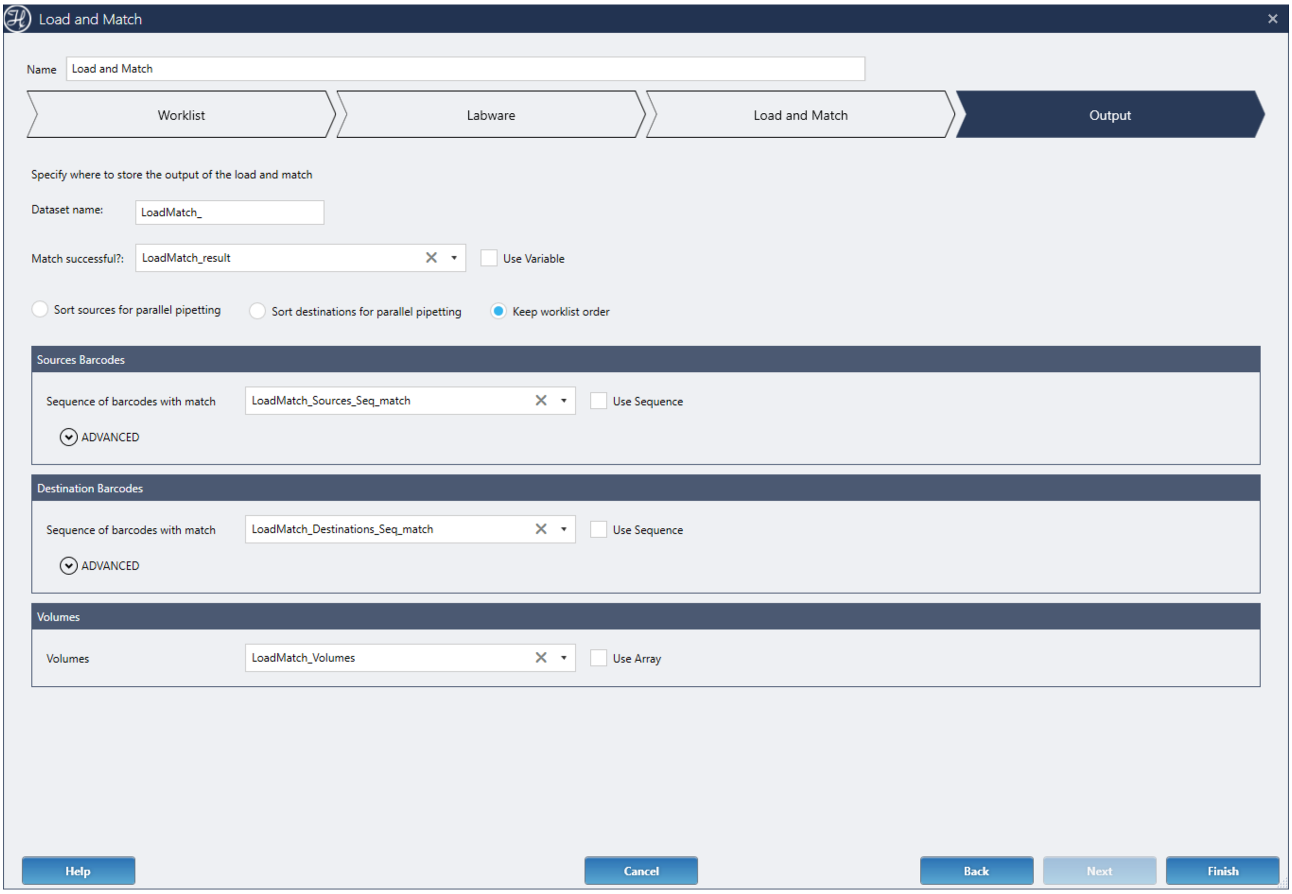
MATCH options:

- <u>All barcodes must be found:</u> If this option is checked, all barcodes provided from the worklist mapped as source or destination barcodes must be found on the deck at runtime.

- <u>Barcodes on the deck must match the order in the worklist :</u> The barcodes found on the deck (from rear to front and left to right) must be located in the same order as they were provided in the worklist. This can be helpful if the samples must be processed in a specific order.
- <u>Barcodes on the deck can be in any order</u>

13.7.7.5 Output

This screen provides the results of the match comparison of the barcodes from the given worklist and the barcodes loaded/found on the carriers from the Labware screen. The result is based on the order/match conditions set in the Load and Match screen.



The screen has multiple fields and options:

Dataset name:

Prefix to automatically populate all the output variables and sequences names generated by the step. Each individual output field can still be renamed if desired.

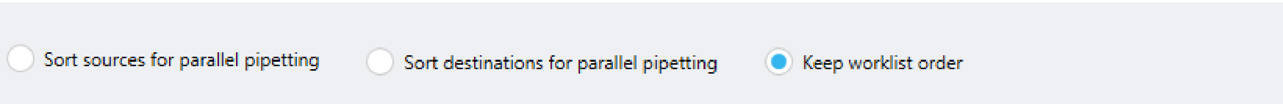
Match successful?:

It will return 1 (true) or 0 (false), depending on the fulfilled conditions set in the Load and Match screen.

Match Successful? Return Values				
All Barcodes Must Be Found	**Match Order**	**Data mapped from worklist**	**Data mapped from worklist**	**Data mapped from worklist**
		Destination only	**Source only**	**Source and Destination**
No	**N/A**	True (1) if at least 1 found, else False (0)	True (1) if at least 1 found, else False(0)	True (1) if at least 1 given pair of source/destination found, else False(0)
Yes	**Order as in Worklist**	True (1) if all found in the worklist order, else False (0)	True (1) if all found in the worklist order, else False(0)	True (1) if all found in the worklist order, else False(0)
Yes	**Any Order**	True (1) if all found, else False (0)	True (1) if all found, else False (0)	True (1) if all found, else False (0)

Output sorting:

The step returns sequences for the source/destination deck positions that have a successful barcode match. These sequences can be sorted for optimized parallel pipetting of sources or destinations, or just keep the given worklist order.



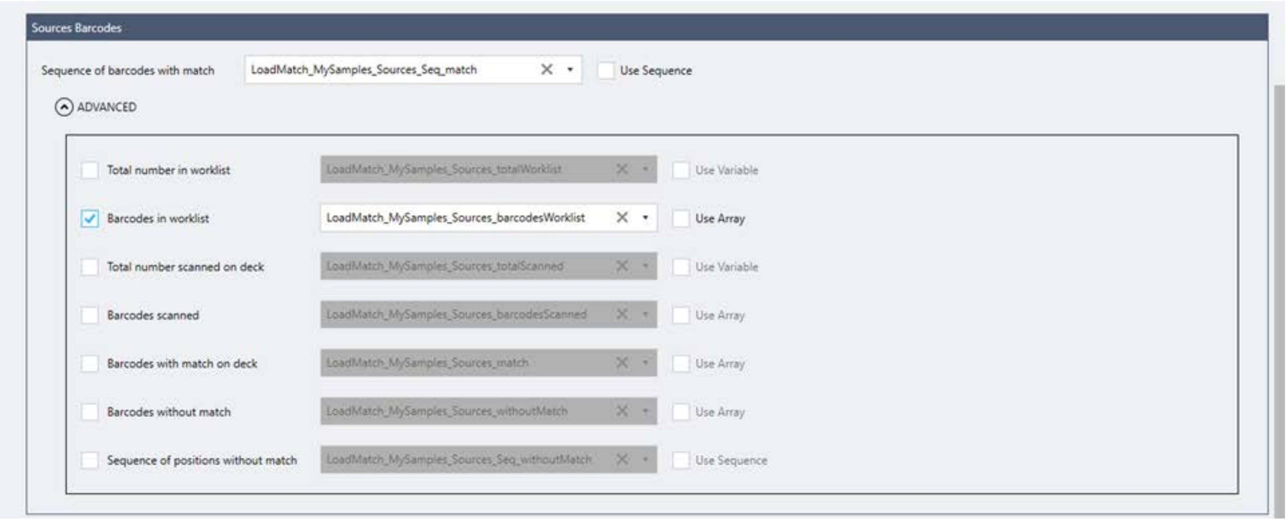
Source Barcodes:

If source barcodes are mapped in the Worklist screen, this section will be displayed to return a **sequence** with the positions of the selected source carriers that have a barcode match with the given worklist data.

<u>If source AND destination barcodes were mapped in the worklist</u>, the returned data in this section will only contain information of the sources of the given pairs of source and destination barcodes were both are found on the deck. If the destination barcode could not be found, the corresponding source barcode will be excluded.

Optionally, in the advanced section, additional outputs can be generated for advanced programmers:

- (integer) Total number of source barcodes in the worklist
- (array) All source barcode strings in the worklist
- (integer) Total number of source barcodes scanned on the deck
- (array) All source barcodes strings scanned on the deck
- (array) Source barcodes scanned on the deck with a match in the worklist
- (array) Source barcodes scanned on the deck without a match in the worklist
- (sequence) Sequence containing the source carriers/positions without a match in the worklist



Destination Barcodes:

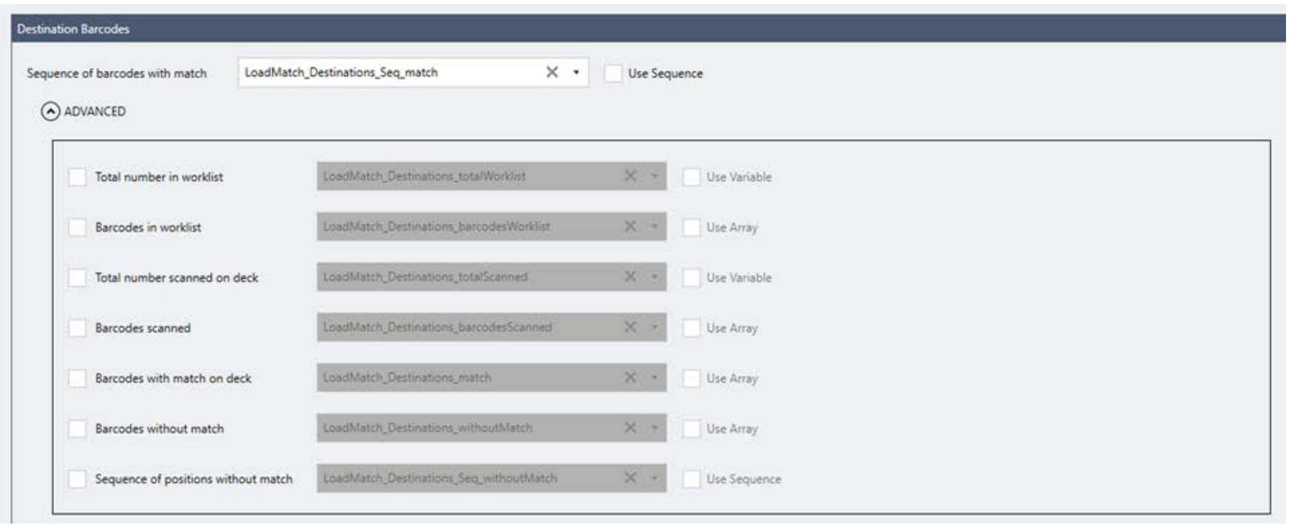
If destination barcodes are mapped in the Worklist screen, this section will be displayed to return a **sequence** with the positions of the selected destination carriers that have a barcode match with the given worklist data.

See the Table Source/Destination return values for determining sequence and barcode match outputs.

<u>If source AND destination barcodes were mapped in the worklist</u>, the returned data in this section will only contain information of the destinations of the given pairs of source and destination barcodes were both are found on the deck. If the source barcode could not be found, the corresponding destination barcode will be excluded.

Optionally, in the advanced section, the user can get outputs of:

- (integer) Total number of destination barcodes in the worklist
- (array) All destination barcode strings in the worklist
- (integer) Total number of destination barcodes scanned on the deck
- (array) All destination barcodes strings scanned on the deck
- (array) Destination barcodes scanned on the deck <u>with</u> a match in the worklist
- (array) Destination barcodes scanned on the deck <u>without</u> a match in the worklist
- (sequence) Sequence containing the destination carriers/positions <u>without</u> a match in the worklist



Volumes:

If volumes are mapped in the Worklist screen, this section will be displayed to return an **array** of the volumes corresponding to positions with a barcode match.

<u>If only source OR destination barcodes were mapped in the worklist screen</u> it will output an array of volumes of the barcodes present on the deck (=with match). Exclude records not present.

<u>If both source AND destination barcodes were mapped in the worklist screen</u> it will output an array of volumes of both barcodes present on the deck (=source AND destination pair must have a match). Exclude records where both are not present.

14 Loading Labware

There are two options to load labware on the deck: Autoload and manual load. In both cases, always have the needed Carriers ready. The following table shows the possible course of action:

Method contains "Load Carrier" Commands	Autoload Instrument	Manual Load Instrument
Yes	Run the method. Insert Carriers in the tracks of the Autoload tray where indicated by LEDs. Click "**Load**" in the dialog. Option to manipulate sequences. Carriers are loaded automatically.	Run the method. The instrument will prompt a message to load the Carriers manually onto the deck.
No	Load the carriers manually into their positions on the deck. Run the method.	Load the Carriers manually into their positions on the deck. Run the method.

NOTE

*If no "**Load Carrier**" Commands are specified in the method, no checking of Carrier presence is possible.*

14.1 Autoload

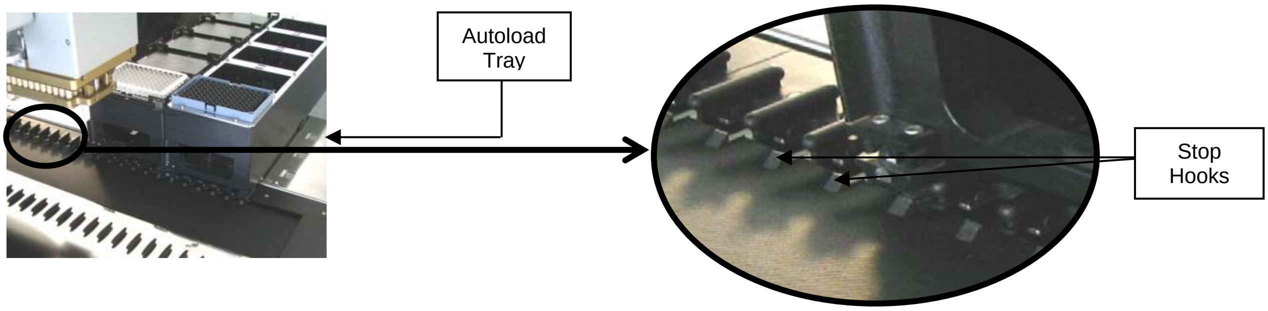
When using the Autoload option, make sure to load the Carriers with the appropriate labware first.

With "Load Carrier" Commands in the method

1. If the "**Load Carrier**" Commands have been incorporated into the method: start the "**Run Screen**" as described in the following sections and run the method.

   The correct positions for the insertion of Carriers will be highlighted by the green LEDs.

2. Insert the Carriers into the tracks of the Autoload tray until they touch the stop hooks on the far side of the tray.



These pictures are made from the rear of the instrument.

3. Click **[Load]** in the dialog.

   The Carriers are loaded onto the deck automatically by the "**Load Carrier**" Command in the method. At the same time, the barcodes of Carriers and labware are read and stored in a file.

NOTE

*If "**Sample Tracking**" is enabled (in the System Configuration Editor), the barcodes will be written to the Database.*

Another way for loading is to place the Carriers onto the defined positions of the Autoload tray before starting the method. Loading and barcode reading will then be performed without user input.

Without "Load Carrier" Commands in the method

1. If there have been no "**Load Carrier**" Commands incorporated into the method, the Carriers must be loaded manually into the positions on the deck defined in the Deck Layout before the run is started.
2. Make sure that the Carriers are inserted completely, until they touch the rear connectors.

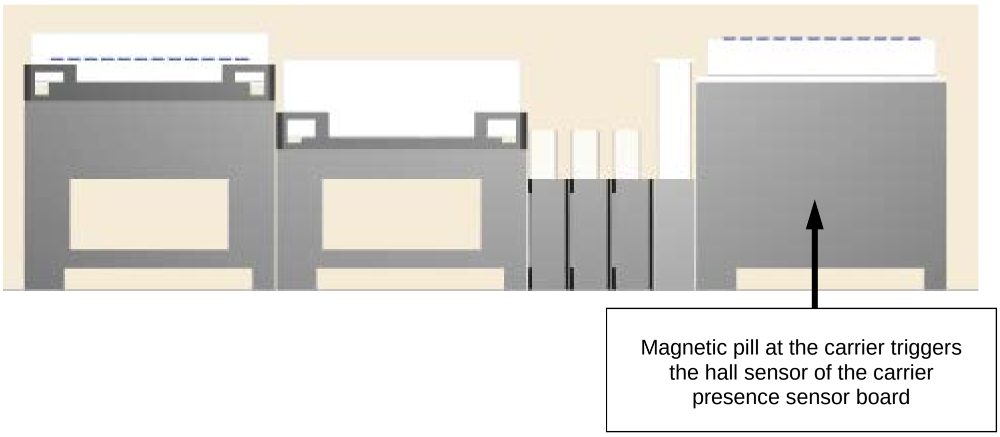
14.2 Manual Load

In order to load the Microlab instrument, fill the Carriers with the appropriate labware first.

With "Load Carrier" Commands in the Method

1. If the "**Load Carrier**" Commands have been incorporated into the method, start the run.

   The instrument will prompt a message to load the Carriers manually onto the deck.
2. Make sure the Carriers are inserted completely, until they lock in the rear snaps. A magnetic sensor checks whether the Carriers have been loaded correctly.



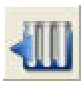
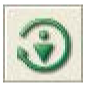
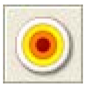
Without "Load Carrier" Commands in the Method

1. If there have been no "**Load Carrier**" Commands incorporated into the method, load the Carriers manually into the positions on the deck defined in the Deck Layout before the run is started.
2. Make sure that the Carriers are inserted completely, until they touch the rear snaps.

14.3 Command Description

The following tables give a brief overview of the available Microlab instrument-specific loading commands.

14.3.1 Single Steps

Preparation		
Command	**Icon**	**Action Performed**
Load Carrier		Load a Carrier on the deck.
Unload Carrier		Unload a carrier from the deck.
Reload carrier		Used to load, unload or reload a carrier.
Lock/Unlock Front Cover		Locks/unlocks the front cover.
Initialize		Initialize the instrument.
Move Auto Load		Moves the autoload to a selected track number.
Calibrate Carrier		Calibrate precise position of a high-density Microplate (1536 wells) before asp/disp.

14.3.2 Smart Steps

ML_STAR Smart Steps		
Command	**Icon**	**Action performed**
Advanced Load setting		Define other settings than the default of the load steps.
Load		Load Carrier on deck.
Unload		Remove Carrier from deck.
Load and match		Load samples and match loading information with worklist information, to generate data for use in the following process steps. An "**Import Worklist**" must be performed prior to this command.

14.4 Barcode Reading / Identification

Carriers, containers, racks and tip racks can be identified by a barcode. The reader device scanning the barcodes is mounted on the Autoload slide. The system must allow specification of ranges (barcode mask) for plausibility checking of barcode information.

14.4.1 Supported Barcode Types

The following barcode abbreviations can be read by the **1D Autoload Option**:

- ISBT standard
- Code 128 (subset B and C)
- Code 39
- Codabar
- Interleaved 2 of 5
- UPC-A/E
- JAN/EAN-8

The following codes abbreviations can be read by the **2D Autoload Option**:

1D Barcodes**:**

- ISBT standard
- Code 128 (subset B and C)
- Code 39
- Codabar
- Interleaved 2 of 5
- UPC-A/E
- JAN/EAN-8
- Code 93

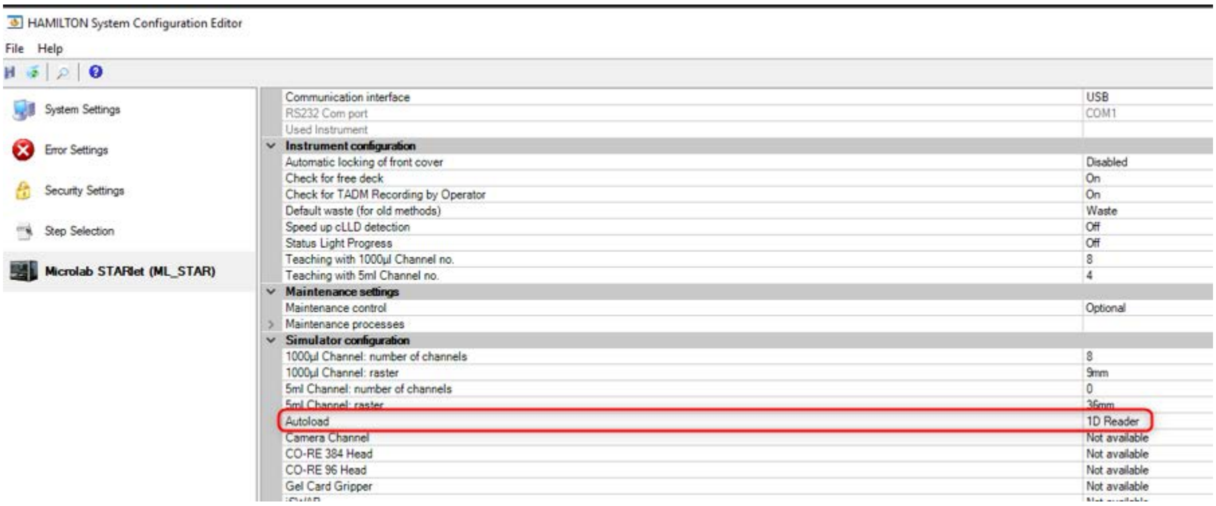
2D Codes**:**

- Data Matrix
- QR Code
- Maxi Code
- Aztec Code
- PDF417
- MicroPDF417
- GS1 DataBar
- GS1 Composite

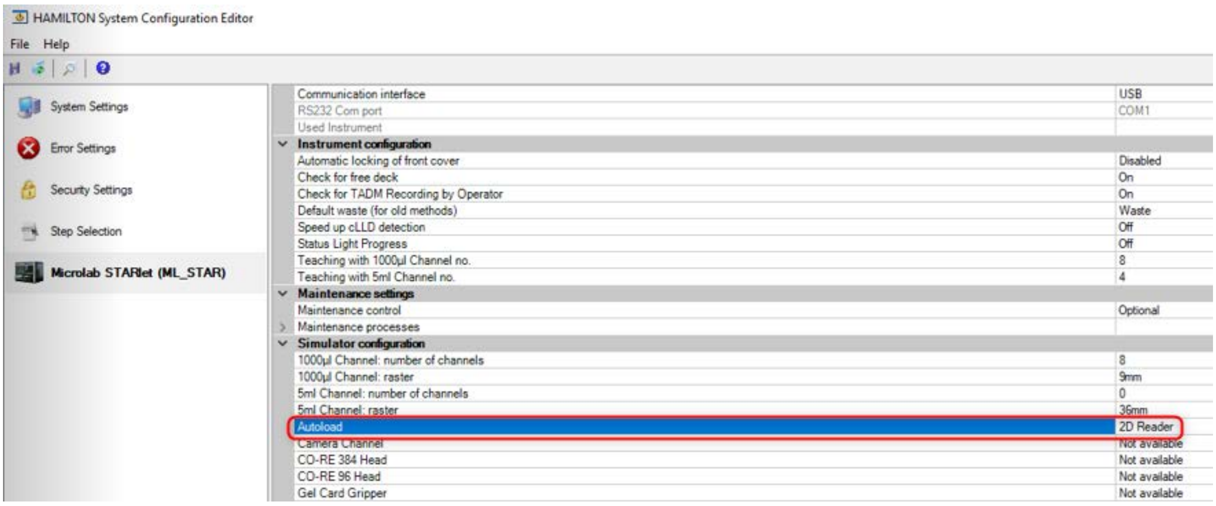
14.4.1.1 Activating the Reader

To activate the 2D Reader, switch from 1D Reader to 2D Reader in the System Configuration Editor.

1D Reader

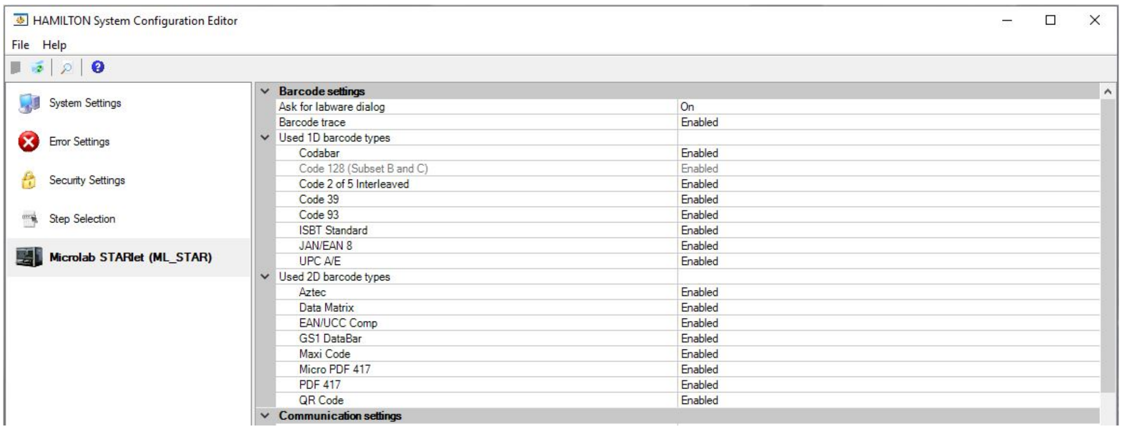


2D Reader



14.4.1.2 Enabling Codes

The barcodes to be read can be enabled through the "**Tools → System Configuration Editor → Microlab STAR**" Menu section.



NOTE

Enable only barcode types which will be used on the system. Disabling the unused barcode types will improve the reliability of the readings.

14.4.2 Unique Barcodes

The unique barcode check controls if a barcode has already been used on the system.

To use this function, four settings must be made:

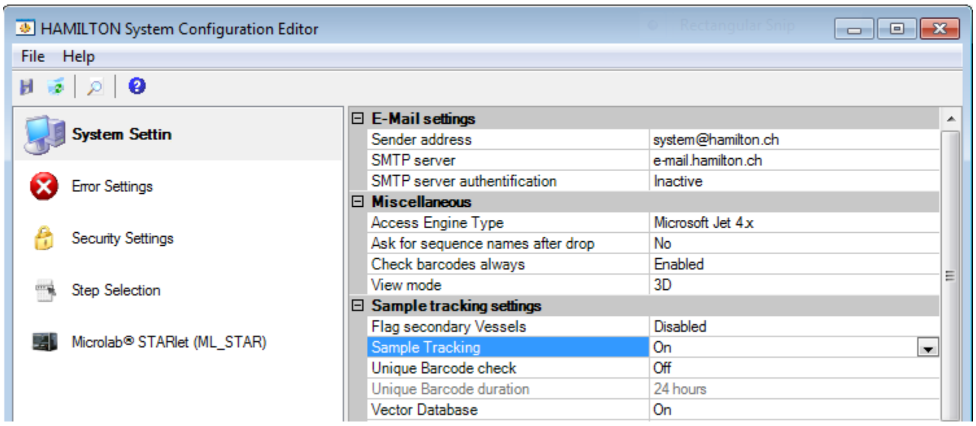
1. Activate sample tracking.
2. Specify the unique barcode check mode.
3. Specify the duration of the check period.
4. Set the unique barcode flag on the desired labware positions.

As soon as the unique barcode checking is set to “**Track only**” or “**Check and Track**”, all read barcodes will be written to the data base with a timestamp (when it was loaded).

Every load step will now refer to this database and check if the barcode has been loaded during the specified check duration.

Activate Sample Tracking

To use the Unique barcode check, Sample Tracking must be set to “**ON**” in the “**System Configuration Editor → System Settings**”:



Specify the “**Unique Barcode check**” Mode:

“OFF” Mode

Unique barcode management is switched off. Meaning, no check for unique barcodes, no writing to the database will be done.

“Track only” Mode

Unique barcode check will be turned on partially. Barcodes and timestamp are added to the database to be checked in the future, but no checks are performed. If the barcode has been loaded previously, the timestamp will be updated. Meaning there will be writing but not checking of barcodes.

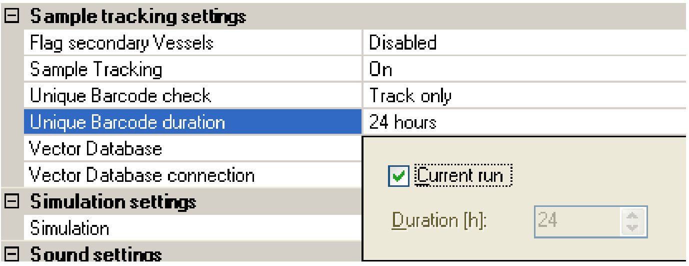
"Check and Track" Mode

Unique barcode check operation is fully turned on. If a barcode is presented a second time within the specified unique barcode duration, a warning is invoked that this barcode has been loaded previously. Meaning it will write at the same time check barcodes.

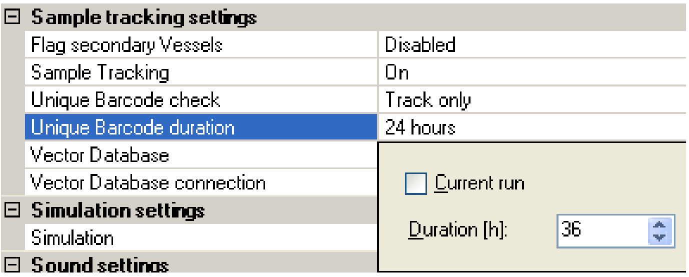
Specify the Unique Barcode Duration

Click the **['...']** Button if unique barcode duration is selected, to open a wizard which allows specification of the duration parameter.

- Current run: Barcodes must be unique within the current run.

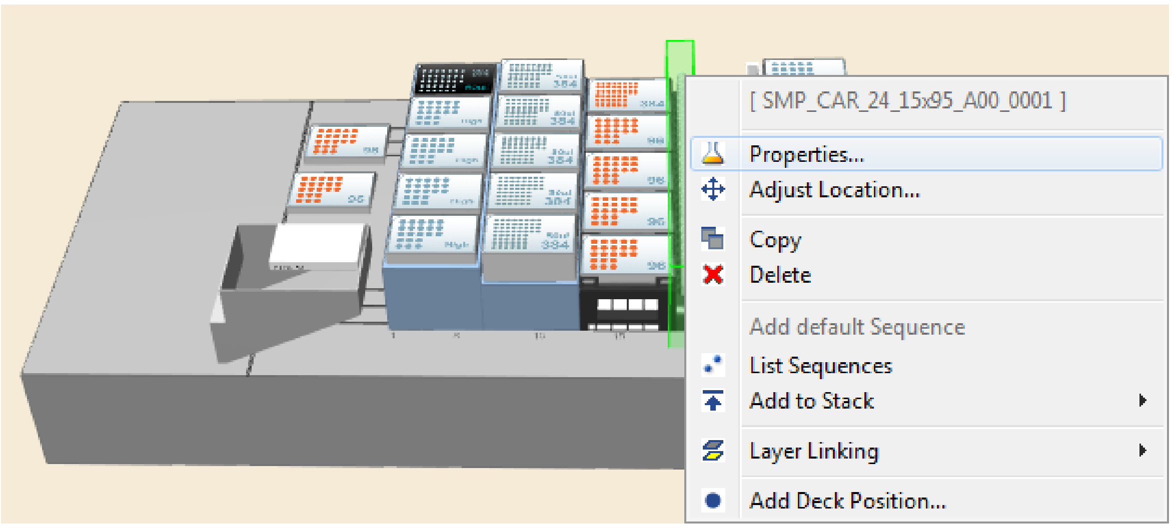


- Time in hours: Duration in hours within a barcode must be unique.

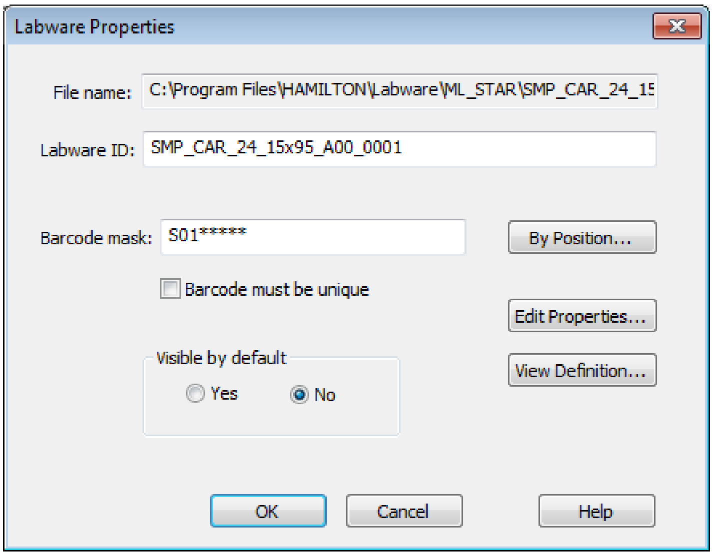


Set the Unique Barcode Check Flag on the Labware:

1. Open the Context Menu of the specified labware in the Deck Layout (right mouse-click).

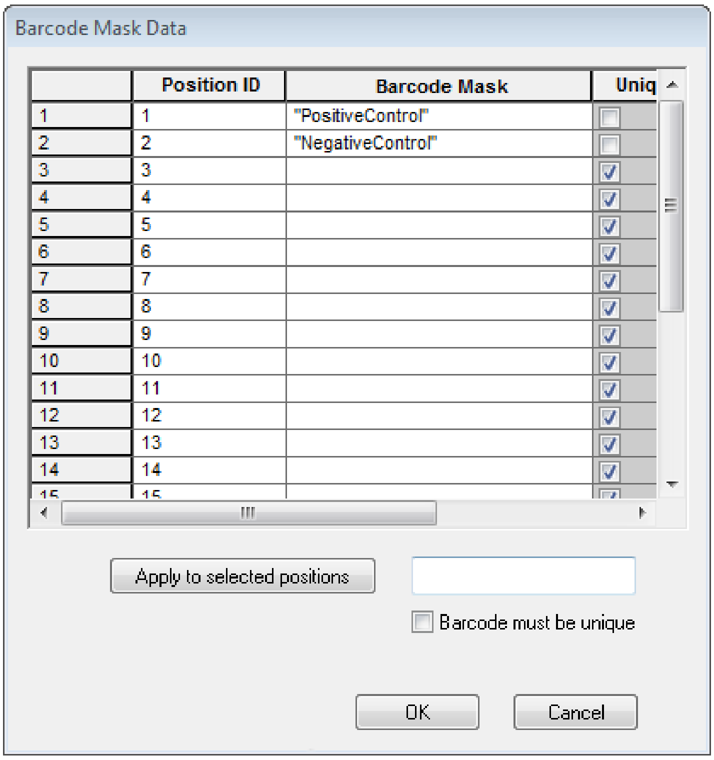


To make the sample Carrier barcode itself unique, tick the “**Barcode must be unique**” Checkbox underneath the barcode mask. Now, the carrier barcode will be controlled at every load step.



2. To set barcodes on the tubes as unique, select the **[By Position…]** Tab.

   In the example shown below, there are two barcodes that should not be checked for distinction. Every barcode in position 1 and 2 is allowed to appear multiple times, but the barcode mask for these barcodes must be correct.



3. All other positions are marked as “**Unique**” in the right most column. This will enable the checking for unique barcodes (as long as all other settings are appropriate).

4. To mark all barcodes for checking, check the "**Barcode must be unique**" Checkbox. Select all positions and click the **[Apply to selected positions]** Tab.

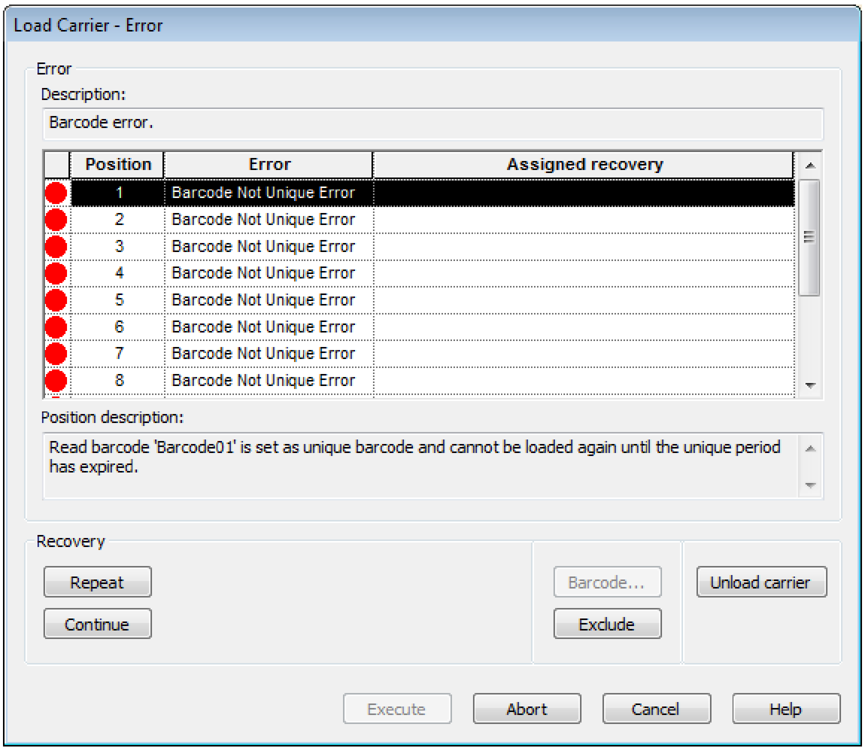
Behavior at Runtime

During runtime, two errors can be raised:

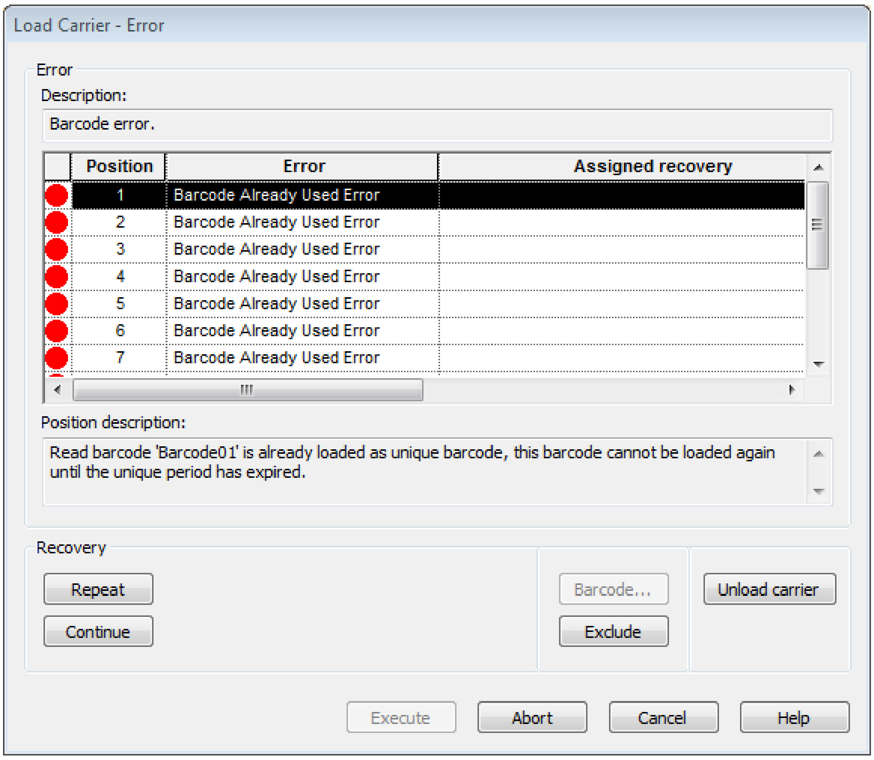
First load	Second Load	Error
Unique flag on Labware not set	Unique flag on Labware not set	No error
Unique flag on Labware not set	Unique flag on Labware set	Barcode Not Unique Error
Unique flag on Labware set	Unique flag on Labware set	Barcode Not Unique Error
Unique flag on Labware set	Unique flag on Labware not set	Barcode Already Used Error

This table is true for "**Check and Track**" when activated within same run or within specified check unique barcode period or when presenting a barcode multiple times.

Barcode Not Unique Error



Barcode Already Used Error

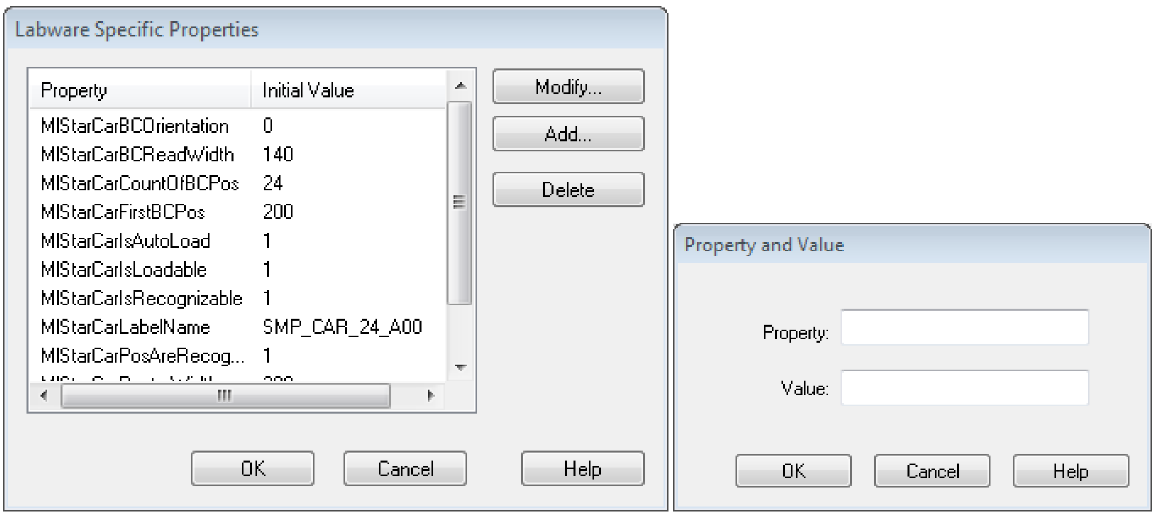


14.4.3 2D Autoload

14.4.3.1 How to Set the Different Properties

In most cases, it is possible to work with the standard labware integrated in VENUS. A new 2D specific property has just to be set when the given labware does not work with those values.

1. Open the labware you want to modify in the labware editor click the labware properties button.
2. Add or modify a property with the specific buttons.



3. Press OK twice and save the labware afterwards.

14.4.3.2 Description of Properties and Usage

See the picture at the end of this chapter for a visual help and read the description to find the function of each property.

Properties used on “Load carrier” only.

<table>
<tr><th>Key name</th><th>Range [unit]</th><th>Default</th><th>Description</th></tr>
<tr><td>MlStar2DBarcodeCarFirstBCPosShift</td><td>+/- value [1/10mm]</td><td>0</td><td>This property allows a correction of read position for first barcode as defined with property “MlStarCarFirstBCPos”.</td></tr>
<tr><td colspan="4">Please do not change the value of “MlStarCarFirstBCPos”!
Depending on carrier design, the light of the 2D barcode reader may produce a shadow on the barcode. Use this property to shift the position by a few millimeters.</td></tr>
</table>

These properties define where and how a barcode can be read using the 2D barcode reader.

All properties can be used for every kind of labware (carrier, rack and container).

<table>
<tr><th>Key name</th><th>Range [unit]</th><th>Default</th><th>Description</th></tr>
<tr><td>MlStar2DReaderRoiYCenterOffset</td><td>+/- 1000 [1/10mm]</td><td>0</td><td>Defines the distance from the barcode trigger position to the barcode center</td></tr>
<tr><td>MlStar2DReaderRoiZCenterOffset</td><td>0…2500 [1/10mm]</td><td>0</td><td>Defines the distance from the bottom of the labware to the barcode center.</td></tr>
<tr><td colspan="4"><u>Example:</u>
• If this property is used on a plate- or tube carrier, the value defines the distance from the instrument deck to the barcode center.
• If this property is used on a rack or a container, the value defines the distance from the rack / container bottom (lowest point) to the barcode center
The default for this property will be set as half height of container length, if the property “MlStarCarOpenRasterBarcodePositions” is defined.</td></tr>
<tr><td>MlStar2DReaderRoiYWidth</td><td>0…960 [1/10mm]</td><td>0</td><td>Width of ROI window.</td></tr>
<tr><td colspan="4">If property “MlStarCarOpenrasterBarcodePositions” is defined, the default for this property will be set to the width of labware.</td></tr>
<tr><td>MlStar2DReaderRoiZHeight</td><td>0…1280 [1/10mm]</td><td>0</td><td>Width of ROI window.</td></tr>
</table>

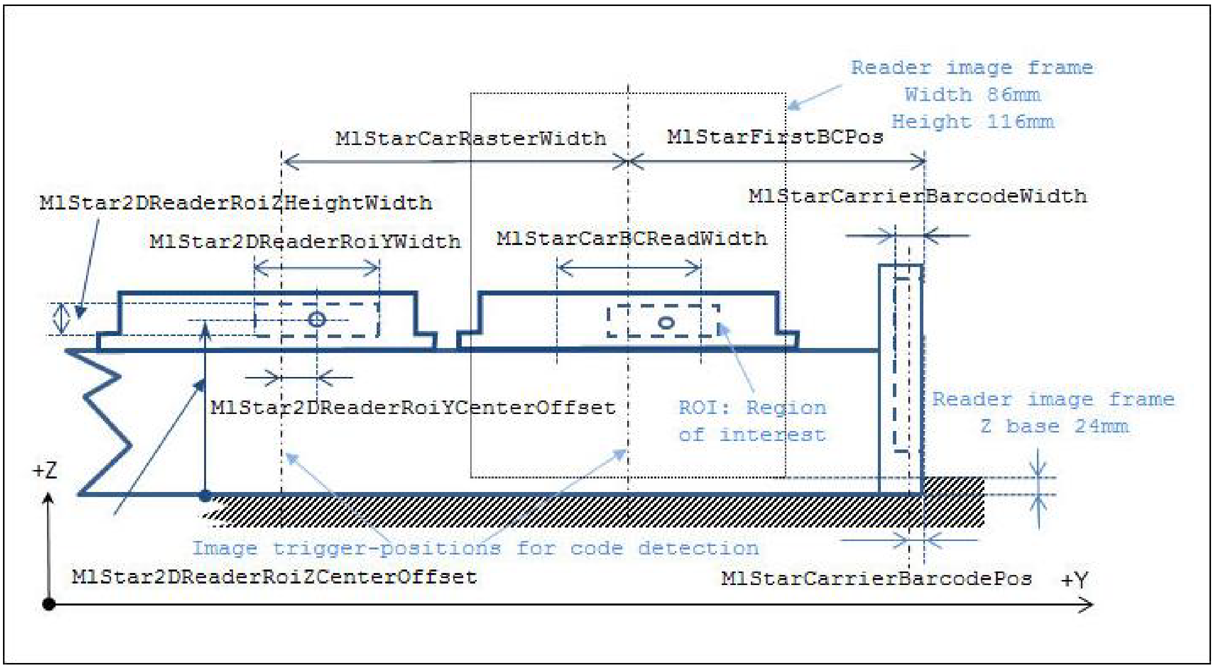
Key name	Range [unit]	Default	Description
If property "MlStarCarOpenrasterBarcodePositions" is defined, the default for this property will be set to height of container length.			
MlStarCarBCOrientation	0…2	0 or 2	Defines the orientation of the applied 1D barcode on a rack respectively on a container.
This property will be used if a carrier is loaded with irregular barcode read mode "MlStarCarOpenRasterBarcodePositions". 0 = Vertical, 1 = Horizontal, 2 = Open mode, both directions possible If this property is not defined, the following defaults will be used: Rack = 2, Container = 0 If this property is defined on a carrier, see "MlStarCarBCOrientation"			
MlStar2DReaderIlluminationSettings	"0;0;0;0;0;0;0"	First 5: 0…2 Sixth: 0…15 Seventh: 50…500 [us]	Defines the 2D barcode reader illumination setting. Define 7 values separated by a semicolon. If this property is not defined, the default will be used
First value:	Reader light upper rear	0 1 2	Firmware standard ON OFF
Second value:	Reader light lower rear	0 1 2	Firmware standard ON OFF
Third value:	Reader light upper front	0 1 2	Firmware standard ON OFF
Fourth value:	Reader light lower front	0 1 2	Firmware standard ON OFF
Fifth value:	Reader external light	0 1 2	Firmware standard ON OFF
Sixth value:	Contrast gain	0 1…15	Firmware standard Low… high [image noise!]

Key name	Range [unit]	Default	Description
Seventh value:	Camera exposure time	0 50… 500	Firmware standard Exposure time [us]
Firmware standard:		“0;0;0;0;1;0;0”	
Exception on horizontal reading:		“1;1;0;0;1;0;0”	

NOTE

In which case which property must be set, depends on the lightning at the locality and the used customer barcodes.

.



14.4.3.3 How to React on Read Failures

- Check if the used barcode type is correctly activated. (See 14.4.1 Supported Barcode Types for more information)
- Check light conditions:
  - o Make sure there are no bright directional light influences such as sunshine or lamps
  - o Avoid shadows.
- Check the region of interest. (See chapter 14.4.3.2 for more information)
- Deactivate not used barcode types to strengthen the read result and speed. (See chapter 14.4.1.2 Enabling Codes)

15 iSWAP / IPG

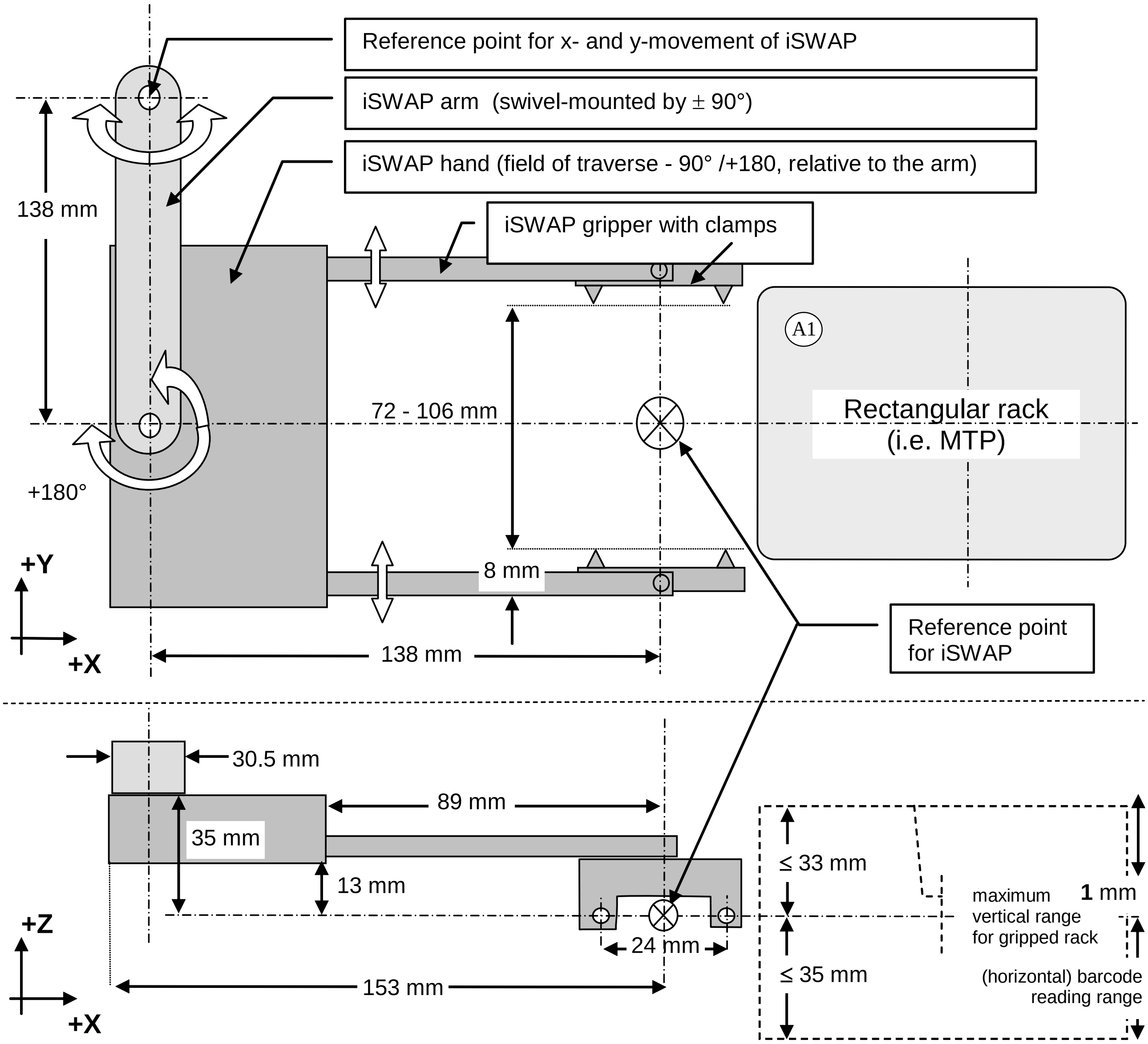
15.1 Plate Handling with iSWAP

iSWAP (internal Swivel Arm Plate Handler) is a robotic arm that transports microplates, covers of micro plates, archive plates or filter plates used for the vacuum box to and from positions on the deck of Microlab STAR.

Like the Microlab STAR pipetting channels, the iSWAP has a "**Traverse height**" of 145 mm above the deck (245 mm above the origin).

IPG (Internal Plate Gripper) is a robotic arm equivalent to the iSWAP for the Microlab STAR^V and VANTAGE instruments. See the Microlab STAR^V and VANTAGE Operator´s Manuals to find more information about the IPG.

15.1.1 iSWAP Geometry



Geometrical definitions of iSWAP

NOTE

Height for gripping is calculated from the upper rim of the plate.

Recommended gripping height for MTP: 3 mm

Recommended gripping height for DWP: 13 mm

(an error will appear if there is an attempt to grip a Deep Well Plate with a grip height < 10 mm traverse height).

15.1.2 Special iSWAP Features

Among the special features of iSWAP are:

- A torque sensor which signals to the robot how tightly it is gripping a labware object.
- The plates can be placed in landscape or portrait orientation.
- Load and unload plates to and from a plate stacker on the left side of the instrument outside the working area (with some restrictions, this can also be done on the right side).
- Handling plates 100mm below the deck (if iSWAP Rev. ≥ 03. e.g. possible on the Microlab STARplus without deck extension).

Not all features are available on the old iSwap Type.

NOTE

Gripping of micro plates in landscape orientation is only possible with the Landscape iSWAP P/N 190220.

*Use the "**Hamilton System Configuration Editor**" to enable the functions of the "**Landscape**" iSWAP: Microlab STAR / Simulator configuration / iSWAP / "**Large Gripper**".*

15.1.3 iSWAP / IPG Positions

The iSWAP is mounted on the pipetting arm of the Microlab STAR. The IPG is mounted on the pipetting arm of the Microlab STAR^{V} and VANTAGE.

During pipetting, the iSWAP / IPG is in park position and does not affect the movement of the pipetting channels.

For plate transport steps, the pipetting tools will be moved away so that the greatest possible transport area will be available to the iSWAP / IPG. The orientation of the arm and gripper will be automatically calculated by the software according to the Deck Layout and programmed grip direction.

Using multi-arm systems can affect the movement range.

15.1.4 Grip and Opening Widths

"**Grip width**" and "**opening width before access**" can vary between 72 and 108 (132) mm.

Archive and filter plates in particular have to be gripped no more than 28 mm below the upper rim. The lower rim of the plate should be no more than 35 mm below the gripping position, to prevent collision with other labware placed on the deck.

15.1.5 Grip Force

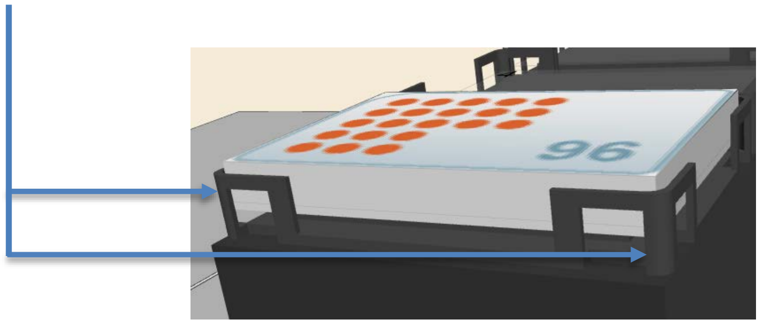
The default grip force is appropriate for common MTPs. It should be set higher for archive plates, and lower for soft plates.

ATTENTION

In order to prevent spillage during transport with iSWAP / IPG, it is recommended not to overfill the wells (e.g. fill maximum 80% of nominal well volume).

The maximum weight to be transported is 300 g. This should not be exceeded.

The rectangular racks should only be lifted from or placed onto carriers with slide guides (see the corresponding Microlab instrument Operator's Manual).



The gripping area should be flat and stable. Ensure that the Gripper's clamps will not pick up strips instead, the lower part of the plate's frame.

Define the correct plate orientation to ensure that plates lifted from MTP stacks via iSWAP can be identified on deck.

Affix plate barcodes correctly (see Technical Specifications in the corresponding Microlab instrument Operator's Manual).

15.1.6 Pick-up by iSWAP / IPG

Rectangular racks are always picked up at the short side. The Gripper has a preferred gripping direction, which can be modified within the advanced settings of the "**GetPlate**" and "**iSwapTransport**" Steps.

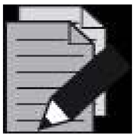
15.1.7 iSWAP / IPG Movement

In the case of a complex movement used to transport plates to or from stacks, shelves, etc., a plate once picked up will be moved up in small steps to the lift-up height enabling collision-free insertion or extraction.

Then it will be moved in horizontal direction by the retract distance out of a slot. Afterwards (as is the case for a single movement) the iSWAP / IPG will be moved in maximum z-position (the greatest possible distance from the labware placed on the deck). The plate will be turned in such a way that the arm of the iSWAP / IPG is bent into a compact position for the x/y-transport. The software controls these movements and prevents any collision with the pipetting channels or side panels.

15.1.8 Plate Release

For plate release, the orientation of the arm and the gripper will be calculated automatically so that the orientation of position A1 of the plate is in accordance with the Deck Layout. If this is not possible, the plate must be gripped inversely (i.e. from the opposite side). See the VENUS Help Function.



NOTE

During plate placement, the plate will be gently pressed onto the rack. Ensure that strips are fixed securely in the frame.

15.1.9 Accessible Areas

The iSWAP can be used over the whole deck area and the adjoining zones. The table below lists the coordinates of the accessible area relative to the zero point of the Microlab instrument:

X-Movement Pipetting Arm	**Minimum Absolute Position**	**Maximum Absolute Position**		
		STARlet	**STAR**	**STARPlus**
173050/51	- 206 mm	+ 1038 mm	+ 1578 mm	+ 2208 mm
173052	- 118 mm	+ 1250 mm	+ 1790 mm	+ 2420 mm
173054	Depending on Dual Arm Configuration [1)]	+ 1215 mm	+ 1755 mm	+ 2385 mm

1) Left: Modular Arm 173050/51;
Right: iSWAP Arm (173054), min. absolute Pos: +121 mm
Left: 96 Arm 173053;
Right: iSWAP Arm (173054), min. absolute pos: -87 mm

X-Movement Pipetting Arm ***Channels***	**Minimum absolute Position**	**Maximum Absolute Position**	**Remarks**
173050/51			On all platforms (STARlet STAR, STARplus)
4 Channels	- 221 mm	+ 605 mm	
8 Channels	- 185 mm	+ 605 mm	
12 Channels	- 149 mm	+ 605 mm	
16 Channels	- 113 mm	+ 605 mm	
173052	- 257 mm	+ 605 mm	
173054	- 257 mm	+ 605 mm	

NOTE

*The minimal and maximal absolute positions of the x- and y-movement cannot be reached by the default settings of transport commands. Use the settings of “**Complex movement**” in the transport commands.*

The standard configuration of an Microlab instrument Series instrument is WITH acrylic covers. The maximal and minimal absolute positions of the x-movement can only be reached if the acrylic glass shields (left, right) are removed.

This is not only a mechanical issue. The instrument configuration has to be changed as well. Please contact a Hamilton Representative.

Movement	Minimum absolute Position	Maximum Absolute Position	Remarks
Z-Movement	+ 100 mm 0 mm	+ 282 mm + 282 mm	iSWAP (P/N 182600) Rev. 00 – 02 iSWAP (P/N 182600) Rev. ≥ 03 and Landscape iSWAP (P/N 190220) Position of clamp tips, if there is no inhibition by the labware
Gripper Movement	72 mm 72 mm	108 mm 132 mm	iSWAP (P/N 182600) Landscape iSWAP (P/N 190220) Default: 82 mm
Arm Rotation	- 90°	+ 90°	See Section 15.1.1 iSWAP Geometry
Hand Rotation	- 90°	+ 90°	See Section 15.1.1 iSWAP Geometry

Coordinates of Accessible Area

Certain plate orientations and gripping directions prevent the iSWAP from reaching the zones on both sides. For these cases, an easy method is recommended for plate transport.

ATTENTION

If a plate has to be gripped down low on the deck, enough space must also be left on the deck for the arm and Gripper of iSWAP.

See the STARV and VANTAGE Operator´s Manuals to find the accessible areas of the IPG.

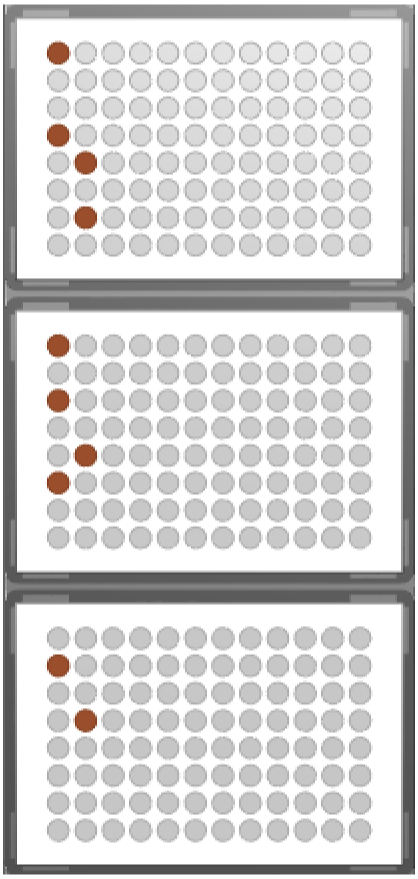
15.1.10 Sequence Definitions for Transport Steps

The iSWAP and IPG work on the basis of sequences (see Section 15.1.7 iSWAP Movement). Plates are moved from one position to another, each defined by its own sequence. The labware definition remains fixed on the deck, but the plates change sequences. To make this possible, definition of target and source plate positions must be of the same labware type.

After processing of a sequence, the current position will be set to the next labware ID. If the same plate should be transported several times, select sequence counting '(0) Manually' or set the current position back to the previous position.

NOTE

Sequence positions for transport steps are always differentiated by the labware ID. The current position of the transport sequence is to set the next labware ID in the sequence.



	Labware	Positio	X	Y	Z
1	Source_1	A1	118.00	338.000	187.150
2	Source_1	D1	118.00	311.000	187.150
3	Source_1	E2	127.00	302.000	187.150
4	Source_1	G2	127.00	284.000	187.150
5	Source_2	A1	118.00	242.000	187.150
6	Source_2	C1	118.00	224.000	187.150
7	Source_2	E2	127.00	206.000	187.150
8	Source_2	F1	118.00	197.000	187.150
9	Source_3	B1	118.00	137.000	187.150
10	Source_3	D2	127.00	119.000	187.150

In the example above, after the transport of plate Source_1, the current position will be set to plate Source_2 (line 5); and after transport of plate Soure_2 to plate Source_3 (line 9).

It is recommended to define separate sequences for transport and pipetting, because the behavior of transport and pipetting is different.

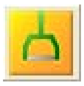
NOTE

*The orientation of the plate depends on plate positioning on the deck and not on sequences. The orientation can be changed under the "**Labware**" Tab by the function "**Adjust Labware Position**" – "**Rotation**").*

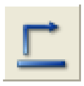
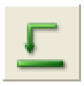
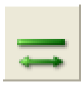
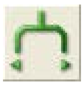
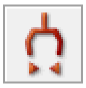
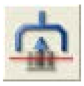
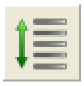
15.1.11 Command Description

The following tables provide a brief overview of the available Microlab instrument-specific iSWAP commands. The same commands control the IPG on STAR^{V} and VANTAGE.

15.1.11.1 Easy Steps

Transport: Easy Steps		
Command	**Icon**	**Action Performed**
iSWAP Transport		Transport a plate with iSWAP.

15.1.11.2 Single Steps

Transport: Single Steps		
Command	**Icon**	**Action Performed**
iSWAP Get Plate		Picks up a plate from the defined position.
iSWAP Place Plate		Sets a plate down in a defined position.
iSWAP Move Plate		Transfers a plate to another sequence.
iSWAP Open Gripper		Spreads the fingers of iSWAP's robotic hand.
iSWAP Close Gripper		Closes the fingers of iSWAP's robotic hand.
iSWAP Read Plate Barcode		iSWAP transports the labware item it is holding to the Barcode Reader so its barcode can be read.
iSWAP Get First Plate Position		Sequence over several plates is checked for the first plate position.
iSWAP Park		Parks the iSWAP under the pipetting arm.

15.2 Programming the iSWAP

The following example demonstrates the use of iSWAP with Microlab instrument. A plate is transported from a stacker to a processing position. Here, its plate barcodes are also read. After pipetting some reagent, the plate is brought to a reader and read (simulated). Then, the plate is transported to an output stack.

Creating the Deck Layout

Use the "**Search Labware**" Field to add the following carriers to the deck:

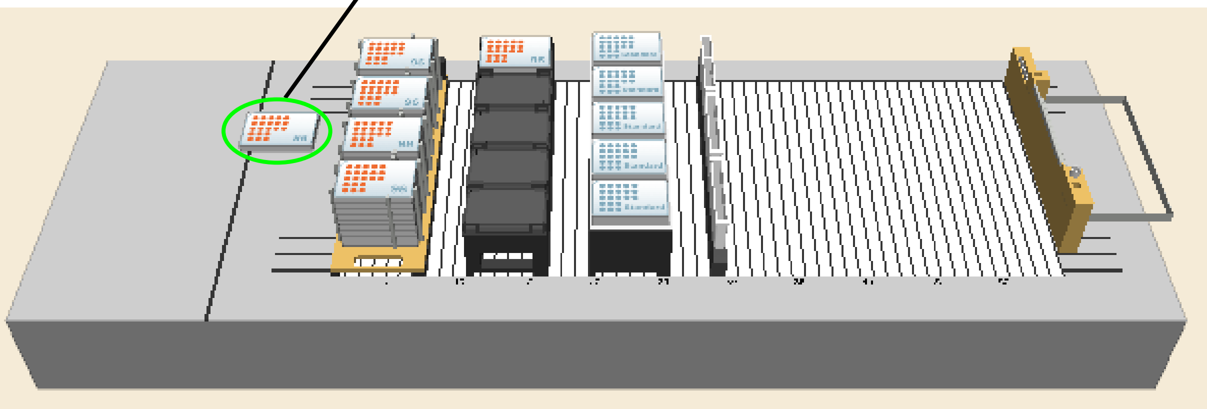
"**PLT_CAR_L4ST_LOW_A00_4x9_Nunc96**"

"**PLT_CAR_L5MD_A00**" with 1, "**Nun_96_Fl_Lb**" on position 1. Name it "**Processing**".

"**Nun_96_Fl_Lb**". Use the coordinates x = -70 / y = 350 / z = 160) and name it "**Reader**".

"**TIP_CAR_480_ST_A00**"

"**RGT_CAR_3R_A01**"



iSWAP Deck Layout

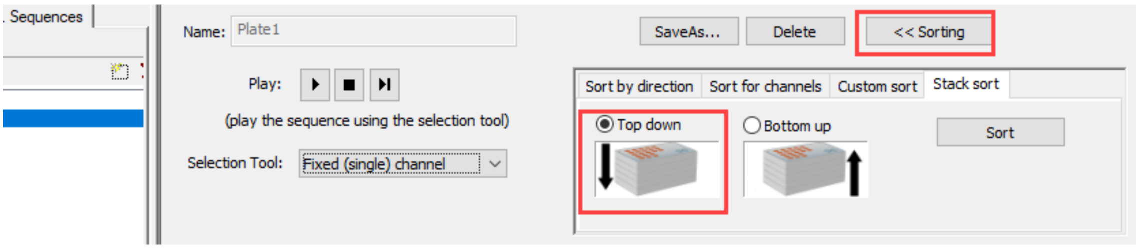
NOTE

To save deck space, the Stacker Carrier can be positioned 4 tracks to the left of the deck (track positions –3 to +3), and the plates will still be fully accessible by the iSWAP.

Creating the Sequences

1. Click the "**Sequences**" Tab and create the following sequences:

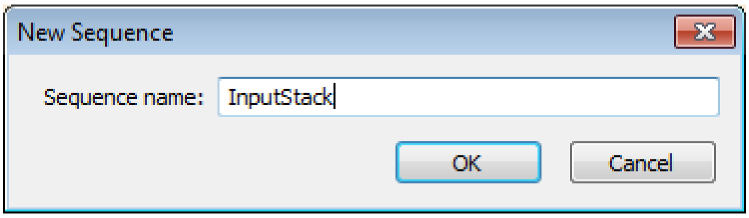
2. On the foremost plate stack, select only the well A1 of all 9 plates. Make sure the stamp tool is set to "**Fixed (single) channel**" and the sorting option is "**Top Down**" as shown below.



3. This means that the stack from top to bottom is used as a SOURCE stack.

	Labware	Positi	X	Y	Z
1	Input_0005	A1	129.25	510.50	172.40
2	Input_0004	A1	129.25	510.50	159.60
3	Input_0003	A1	129.25	510.50	146.80
4	Input_0002	A1	129.25	510.50	134.00
5	Input_0001	A1	129.25	510.50	121.20

4. In the "**Show Positions**" Window, check if the plate on top (the one with the highest z-coordinate is on sequence position one. Save the sequence as 'InputStack'.

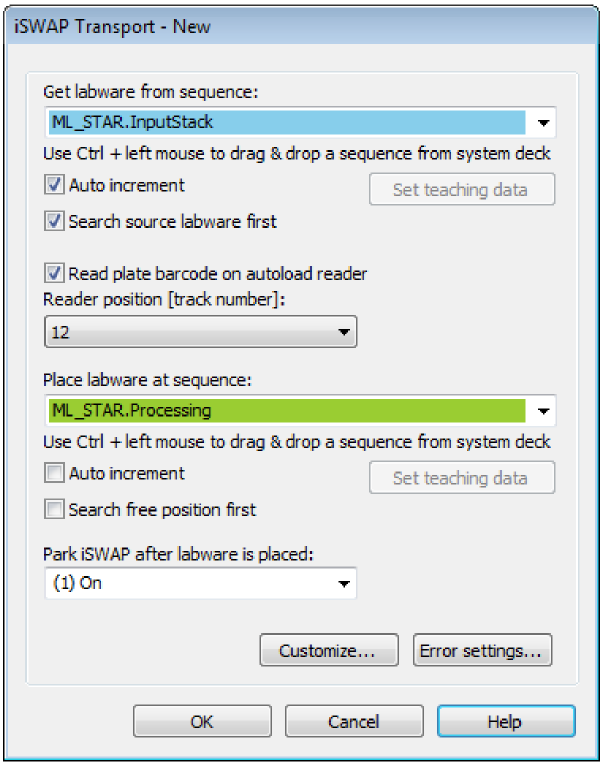


5. Do the same for the output stack on position 4. But this time, sort the sequence to start with the deepest position (sorting option is "**Descending**").
6. Save this sequence as "**OutputStack**"
7. Rename the sequence "**rgt_cont_120ml_a00_0001**" to "**Reagent**".

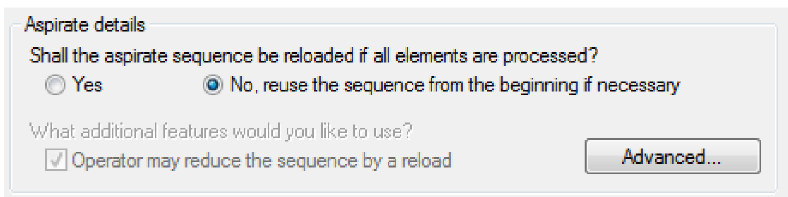
Creating the Method

To create the required method:

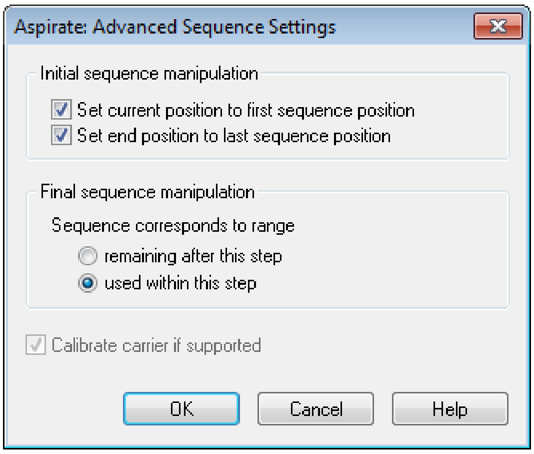
1. Drag and drop an "**iSwap Transport**" Step into the method. Since this is an Easy Step, an initialize step is not needed to initialize the instrument.



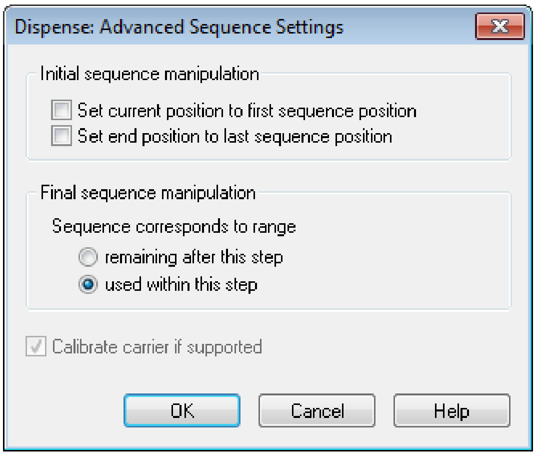
2. Set the 'InputStack' sequence as the get sequence. Activate the 'Auto increment' to make sure the current position points to the next plate after getting this one.
3. Setting the 'Search source labware first' will force the iSWAP to check where the first free position is. This is especially helpful if the loading state of a stack is not known. The iSWAP will check this and set the current position according to the first free position.
4. Activate the 'Read plate barcode' box to identify the plate. The Autoload will move to track 12 to read the plate's barcode. If the default settings from this step are being used, then there is no risk of collision. That means that the barcode reading with iSWAP is possible even on a fully loaded deck.
5. Set the "**Parking the iSWAP after labware is placed**" to On.
6. Use a "**Aliquote**" Smart Step to pipette 50µl of reagent over the full plate. Even if the sequence of the Reagent trough is used up, it will still restart at the beginning, since the Radio Button is set to **[No, reuse the sequence from the beginning if necessary]**:



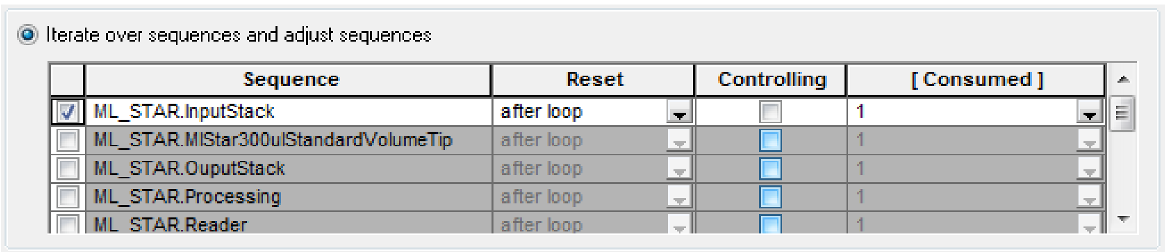
7. Also make sure the “**Initial sequence manipulation**” are set as follows:



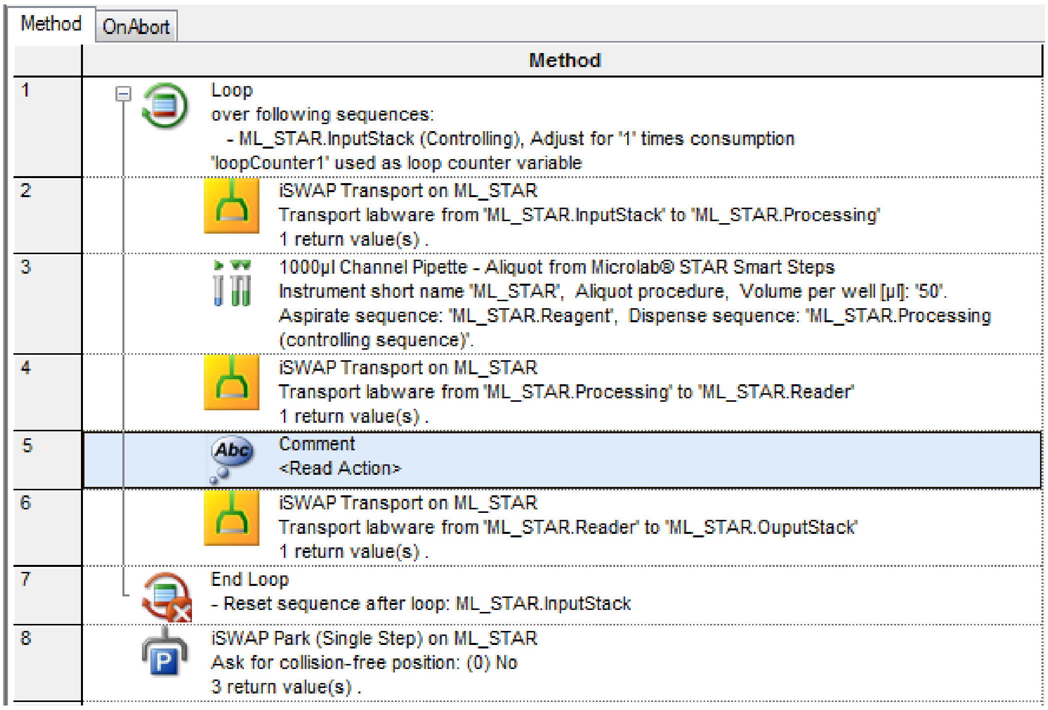
8. Because the sequence ‘Processing’ is used both for pipetting and transport, the “**Final sequence manipulation**” should be used as follows:



9. Now, use an “**iSwapTransport**” Step to bring the plate from the “**Processing**” Position to the “**Reader**” Position. Do not park the iSWAP.
10. Since there is no reader connected, a comment step is used to simulate this.
11. After reading, the plate has to be taken at the reader position and brought to the output stack. Make sure the “**Auto increment**” is only set at the “**Output Stack**” Place Sequence.
12. To have all plates processed, create a loop, and move all steps inside. The loop limitation should be the sequence ‘InputStack’.



13. To make the method perfect, add an iSWAP park step at the end (the transport step in line 6 was set to park iSWAP = Off).
14. When all of the steps have been done, the method should look like this:



15.3 Helpful Hints

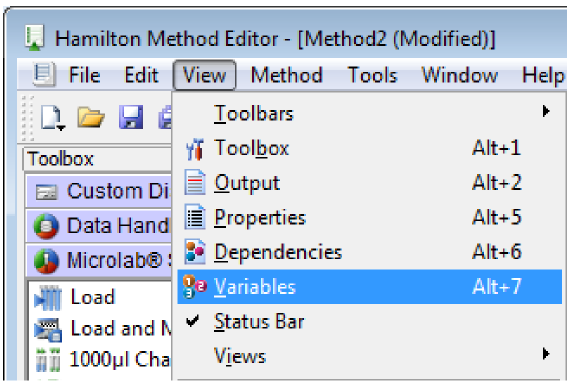
15.3.1 Define Special Labware Data / Parameters Only Once

NOTE

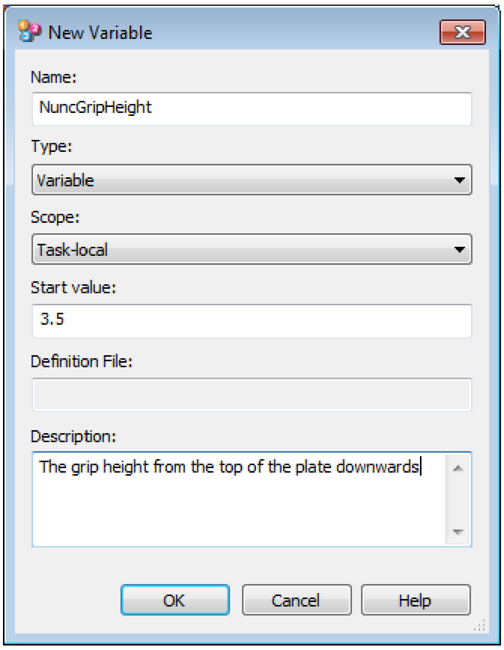
*It is possible to unmark the "**Generate default deck sequence**" in the "**Labware**" Tab. In this case, no default sequences are created.*

This option is helpful if many special sequences have to be created.

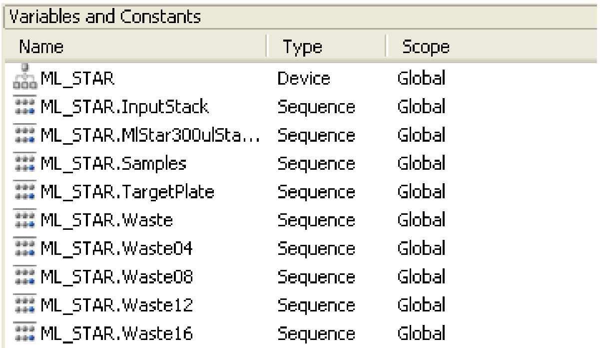
If the default values for GripForce, GripWidth, GripHeigth, OpeningWidthBeforeAccess (because an own new plate labware has been created or the method must use different values) are not used, it would seem sensible to save these parameters in a variable. This can be done by selecting "**View**" → "**Variables**" in the Method Editor.



1. To create a new variable, enter the Context Menu with a right mouse click and select “**New**”. The dialog box below will then appear and can be filled out.



2. It is now possible to create many variables as needed.



Variables and Constants

Name	Type	Scope
ML_STAR	Device	Global
ML_STAR.InputStack	Sequence	Global
ML_STAR.MlStar300ulSta...	Sequence	Global
ML_STAR.Samples	Sequence	Global
ML_STAR.TargetPlate	Sequence	Global
ML_STAR.Waste	Sequence	Global
ML_STAR.Waste04	Sequence	Global
ML_STAR.Waste08	Sequence	Global
ML_STAR.Waste12	Sequence	Global
ML_STAR.Waste16	Sequence	Global

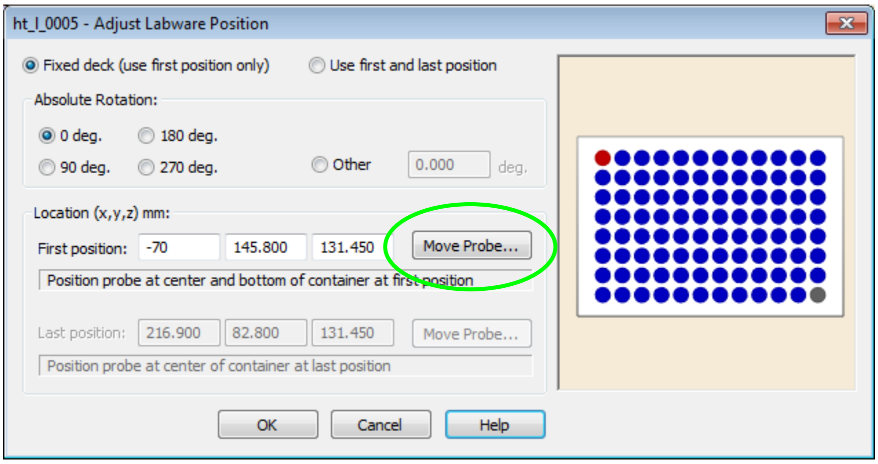
3. These variables can now be used in all iSWAP or CO-RE Gripper commands for the special plate.

4. If there are multiple grip commands for this plate in the method, it is much faster to change these values once in the variable definition. Of course, every plate type needs its own parameters.

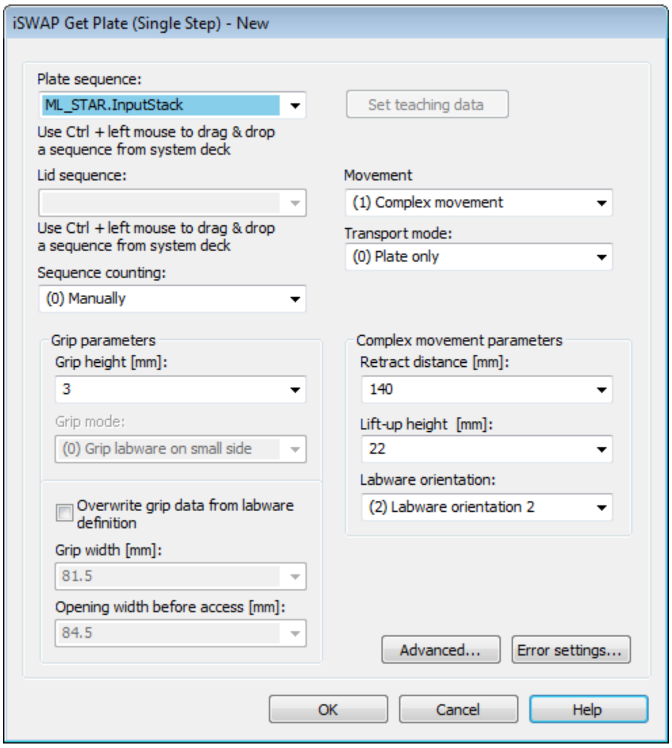
15.3.2 Move to Positions Outside of the Slot Area

The iSWAP is able to reach positions outside the slot area or below the deck's surface (check the specifications for the exact range).

Use the "**AdjustPosition**" Function (double-click on the labware) to move a plate outside the slot area. Find the position using the **[Move Probe]**.



If a collision is possible (e.g. with the housing of a reader), select "**Complex Movement**" as the "**Movement Type**" and define a "**Retract Distance**" greater than the plate length. Refer to the image below.



All bending and turning moves are now executed in a way which is directed away from the housing. The iSWAP will drive to the position using x-drive.

16 CO-RE Gripper / Quad CO-RE Gripper

This section provides information about labware handling using the CO-RE gripper on the Microlab STAR, and the Quad CO-RE Gripper (QCG) on Microlab STAR^{V} and VANTAGE.

The CO-RE Gripper transports different labware within the instrument deck using the pipetting channels:

- Microplates
- Covers of micro plates
- Archive plates
- Filter plates
- Manifold top of the BVS / CVS system

Plates can be gripped in landscape or portrait format within the working area. Rotation of plates is not an option and no process control - checking whether plates are gripped or not - is integrated in the software. An additional “**Plate Turning Station**” can be used to rotate a plate.

The **CO-RE Gripper** consists of two gripping paddles that are held by 2 neighboring 1mL pipetting channels of the Microlab STAR.

The **Quad CO-RE Gripper** (QCG) consists of two gripping “fingers” that are held by 4 neighboring 1mL pipetting channels of the Microlab STAR V or VANTAGE.

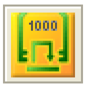
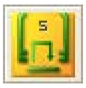
The pipetting channels can be freely selected. The “**Traverse height**” of the pipetting channels with the gripping jaws is the same as with the tips: 145 mm above the deck.

Given that there is no torque sensor at work here, ensure that the CO-RE Gripper does not grip plates too tightly. The distance between the jaws that is specified for gripping should be smaller than the length or width of the plate.

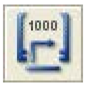
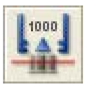
16.1 Command Description

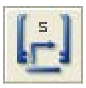
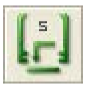
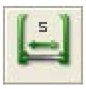
The following tables give a brief overview of the available Microlab STAR-specific CO-RE Gripper commands. The same commands are used for the Quad CO-RE Gripper on Microlab STAR^{V} and VANTAGE.

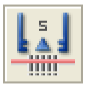
16.1.1 Easy Steps

ML_STAR		
Command	**Icon**	**Action Performed**
1000 µL Channel CO-RE Grip Transport		Transports a plate from start sequence to end sequence using the 1000µl-pipetting hannels and the 1000µl CO-RE Gripper tool.
5 mL Channel CO-RE Grip Transport		Transports a plate from start sequence to end sequence using the 5 mL-pipetting channels and the 5 mL CO-RE Gripper tool.

16.1.2 Single Steps

Transport		
CO-RE Grips with 1000µl-Pipetting Channels		
Command	**Icon**	**Action Performed**
1000 µL Channel CO-RE Grip Get Plate		Picks up a plate from the defined position.
1000 µL Channel CO-RE Grip Place Plate		Sets a plate down in a defined position.
1000 µL Channel CO-RE Grip Move Plate		Transfers a plate to another sequence.
1000 µL Channel CO-RE Grip Read Barcode		Barcode of the plate held is read (after unloading the carrier that the CO-RE Gripper will be moved to).

Transport		
CO-RE Grips with 5 mL Pipetting Channels:		
Command	**Icon**	**Action Performed**
5 mL Channel CO-RE Grip Get Plate		Picks up a plate from the defined position.
5 mL Channel CO-RE Grip Place Plate		Sets a plate down in a defined position.
5 mL Channel CO-RE Grip Move Plate		Transfers a plate to another sequence.

5 mL Channel CO-RE Grip Read Barcode		Barcode of the plate held is read (after unloading the carrier that the CO-RE Gripper will be moved to).

16.2 Programming the CO-RE Gripper / Use of Arrays

The following example demonstrates the use of the CO-RE gripper robotic plate handler with Microlab instrument. A set of plates are taken from the input carrier and brought to the barcode reading position of a third-party barcode reader. Then, the plate is placed on another carrier for further actions (that are not part of this example).

Reading of the plate's barcode is done by a third-party barcode reader. In the example, a comment to simulate this is used. In real life, this could be a Metrologic Orbit reader that is controlled through the HSLBarcodereader library. Please activate this library to read the barcode when working on a real system.

The positions where to get the plates and where to put it after reading are not specified directly, but in an array. An array can be imaged as a table with two columns. In column one, the index is stored. This value is not changeable by the user. It always starts at one. The second column of the array can be filled by the user. Either a variable of type integer, float, and string can be stored, or a sequence name can be stored in this cell.

Index	Value
1	100
2	12.5
3	99
4	"Text"
5	21
6	112132
7	32

Array of VARIABLES

Index	Value
1	MLStar.OutputPlate1
2	MLStar.OutputPlate2
3	MLStar.IntermediatePlate1
4	MLStar.IntermediatePlate3
5	MLStar.IntermediatePlate5
6	MLStar.IntermediatePlate7
7	MLStar.OutputPlate3

Array of SEQUENCES

Far ahead in the method, these arrays can be read out and the values from e.g. the second column can be used as pipetting volumes, number of loops etc. (in case of an array of variables) or these values specify where to aspirate, where to get a plate etc. (in case of an array of sequences).

NOTE

The CO-RE Gripper works on the base of sequences. Plates are moved from one sequence to another. The sequences remain fixed on the deck, but the plates change sequences.

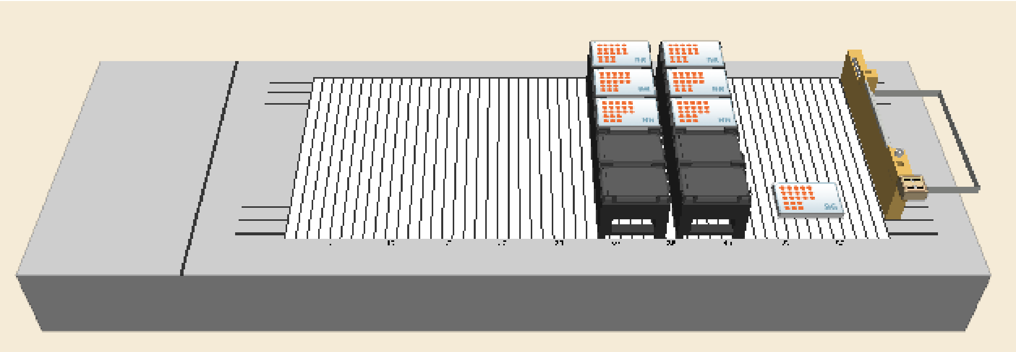
NOTE

Target and source plate positions must be of the same labware type.

Creating the Deck Layout

To work through this example, create the Deck Layout by following these steps:

1. Click the "**Labware**" Tab in the Method Editor.
2. Add two plate carriers "**PLT_CAR_L5MD_A00**" from the Microlab instrument carriers under the Plate Carrier folder onto the deck.
3. Add three plates on each carrier. The plate type could be "**Nun_96_Fl_Lb (low border)**". Name these plates 'Input1' to 'Input3' on one carrier and 'Output1' to 'Output3' on the other carrier.
4. Add a plate "**Nun_96_Fl_Lb (low border)**" directly on the deck. The coordinates should be x = 1100 / y = 150 / z = 150. This will be our position to read the plate's barcode. Name it "**ReadPosition**". External in this case means that is not built in the Microlab instrument, but of course it must be located on the track area so the CO-RE Gripper is able to reach the position.
5. Add the COREGripTool_AtWaste_1000µl to the waste (it will snap).

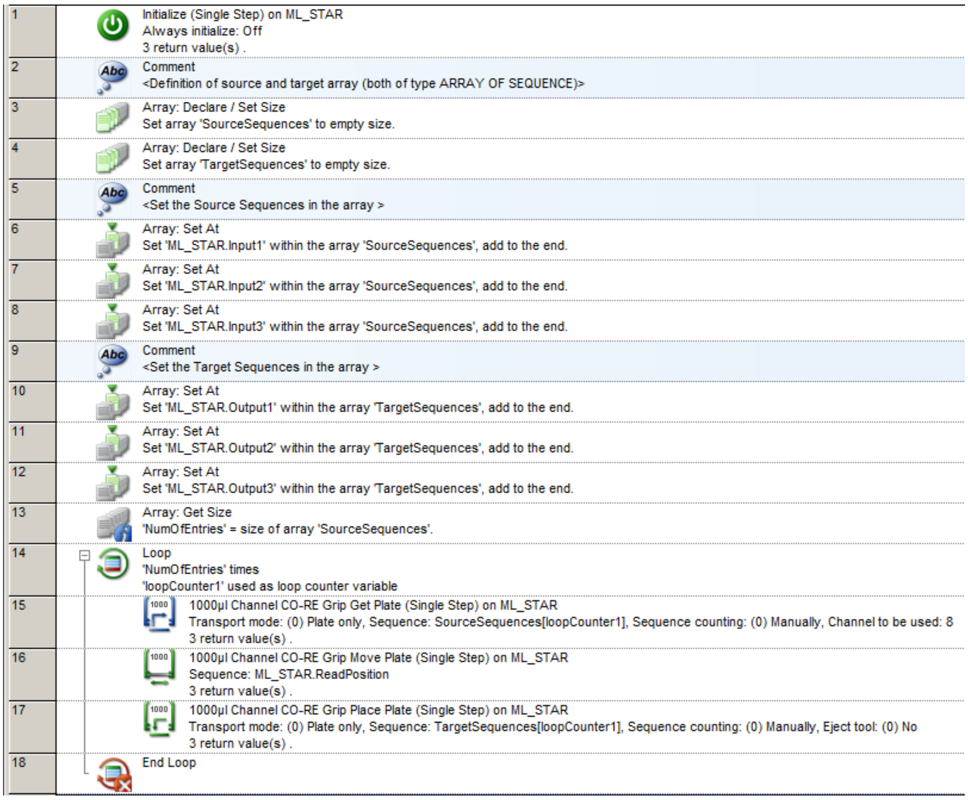


Creating the Sequences

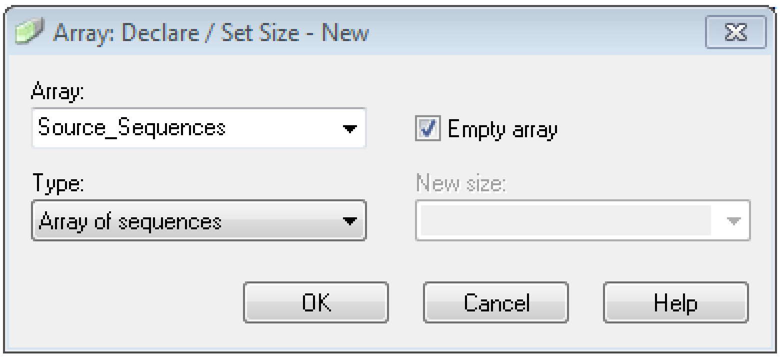
In this example, only the default generated sequences were used.

Creating the Method

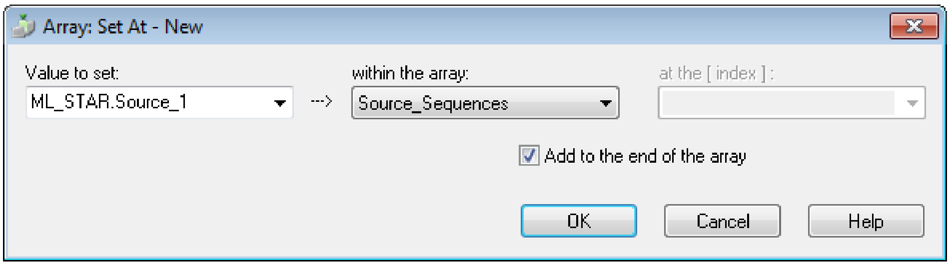
The resulting method should look like shown below:



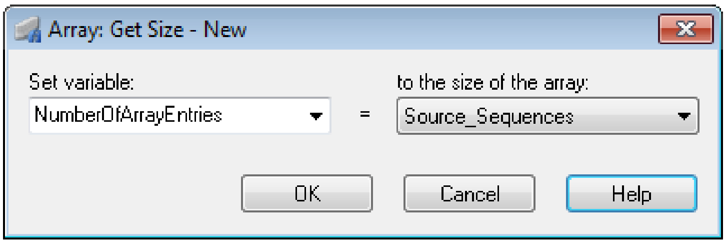
1. The method will start with an initialization step. This step is always needed if a method does not start with a Smart Step or an Easy Step.
2. Steps 3 and 4 are used to create two empty arrays, one for the source- and one for the target sequences. The array: Declare / Set Size step shows that an array of sequences is generated and shall be empty.
3. Empty array means that the number of entries the array will have is unknown. However, if the number of entries is already known, specify the array with the corresponding size. To do this, use the **[New size]** field.



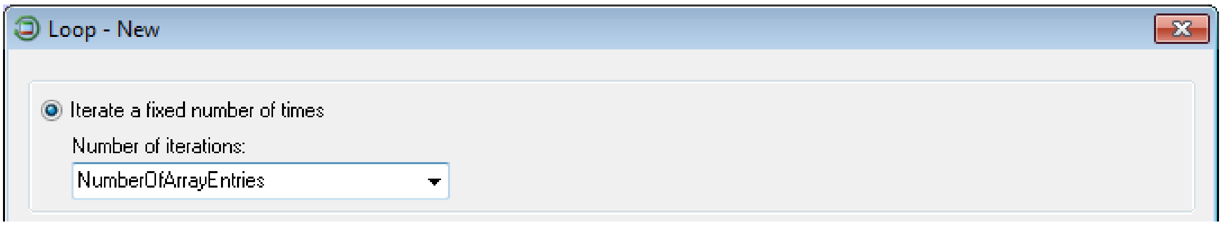
4. The next sets of steps (6-8 and 10-12) are used for filling both the array of GetPlatePositions and the array of PlacePlatePositions. Using the option "**Add to the end of the array**" will store the sequences in order of adding:



5. Steps 6 to 8 and 11 to 12 are writing sequences into the newly declared arrays of sequences.
6. Step 13 is used to find out how many entries are stored in an array. It will store the number of array entries into the variable NumOfEntries. This is especially used if the number of entries is unknown (e.g. if reading values or sequences from a file).

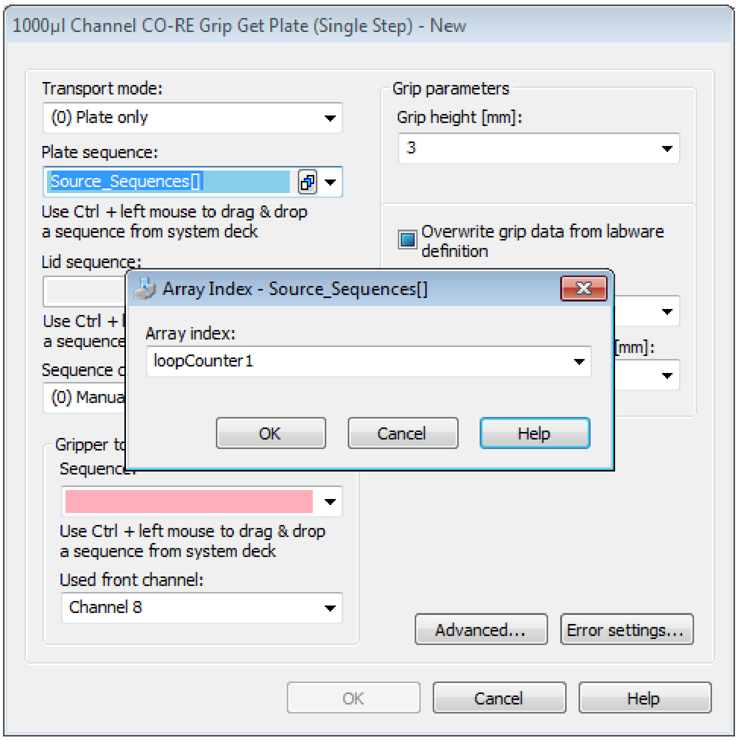


7. The number of entries can be used as the number of loop runs:

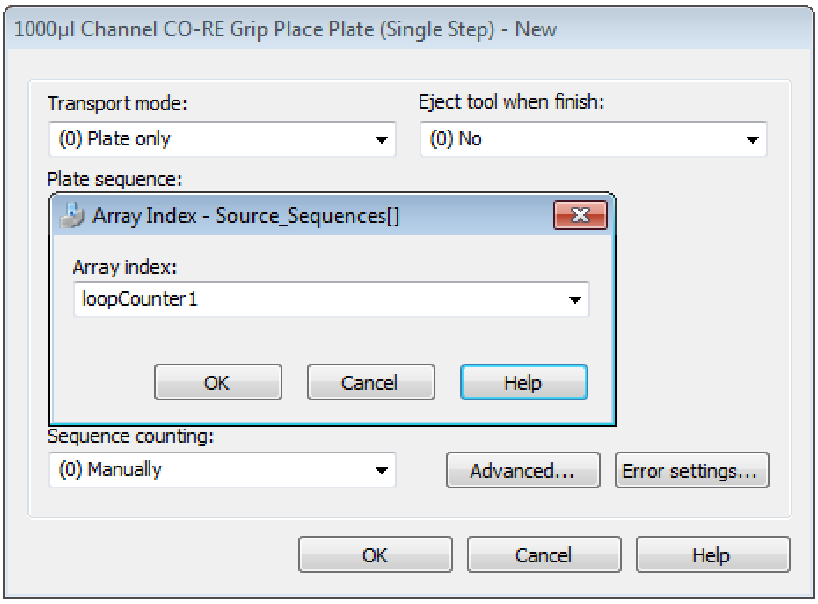


8. The GetPlate step in line 15 does **not** have a sequence included. The sequence is read out of the array 'GetPlatePositions' and the position where to read is given by the LoopCounter1 variable.

9. Click the  sign to set the array index seen below.

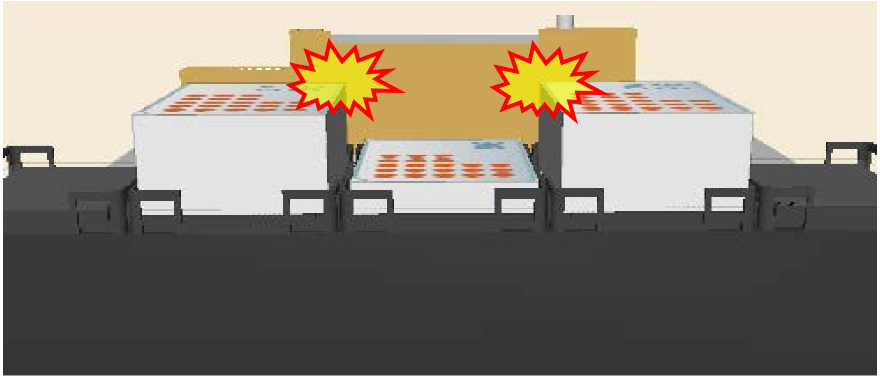


10. In the first loop round, the GetPlate step is using the sequence stored in the array under position 1 (specified by the loopCounter1).
11. The MovePlate in step 16 will move the plate to the barcode reader position. In the trace file, the barcode of the plate can be seen.
12. Step 17 will place the plate on a specific position. This position is defined by a sequence and stored in the second array. All of these must be done similar to the GetPlate step in line 16: simply define the array and the index then the sequence will be read out of the array.



16.3 CO-RE Gripper Transport: Avoiding Z-Step Loss

Keep in mind that the CO-RE Gripper for the 5 mL-pipetting channel needs more space than the 1000µl-pipetting channel CO-RE Gripper tools. This can lead to Z-Step Loss when a Deck Layout is not properly set up (see below).



Side view of the instrument (turned towards the waste)

16.4 CO-RE Gripper Transport: Avoiding Y-Step Loss

The 5 mL-pipetting channels are less flexible than the 1000µl-pipetting channels due to their massive construction. Therefore, it makes sense to try low grip forces for labware transports.

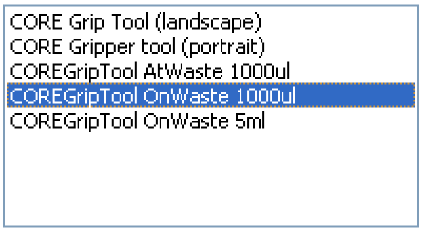
In the event of heavy labware (e.g. filled Deep Well Plates), switching to higher grip forces can be done.

16.5 CO-RE Gripper Transport: Adding the Suitable Tool

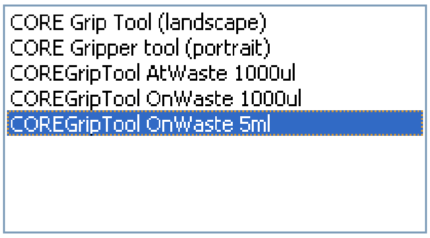
The Deck Layout Editor shows several tools for the CO-RE Gripper. The next statement is the correct use of the labware.

COREGripTool OnWaste 1000µl uses a holder for the paddles mounted on top of the waste block. This is a holder that is capable to hold both 1000µl-pipetting channel and 5 mL-pipetting channel paddles.

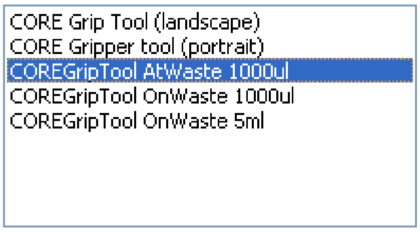
Using this labware will create the default sequence for 1000µl-pipetting channel paddles.



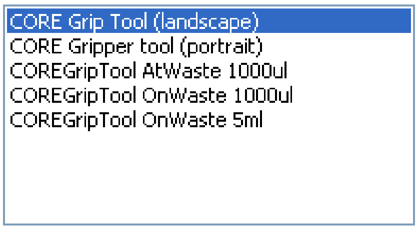
COREGripTool OnWaste 5 mL uses a holder for the paddles mounted on top of the waste block. This is a holder that is capable to hold both 1000µl-pipetting channel and 5 mL-pipetting channel paddles. Using this labware will create the default sequence for 5 mL-pipetting channel paddles.



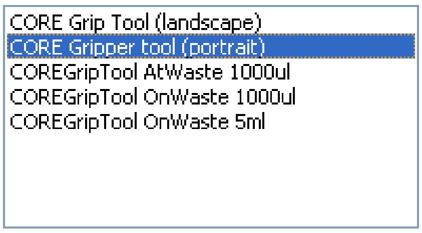
COREGripTool AtWaste 1000µl uses a holder for the paddles mounted on the side of the waste block (only available for 1000µl-pipetting channels).



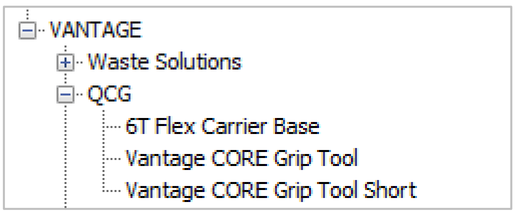
CORE Grip Tool (landscape) uses a holder for the paddles on a landscape plate position (only available for 1000µl-pipetting channels).



CORE Grip Tool (portrait) uses a holder for the paddles on a portrait plate position (only available for 1000µl-pipetting channels).



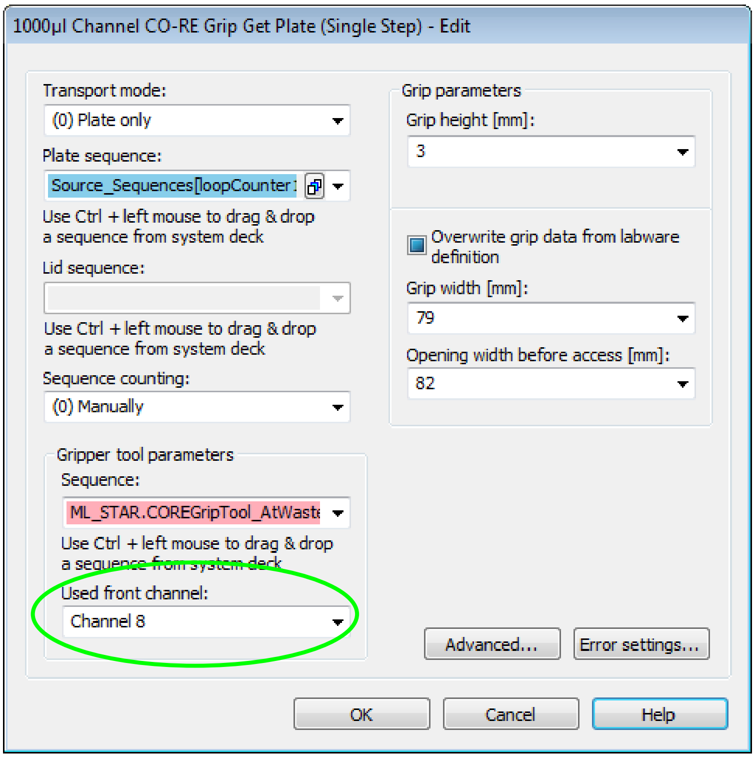
Quad CORE Grip Tools can be found under the VANTAGE category in the deck editor.



16.5.1 Use Various Channels for CO-RE Gripper Transports

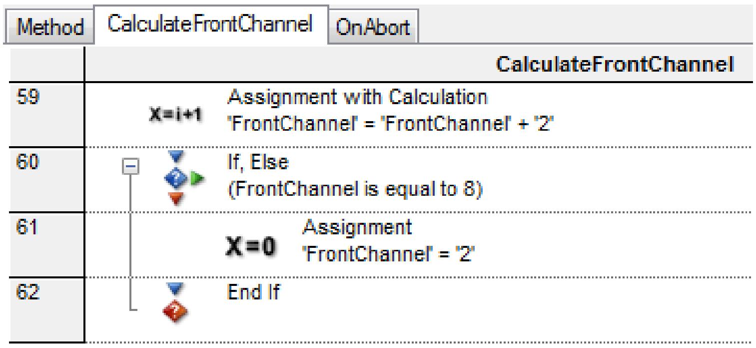
By default, pipetting channels 7 and 8 are used to execute CO-RE Gripper transports. Having a method with many transport steps, these two pipetting channels are stressed more than the others. The input field "**Used Front Channel**" was designed to use different pipetting channels for the transports, which shares the applied load to all pipetting channels of an instrument.

The following program lines make sure that first, pipetting channels 1+2 are used. In the next "**Get Plate**" Command, pipetting channels 3+4 will be used (and so on). In this manner, the pickup commands are spread over all the pipetting channels and the mechanical system of all pipetting channels are strained to the same degree.

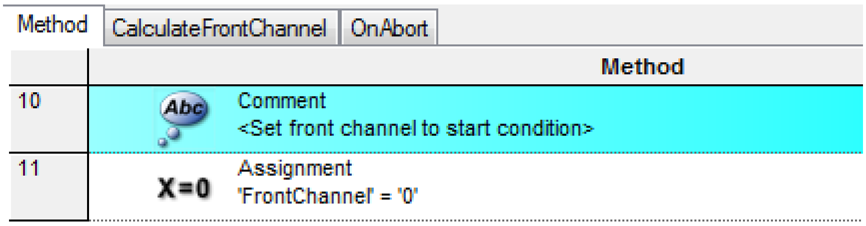


1. Create a sub-method called "**CalculateFrontChannel**"
2. Make an assignment with calculation: "**FrontChannel = FrontChannel + 2**"
3. Make an if statement: "**If FrontChannel is greater than 8* → FrontChannel = 2**"
4. Having a different number of pipetting channels on the instrument, enter e.g. 4, 12, 16
5. Now, make sure that the "**FrontChannel**" Variable is global and valid in all sub-methods:
6. Click on "**Method → Export Local Variables"** and select *FrontChannel*, then click **[OK]**.

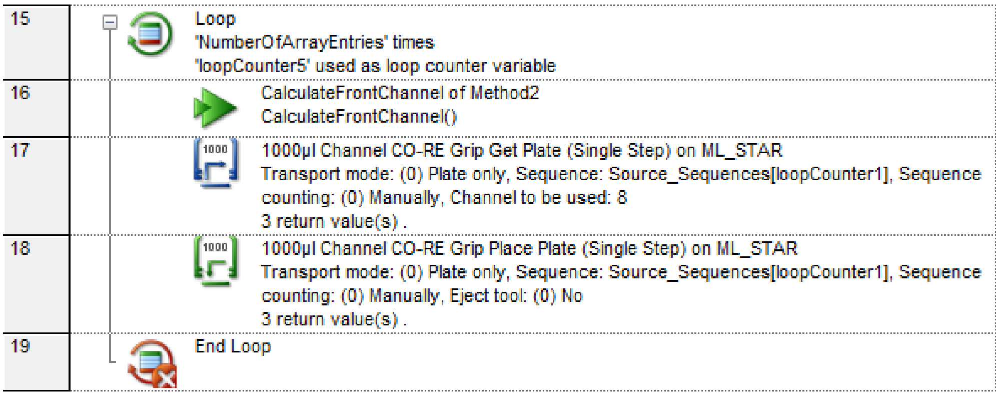
7. The sub-method should look like this:



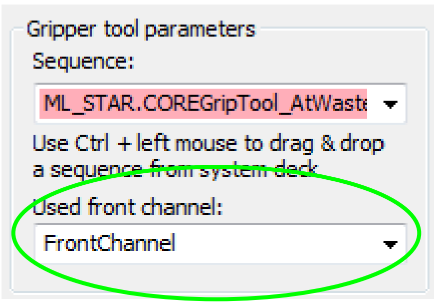
8. All that remains to be done is to set a start value in the main method:



9. Insert the CalculateFrontChannel sub method in the method right before the GetPlate, as seen below.

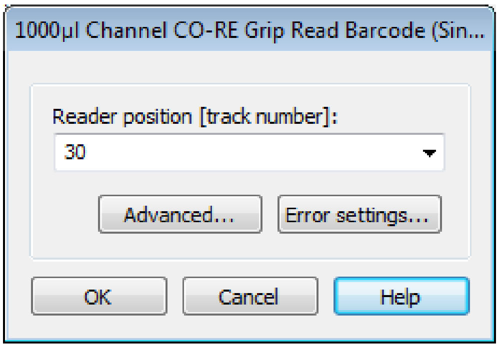


10. In the CO-RE Gripper "**Get Plate**" Command, set the "**FrontChannel**" Variable.

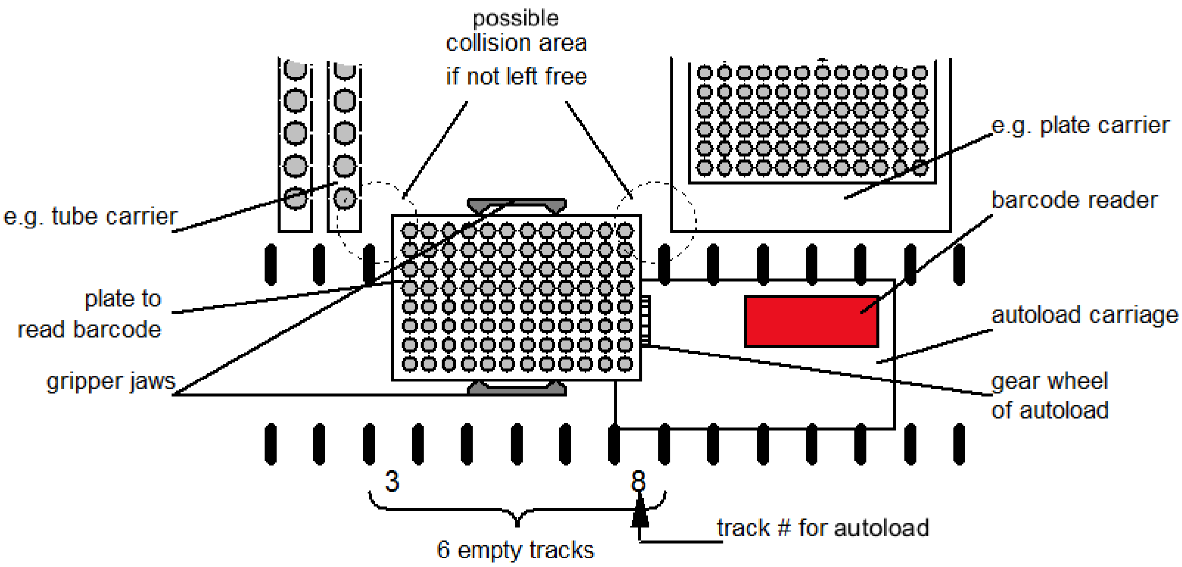


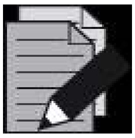
16.5.2 Read Plate Barcode using the CO-RE Gripper (only available with Autoload)

If the command "**CO-RE Gripper Read Barcode**" is used, like this:



Six tracks on the left side of the barcode reading position (inclusive) have to be empty, otherwise the plate will crash into the carriers on these positions. For example, if the track number of the barcode reader (Autoload) is set to 8, track numbers 3-8 have to be empty. The following sketch shows the situation:





NOTE

It is recommended to use the pipetting *channels closest to the front for transporting plates to the barcode reader.*

17 Tube Handling Tools

17.1 Tube-Gripper

This section provides all necessary information about the Tube-Gripper. This tool can be used to move tubes to a barcode reader (if the Autoload option is not installed), to load/unload tubes to an integrated centrifuge, to pick specific tubes from multiple carriers into one target carrier, to move tubes to a balance (measuring evaporation), etc.

This tool is mounted on a pipetting channel and can be moved over the deck in the same way as the pipetting channels.

17.1.1 Using the Tube-Gripper Steps

First, make sure that the Tube-Gripper is activated in the Hamilton System Configuration Editor. Set both the Tube-Gripper and the Autoload option line to ‘Available’ as shown below:

⊟ Simulator configuration	
1000µl Channel: number of channels	8
1000µl Channel: raster	9mm
5ml Channel: number of channels	0
5ml Channel: raster	36mm
Autoload	Not available
Camera Channel	Not available
CO-RE 384 Head	Not available
CO-RE 96 Head	Not available
iSWAP	Small gripper
Nano pipettor	Not available
Pump Station 1 (node HW)	Not available
Pump Station 2 (node HU)	Not available
Pump Station 3 (node HV)	Not available
Simulation	Off
Simulator Delay	0%
S-Tube Capper: number of channels	0
S-Tube Capper: raster	36mm
Temperature-controlled Carrier 1	Not available
Temperature-controlled Carrier 2	Not available
Tube Gripper	Available
Wash station 1	Not available
Wash station 2	Not available

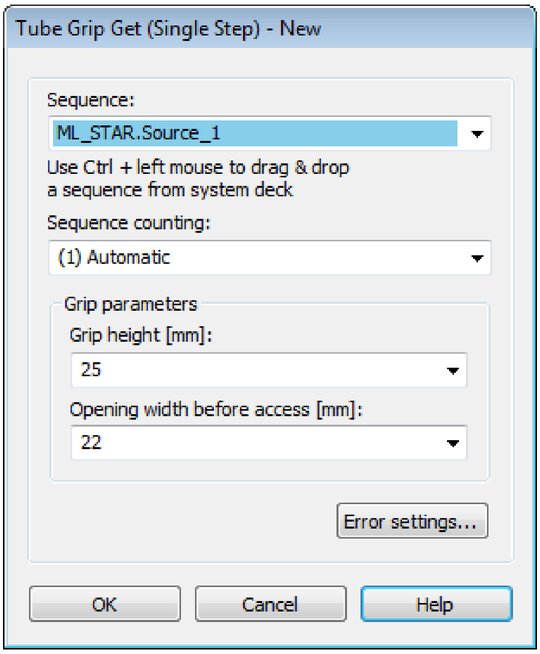
In the Method Editor, the Tube-Gripper steps are found:

- Tube Grip Get (Single Step)
- Tube Grip Place (Single Step)
- Tube Grip Move (Single Step)
- Tube Grip Read Barcode (Single Step)

"Tube Grip Get"

This step is used to pick a tube. The parameters to pass are the following:

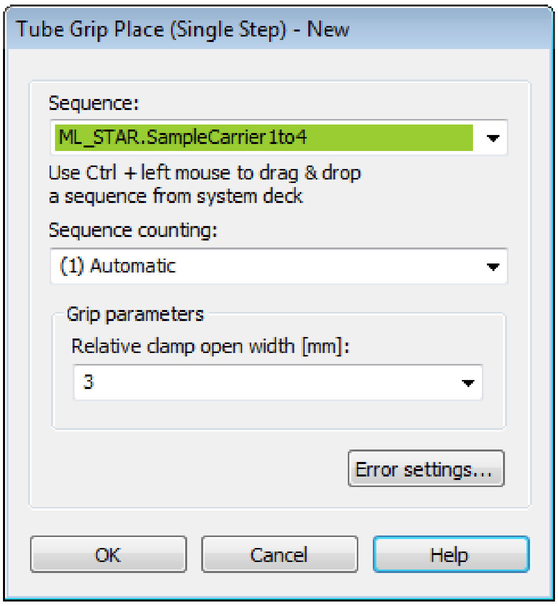
- Sequence (where to get the tube)
- Sequence counting (manually, to leave the current position untouched)
- Grip height in mm, measured from the top of the tube
- Opening width before access in mm (maximum = 24 mm)



"Tube Grip Place"

This step is used to place a gripped tube. The parameters to pass are the following:

- Sequence (where to put the tube)
- Sequence counting (manually, to leave the current position untouched)
- Relative clamp open width in mm.



NOTE

The relative clamp open width is a relative value. If the gripped tube requires a clamp position of 20 mm, there is only 4 mm left to open (because 24 mm is the maximum opening width).

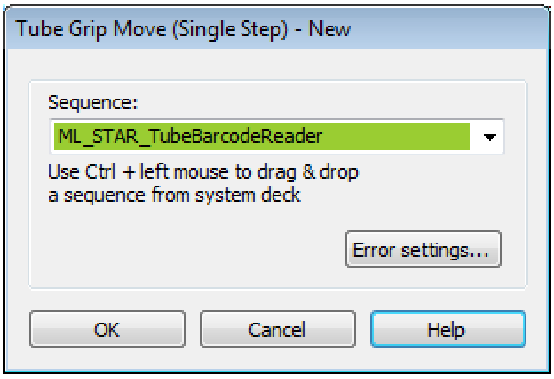
"Tube Grip Move"

This step is used to move a gripped tube to a specific position without opening the clamp. This could be used e.g. to move the tube in front of an external Barcode Reader.

The parameter to pass is the:

- Sequence (where to move [and hold] the tube).

Keep in mind that the step will move the tube to the current position of the sequence that has been passed.



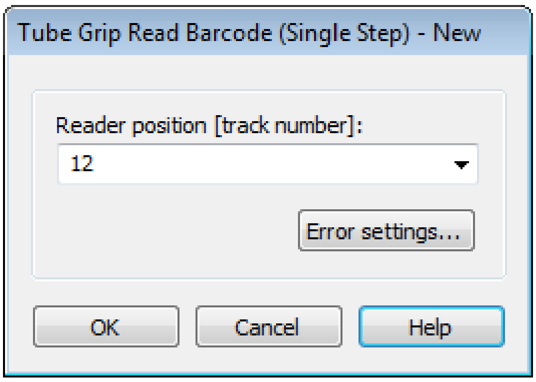
"Tube Grip Read Barcode"

This step is used to move a gripped tube to a specific position without opening the clamp. This step will grip a tube and bring it to the Autoload Barcode Reader for reading.

The parameter to pass is the:

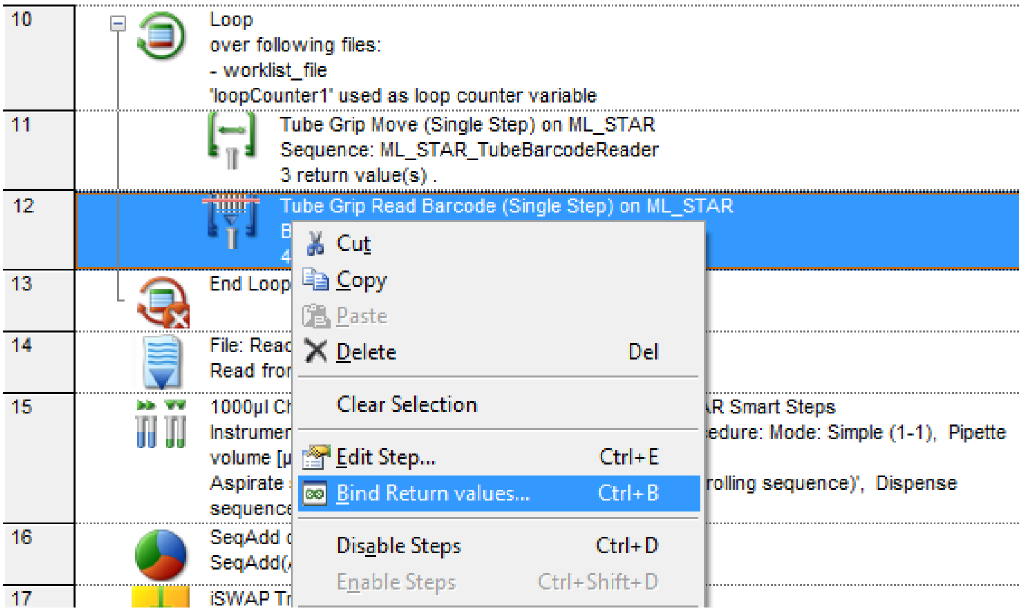
- Reader position [track number]

This value specifies the track, where the Autoload Barcode Reader will move to. After the movement to this position, the Tube-Gripper will bring the tube in front of the reader and read the barcode.

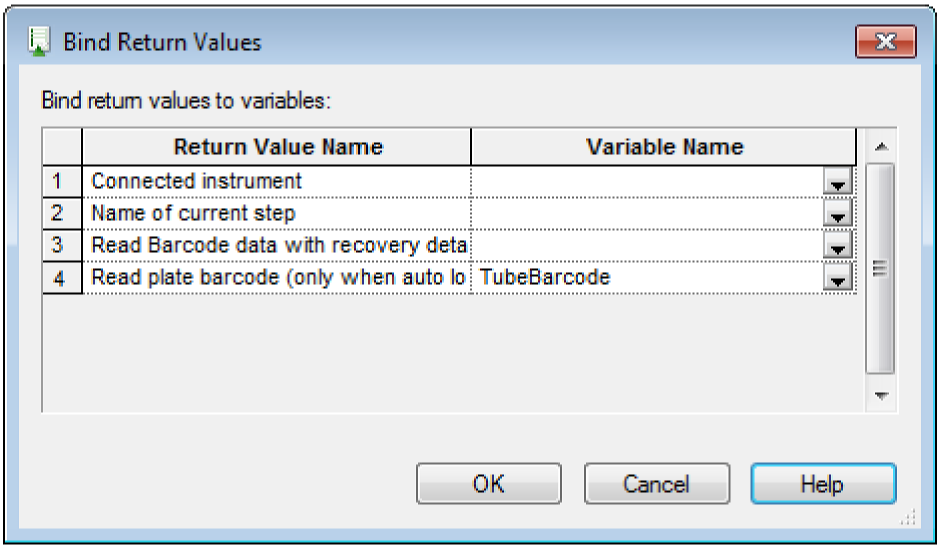


In order to link the value of the barcode to a variable, the return value of the step has to be linked.

1. To do so, right-click the Tube Grip Read Barcode in the method and select 'Bind Return values':



2. A window opens to link the different return values to variables. The value from line 4 (which is the barcode) will be used in the example as seen in the image below.

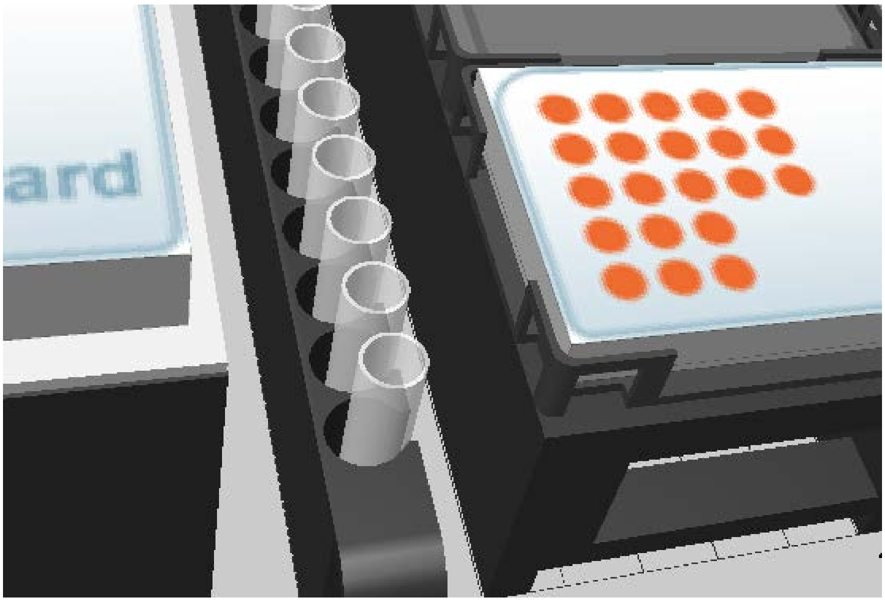


3. Type in or select a variable that will hold the value of the barcode. Finish with **[OK]**.

NOTE

The Tube-Gripper is able to grip tubes with an outer diameter of 8 to 22 mm.

The Tube-Gripper clamps need additional space when gripping tubes with large diameter (15 – 22 mm). In an event like this, leave a space on the left and right of the tube carrier.



18 Hamilton Heater Shaker

The Hamilton Heater Shaker (HHS) is designed to heat and/or shake standard micro plates in SBS format.

To operate the HHS, a specific library is required. The maximum shaking speed depends on the labware used. For details, see Section 11.12.23 HHS Library. Shaking can be performed clockwise or counter-clockwise.

Before the heating or shaking is started, the plates are locked and positioned in the center of the HHS. When heating or shaking has been finished, the plates are unlocked and can then easily be removed from the HHS.

The HHS can be heated to temperatures up to 105°C. The temperature is constantly measured by two sensors, one located in the middle and one at the edge of the adapter plate.

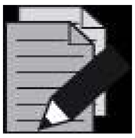
18.1 Programming the Hamilton Heater Shaker

The following two examples demonstrate the use of the HHS.

Before using the HHS, a connection and device number has to be generated. After the connection has been established, the Heater Shaker will be initialized.

The library offers two commands for this task, depending on the kind of connection determined by the hardware. For connections made via TCC, use the "**CreateStarDevice**" Command (used for connecting 1 or 2 Heater Shakers); otherwise use the "**CreateUSBDevice**" Command. In both cases, a device number is generated that can be stored in a variable. This variable must be used in all other commands of the library to control the Heater Shaker.

Library Function		
Number of HHS	**Icon**	**Action Performed**
1 – 2		Create Star Device (Establish a connection to 1 or 2 shakers connected via TCC)
1 – 8		Create USB Device (Establish a connection to up to 8 Heater Shakers using the Heater Shaker box)

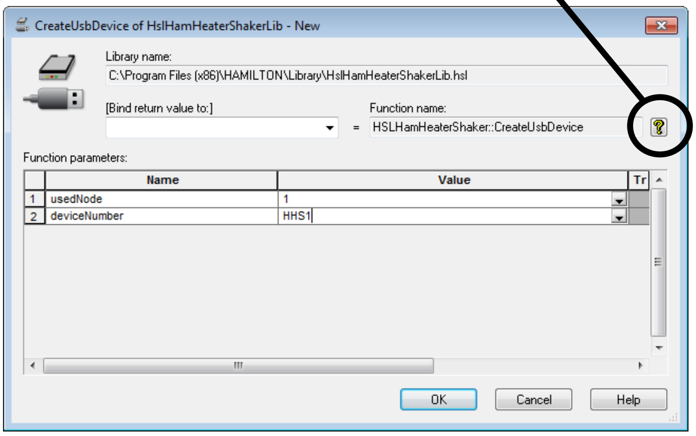


NOTE

*It is required to establish a connection to all Heater Shakers that will be used in the method. If, for example, only two out of four shakers have been initialized, only these two shakers can be started using the "**StartAllShaker**" Function.*

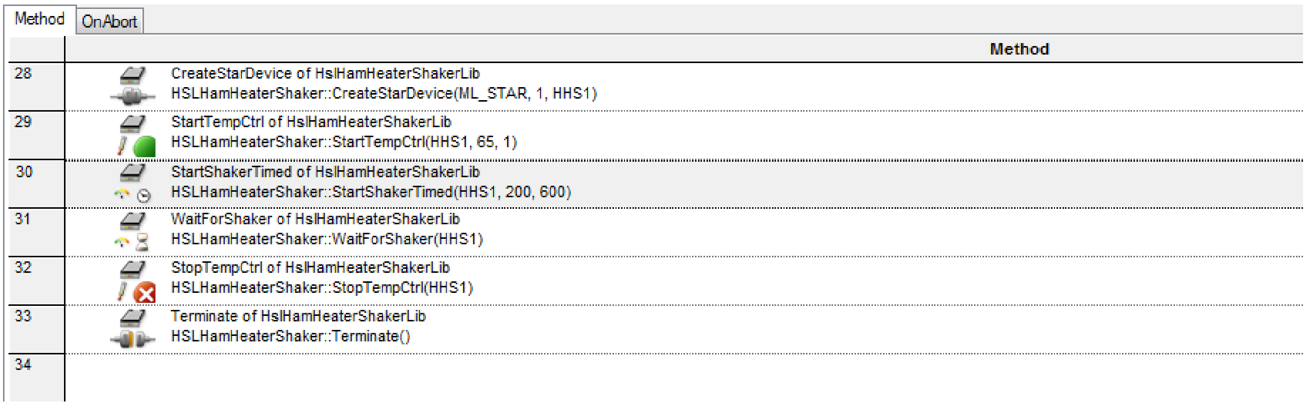
NOTE

The *<u>VENUS Help Function</u>* *contains a description of all functions of the Heater Shaker library with a description of parameters passed to the functions or obtained by them. The help documentation can be accessed by clicking on the yellow question mark within the dialog windows:*



18.1.1 Example 1: Controlling One Hamilton Heater Shaker

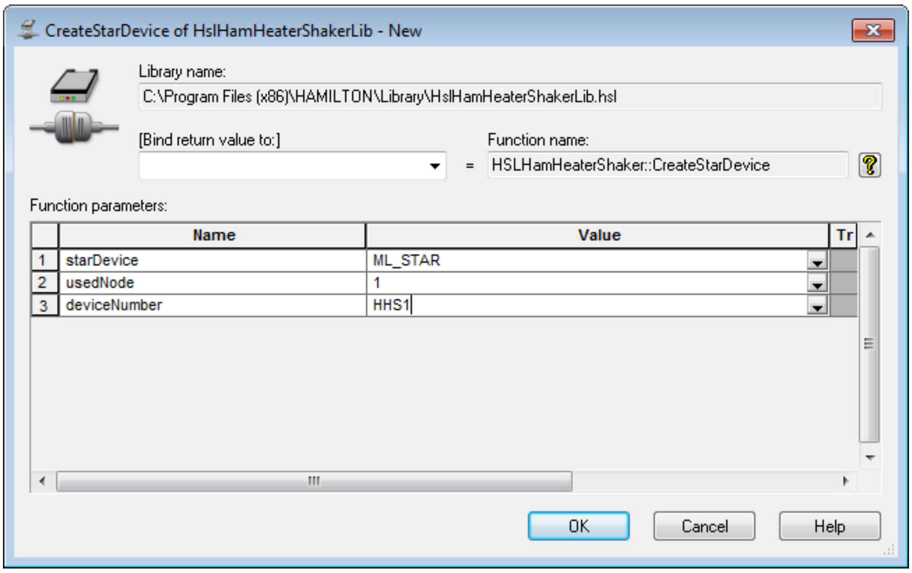
In this example, a single Heater Shaker is connected via the Microlab instrument TCC connector. The Heater Shaker is heated to 65°C followed by 10 mins shaking at 200 rpm. A detailed description of all the steps can be found in the following section.



Overview of a method to run the Heater Shaker at 200 rpm for 10 minutes at 65°C

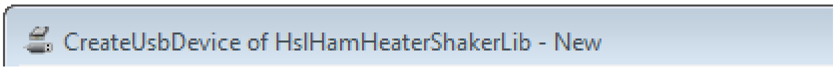
18.1.1.1 Step by Step Analysis of Example 1

Step 1: Create a connection to the shaker through the "**CreateStarDevice**" Command. The node can be "**1**" or "**2**" if the Heater Shaker is connected via TCC. The device number can be assigned, which is generated by this command, to any variable of choice. This variable has to be used for all subsequent commands regarding the control of the Heater Shaker.

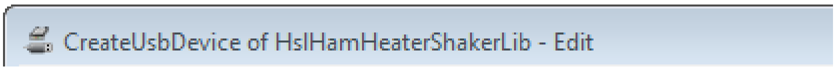


NOTE

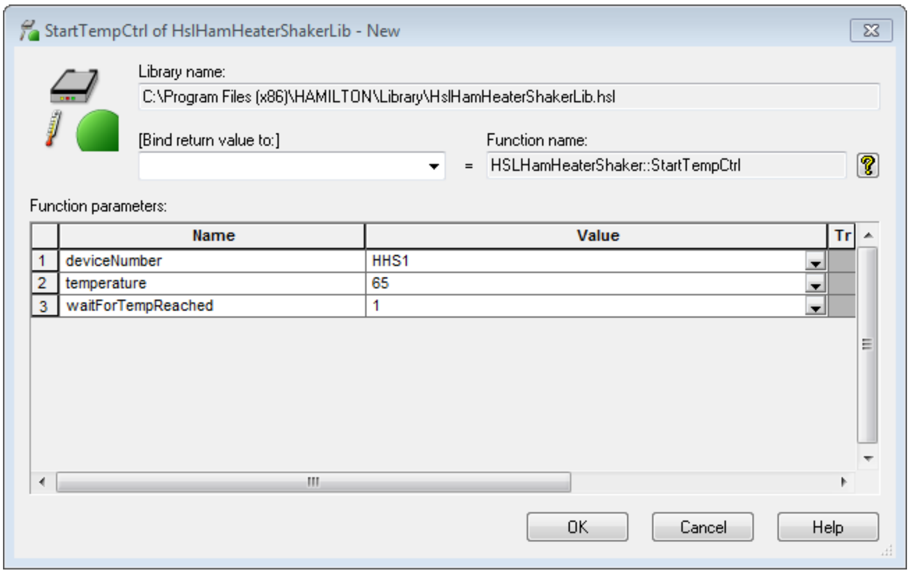
*When inserting a step in the method for the first time, on the toolbar, the "**New**" Add-on is imprinted.*



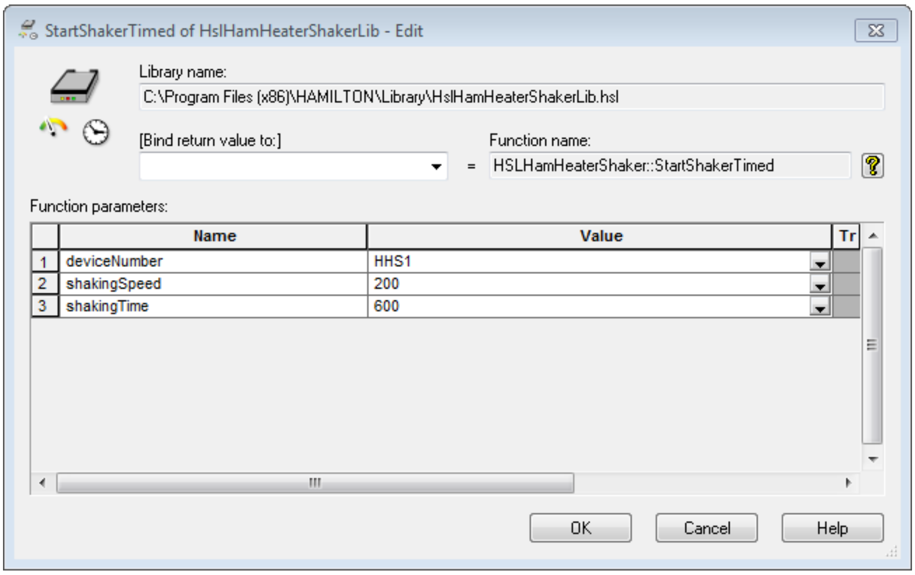
*If there has been already a method inserted, when editing, the add-on "**Edit**" is imprinted.*



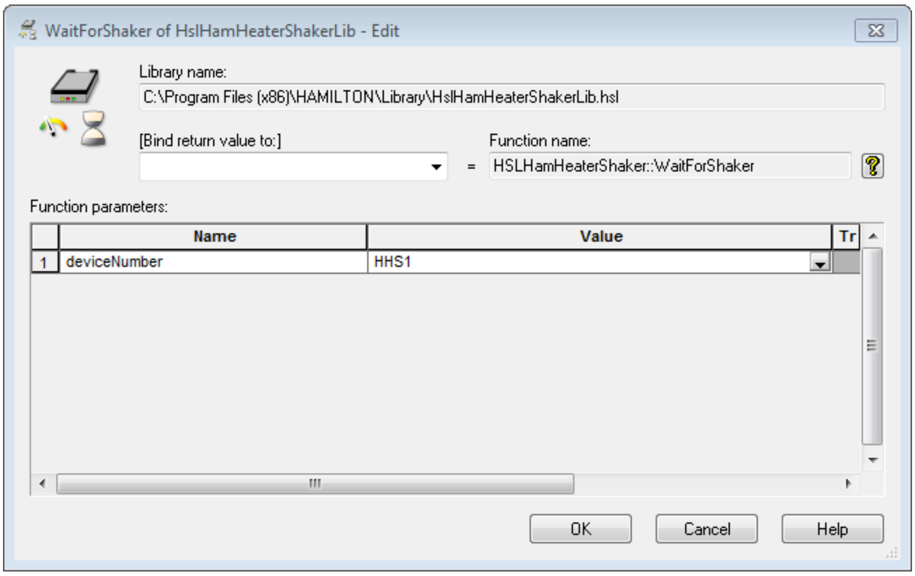
Step 2: Use the variable containing the device number to access the correct heater shaker. Set the temperature to 65°C. If the "**waitForTempReached**" Option is chosen with the setting "**1**", the method will pause at this step until the defined temperature has been reached and is stable for 180 seconds. Only then the method will continue to the next step. If a pause to the method is not wanted but rather carry out other tasks in parallel to the heating process, the "**waitForTempReached**" should be set to "**0**". At a later time, the temperature can be checked and if necessary, wait for the heating process to finish using the "**WaitForTempCtrl**" Command.



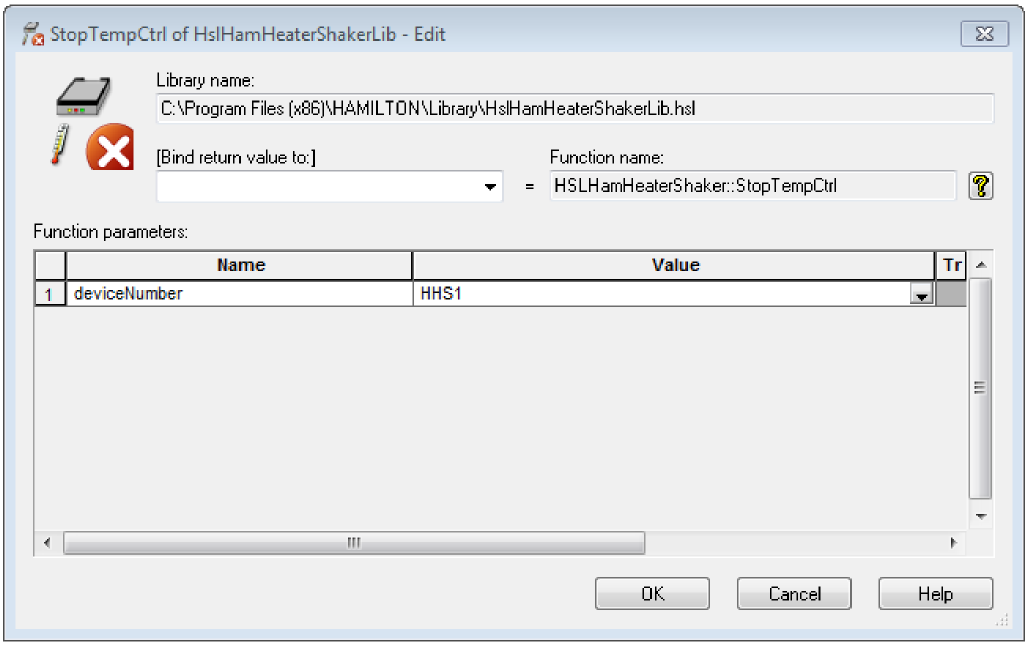
Step 3: Here, the shaking parameters, shaking speed and duration of shaking can be set. The speed is defined in rpm and the time in seconds. The shaking speed ranges from 30 rpm to 2500 rpm. The maximum speed depends on the orbit and adapter of the shaker and must not exceed the maximum given in the "**Technical Specifications**" (Please refer to the <u>Technical Specifications in the Corresponding Microlab instrument Operator´s Manual)</u>. Any function to start shaking will also close the plate lock automatically.



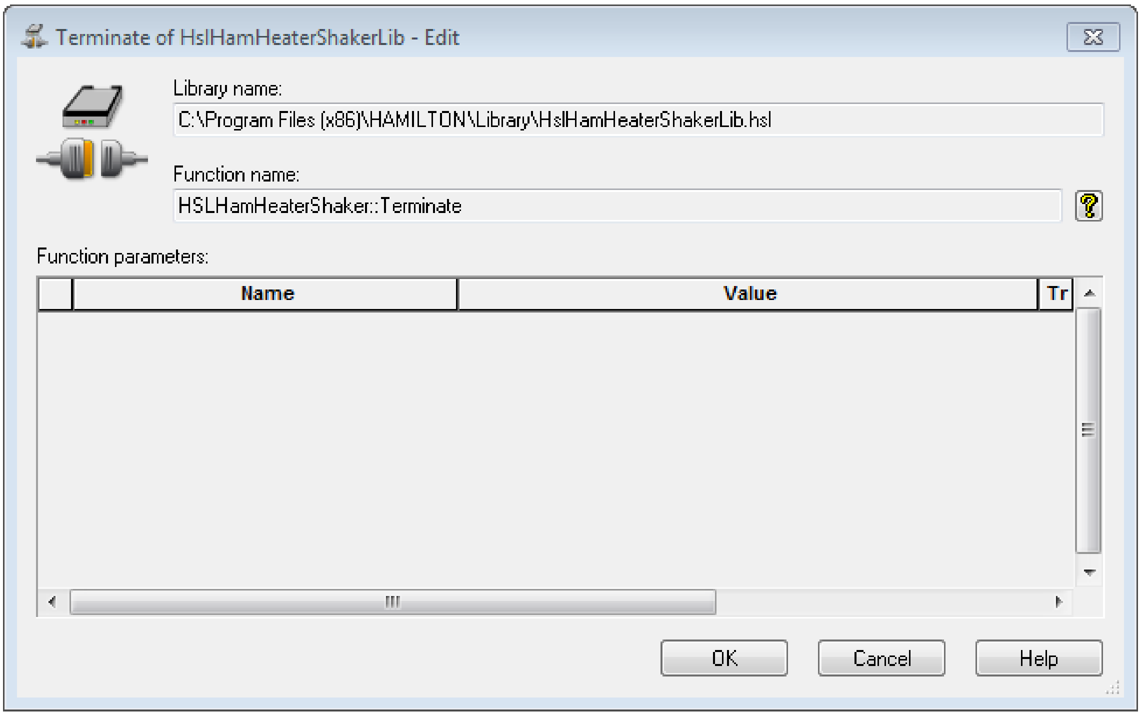
Step 4: This command will wait until the shaker defined by the device number has finished the timed shaking process. If the shaker is already finished before this function is called, the method will immediately proceed with the next step. This function will also open the plate lock.

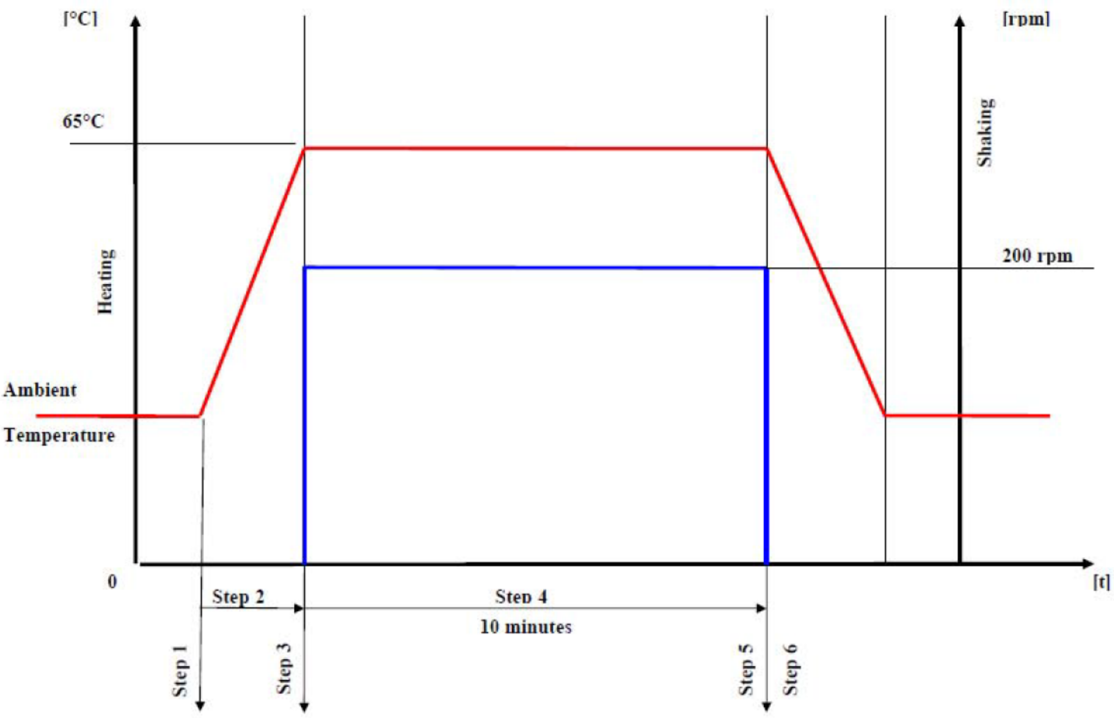


Step 5: As heating is controlled independently from shaking, the heating process has to be terminated explicitly. Otherwise, the Heater Shaker will continue heating even if it is not used anymore.



Step 6: If the Heater Shaker will not be used any longer, the connection can be terminated. At the end of a method or upon abort, the connection is automatically terminated and will stop heating as well as shaking. The plate lock will then open.





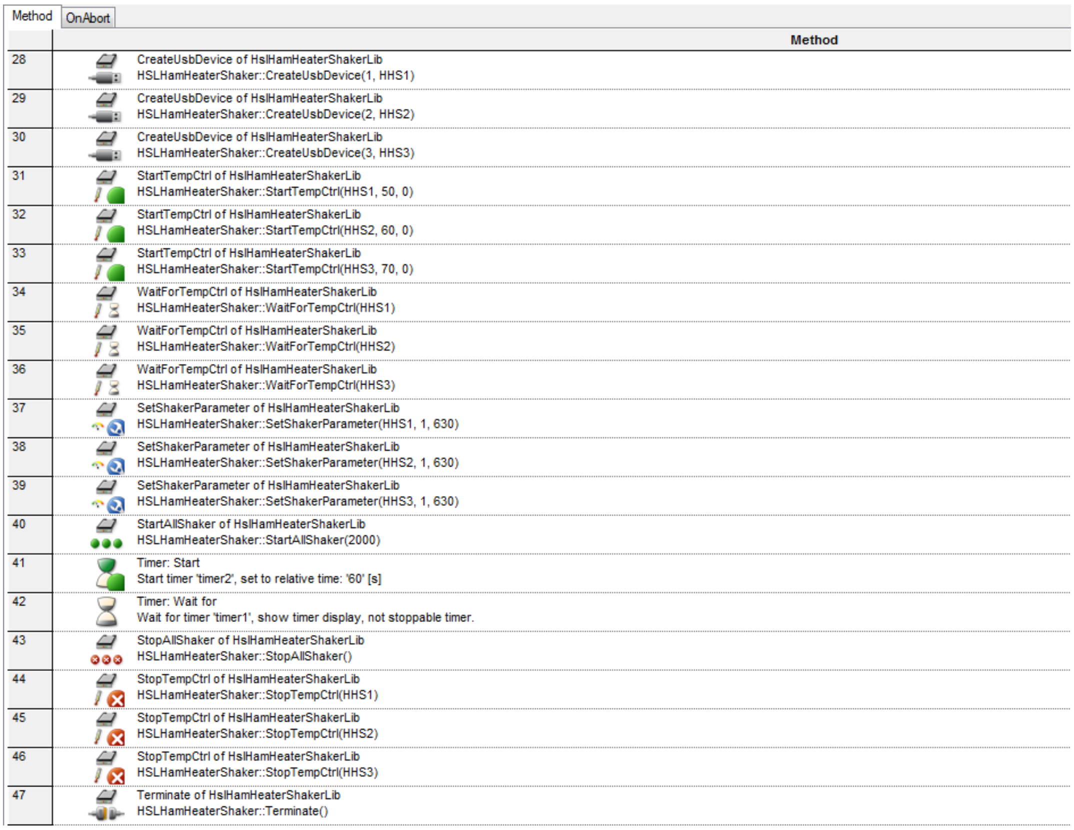
The figure above shows a schematic view of the heating and shaking process of Example 1.

18.1.2 Safety Measures Upon Method Abort

As soon as the method is aborted, the heater shaker will automatically be stopped. This implies that the heating and shaking process is stopped. The plate lock is opened and the connection to the heater shaker is terminated. No further precautions within the submethod **"OnAbort"** are required.

18.1.3 Example 2: Controlling Multiple Hamilton Heater Shakers

This example shows the usage of multiple Heater Shakers via USB and the Heater Shaker box. The connection via USB and HSB is needed in order to control more than two Heater Shakers. The method is using three Heater Shakers with different temperature settings.



Method | OnAbort

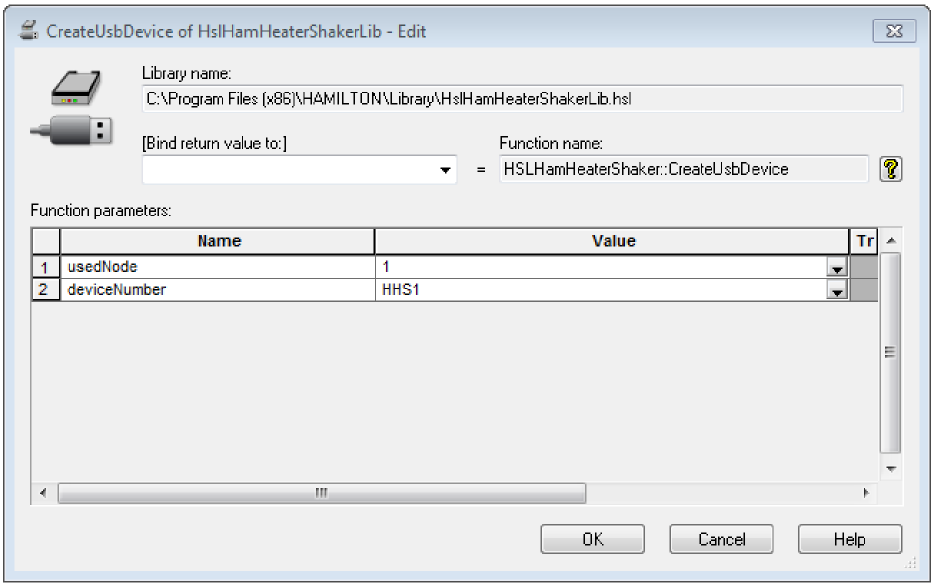
	Method
28	CreateUsbDevice of HslHamHeaterShakerLib HSLHamHeaterShaker::CreateUsbDevice(1, HHS1)
29	CreateUsbDevice of HslHamHeaterShakerLib HSLHamHeaterShaker::CreateUsbDevice(2, HHS2)
30	CreateUsbDevice of HslHamHeaterShakerLib HSLHamHeaterShaker::CreateUsbDevice(3, HHS3)
31	StartTempCtrl of HslHamHeaterShakerLib HSLHamHeaterShaker::StartTempCtrl(HHS1, 50, 0)
32	StartTempCtrl of HslHamHeaterShakerLib HSLHamHeaterShaker::StartTempCtrl(HHS2, 60, 0)
33	StartTempCtrl of HslHamHeaterShakerLib HSLHamHeaterShaker::StartTempCtrl(HHS3, 70, 0)
34	WaitForTempCtrl of HslHamHeaterShakerLib HSLHamHeaterShaker::WaitForTempCtrl(HHS1)
35	WaitForTempCtrl of HslHamHeaterShakerLib HSLHamHeaterShaker::WaitForTempCtrl(HHS2)
36	WaitForTempCtrl of HslHamHeaterShakerLib HSLHamHeaterShaker::WaitForTempCtrl(HHS3)
37	SetShakerParameter of HslHamHeaterShakerLib HSLHamHeaterShaker::SetShakerParameter(HHS1, 1, 630)
38	SetShakerParameter of HslHamHeaterShakerLib HSLHamHeaterShaker::SetShakerParameter(HHS2, 1, 630)
39	SetShakerParameter of HslHamHeaterShakerLib HSLHamHeaterShaker::SetShakerParameter(HHS3, 1, 630)
40	StartAllShaker of HslHamHeaterShakerLib HSLHamHeaterShaker::StartAllShaker(2000)
41	Timer: Start Start timer 'timer2', set to relative time: '60' [s]
42	Timer: Wait for Wait for timer 'timer1', show timer display, not stoppable timer.
43	StopAllShaker of HslHamHeaterShakerLib HSLHamHeaterShaker::StopAllShaker()
44	StopTempCtrl of HslHamHeaterShakerLib HSLHamHeaterShaker::StopTempCtrl(HHS1)
45	StopTempCtrl of HslHamHeaterShakerLib HSLHamHeaterShaker::StopTempCtrl(HHS2)
46	StopTempCtrl of HslHamHeaterShakerLib HSLHamHeaterShaker::StopTempCtrl(HHS3)
47	Terminate of HslHamHeaterShakerLib HSLHamHeaterShaker::Terminate()

18.1.3.1 Step by Step Analysis of Example 2

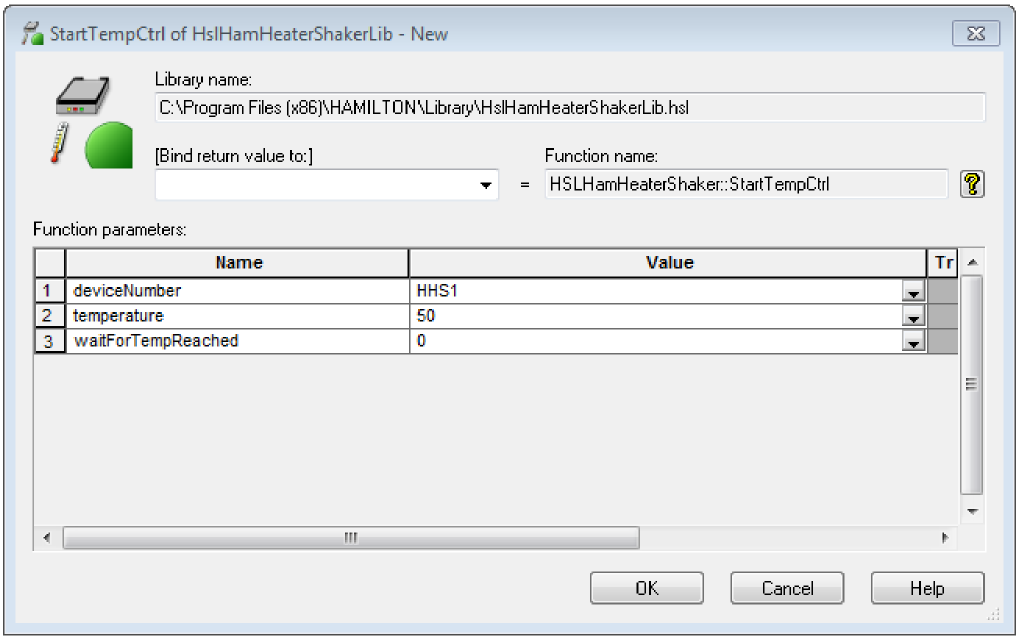
Step 1-3: Create a connection to all heater shakers to be used in this method using the function "**CreateUSBDevice**". It is required to initialize each shaker individually in order to create a device number and to control the Heater Shaker throughout the method. Heater Shakers that are not initialized cannot be started with the "**StartAllShaker**" Function.

The usedNode can range from 1 to 8, depending on the number of Heater Shakers in use. Node 1 corresponds to the master heater shaker.

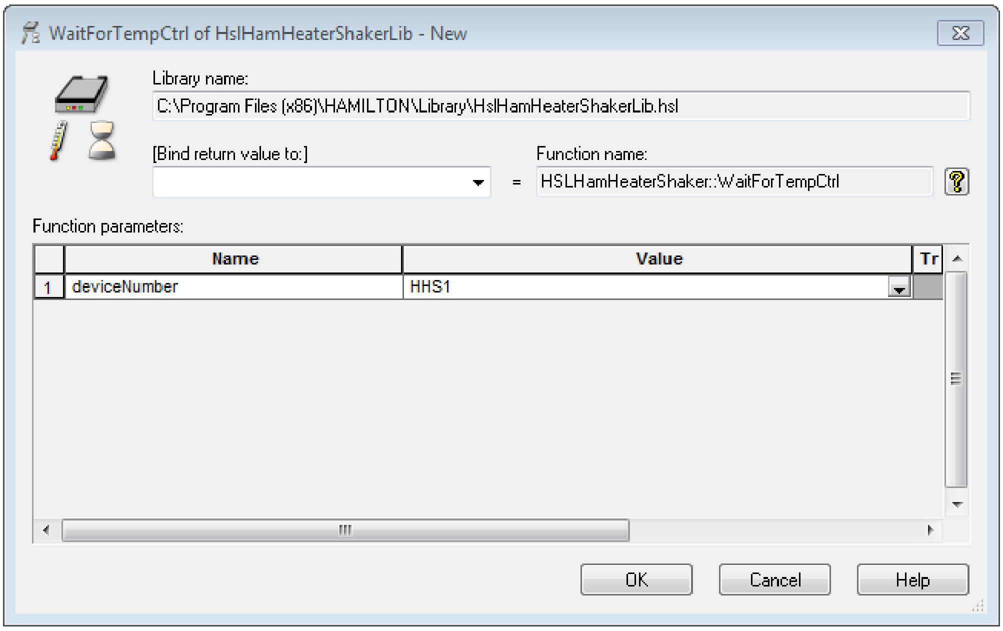
The deviceNumber is stored in a variable, which is needed to control the Heater Shaker in all subsequent functions.



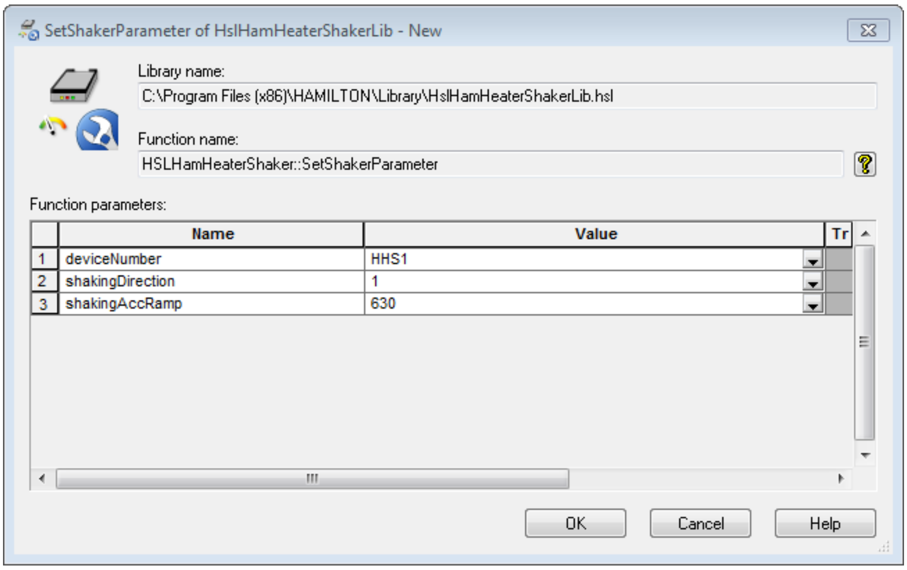
Step 4-6: Set the temperatures for each individual Heater Shaker, for example to 50°, 60° and 70°C as shown below. If working with several Heater Shakers, the option "**waitForTempReached**" has to be set to "**0**" so that all shakers are heated in parallel.



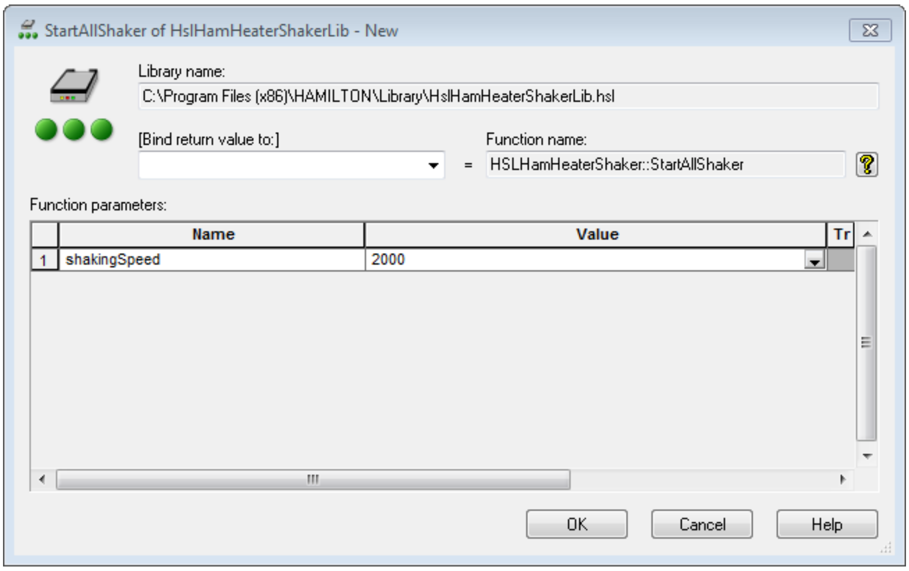
Step 7-9: Wait until all Heater Shakers have reached the set temperature before proceeding to the next step.



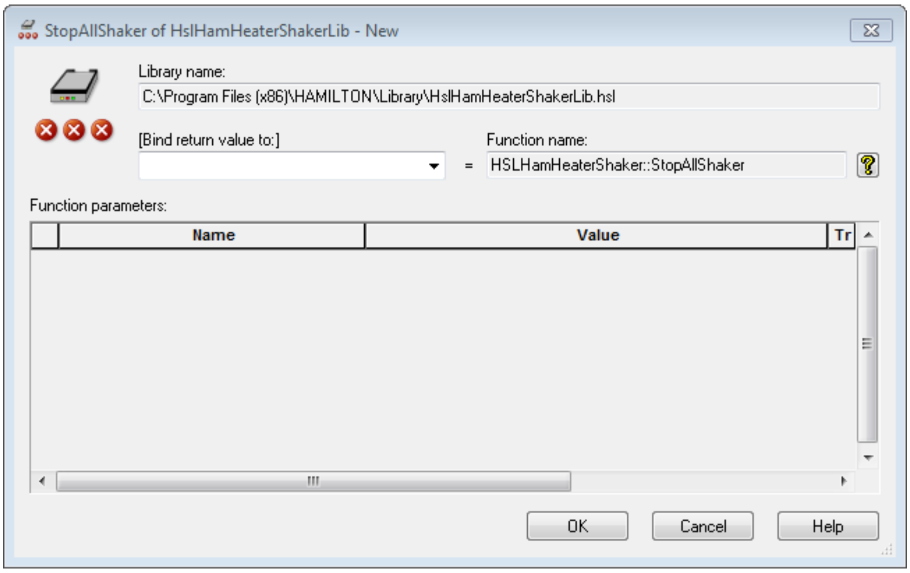
Step 10-12: The "**SetShakerParameter**" function can be used to change the shaking direction or acceleration of the Heater Shaker. Usually, the default settings can be used. The default settings are clockwise shaking and an acceleration of 1250. The shaking direction can be changed by entering "**1**" for counter-clockwise shaking. The value for the acceleration (shakingAccRamp) ranges from 630 to 125000. The default setting of 1250 matches up the acceleration from 0 to the maximum speed of 2500 rpm within two seconds.



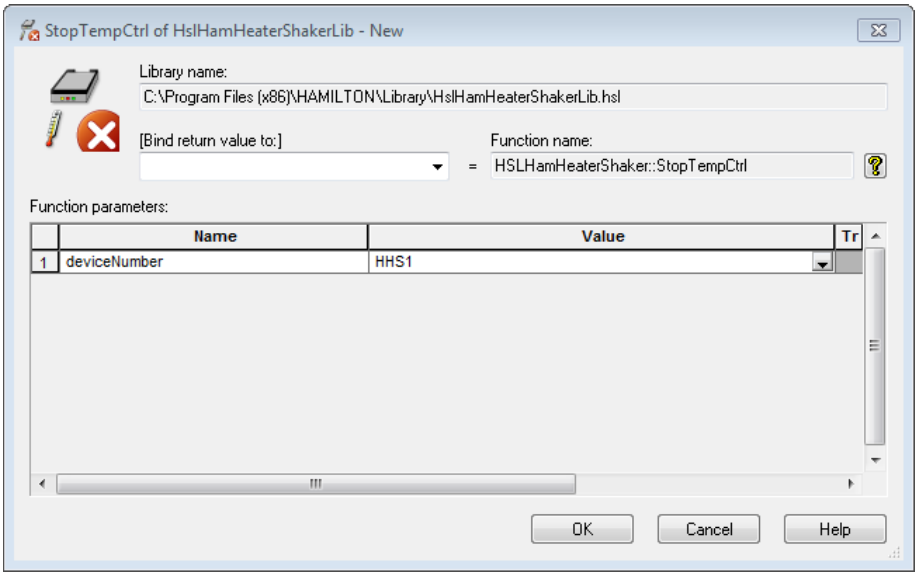
Step 13: All shakers can be started in parallel. Here, the shaking speed is set to 2000 rpm. This function will automatically close the plate lock on all shakers.



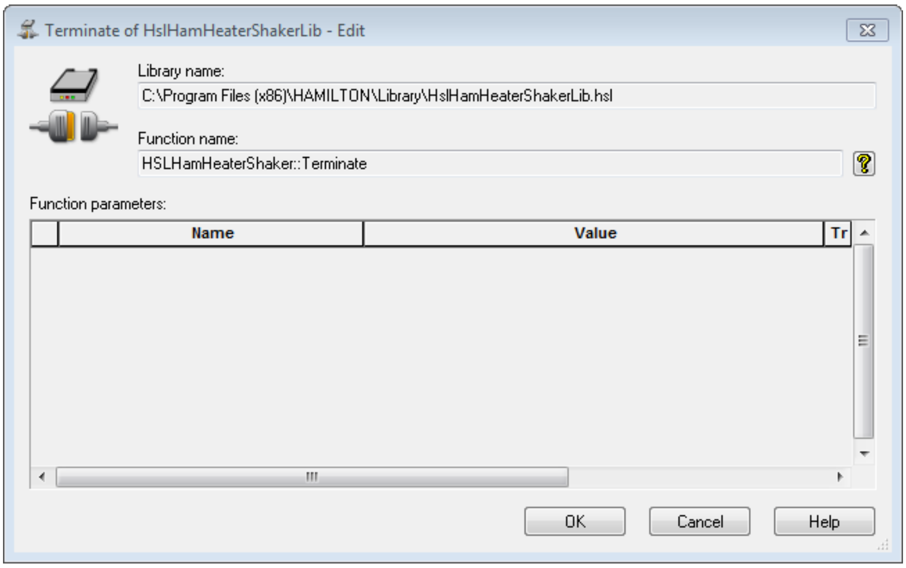
Step 16: To finish shaking of all shakers at the same time use the "**StopAllShaker**" Function. This function does not require any device numbers but will stop all initialized shakers and will automatically open the plate lock.



Step 17-19: The temperature control of the Heater Shaker can be terminated after its usage. If the temperature control is not stopped, heating will continue even after terminating the shaking process.



Step 20: The connections to the Heater Shakers are terminated. This step can be omitted at the end of a method, since the connections to all Heater Shakers are anyway automatically terminated at the end of a method or upon abort of a method.



18.1.4 Monitoring the Performance of the Hamilton Heater Shaker

For some applications, it might be desirable to monitor the performance of the Heater Shaker. The library offers the possibility to monitor the shaking speed and temperature during an application. The status of the Heater Shaker is continuously written to the trace file. The settings for the monitoring can be adjusted within the functions "**BeginMonitoring**" and "**SetTempParameter**".

Within the function "**BeginMonitoring**" the intervals can be defined, like how often the performance of the Heater Shaker will be checked and the deviation from the set shaking speed that will be tolerated. Choosing which action will be taken if monitoring reports an out-of-range measurement is also possible.

All settings regarding the temperature control must be made within the function "**SetTempParameter**", although the default settings are usually already sufficient.

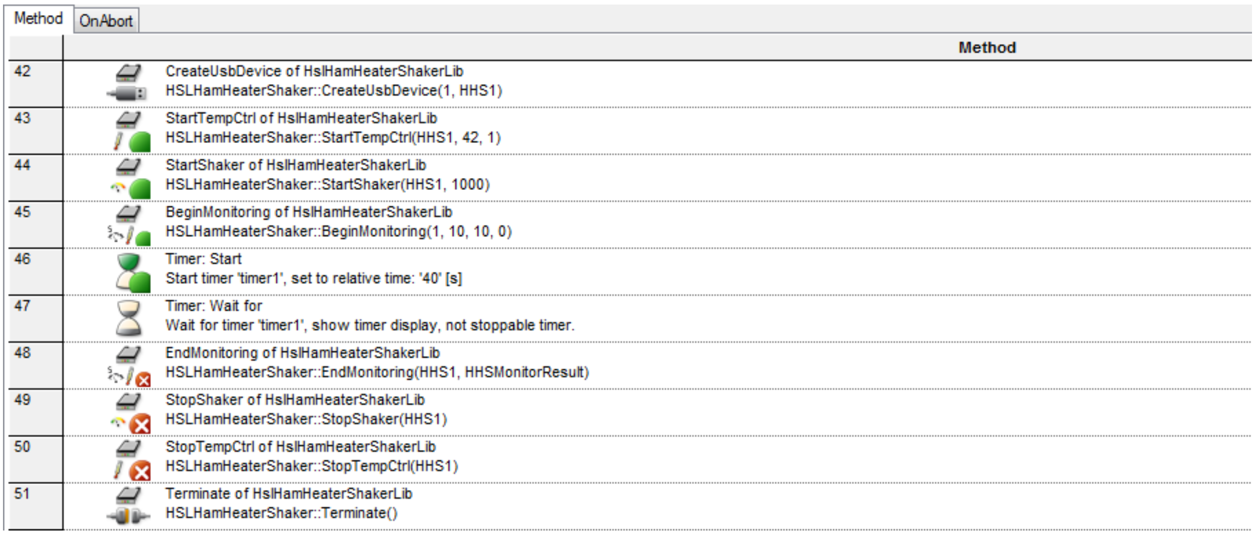
After monitoring, the return value from the function "**EndMonitoring**" can be examined. The function reports whether heating or shaking or both were out of range.

ATTENTION

*Placing a cold plate on the hot heater shaker will cool down the heater shaker so that the temperature might fall below the defined tolerated temperature range. In this case, the "**BeginMonitoring**" Function will return an error even if heating is working correctly.*

*To avoid this kind of error, the "**StartTempCtrl**" Function must be used immediately before the transport step. This results in heating the heater shaker again until the temperature is stable for 3 min. Monitoring is paused during the "**StartTempCtrl**" Step.*

The image below is an example of the monitoring function.



19 CR Needle Wash Station

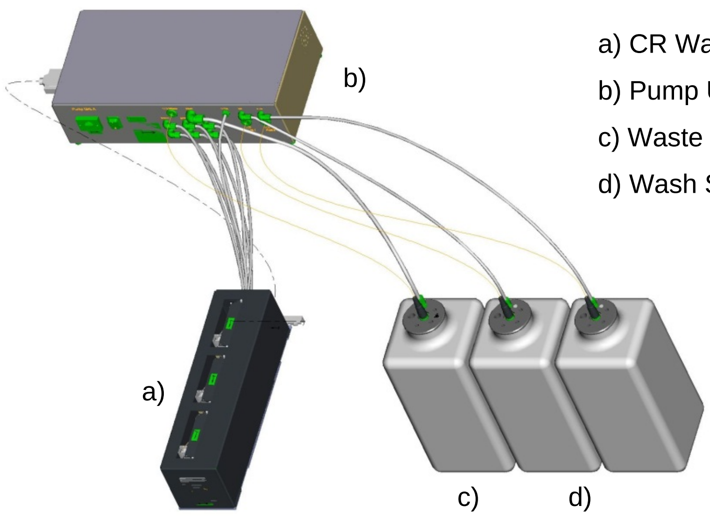
19.1 Needle Washing Using the CR Needle Wash Station

In combination with a Wash Station, re-usable steel needles can be used for pipetting with the spreadable 1000µl-pipetting channels of the Microlab instrument, instead of using the disposable tips. The result of washing needles depends on the wash setting.

NOTE

If carry-over is not acceptable for the application, use new disposable tips for each aspiration step instead of washed steel needles.

The picture shows a completely hooked-up CR needle Wash Station:



a) CR Washer Unit

b) Pump Unit A and Tubing

c) Waste Container (one)

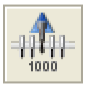
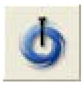
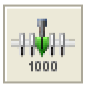
d) Wash Solution Containers (two)

The wash cycle of the CR needle wash station works in parallel to the pipetting steps. Liquid level sensors in the containers recognize whether the Wash Solution is used up or if the waste bottle is full. During run time, a message box is displayed giving the opportunity to refill/empty the containers.

19.2 Command Description

The following tables in this section give a brief overview of the available commands for the CR needle Wash Station.

19.2.1 Single Steps

Liquid Handling		
1000 µL-Pipetting Channel		
Command	**Icon**	**Action Performed**
1000µl Channel Tip Pick Up		Picks up a CO-RE tip or needle.
1000µl Channel Start Needle Wash		Starts a needle wash module.
1000µl Channel Wait For Needle Wash		Waits for the needle wash module to be ready.
1000µl Channel Tip Eject		Discards the tip into the tip waste or the needle into the wash station or rack.

NOTE

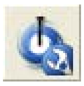
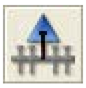
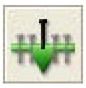
*It is recommended to use the "**SmartStep Needle Pickup**". If the "**1000µl-Channel Tip Pick Up**" Single Step is used, a "**1000µl Channel Wait for Needle Wash**" Step has to be programmed before the tip pick up step.*

With the Easy Step ASPIRATE, the needle pick up is also possible.

With the Easy Step DISPENSE, needle eject and start washing is also possible.

19.2.2 Smart Steps

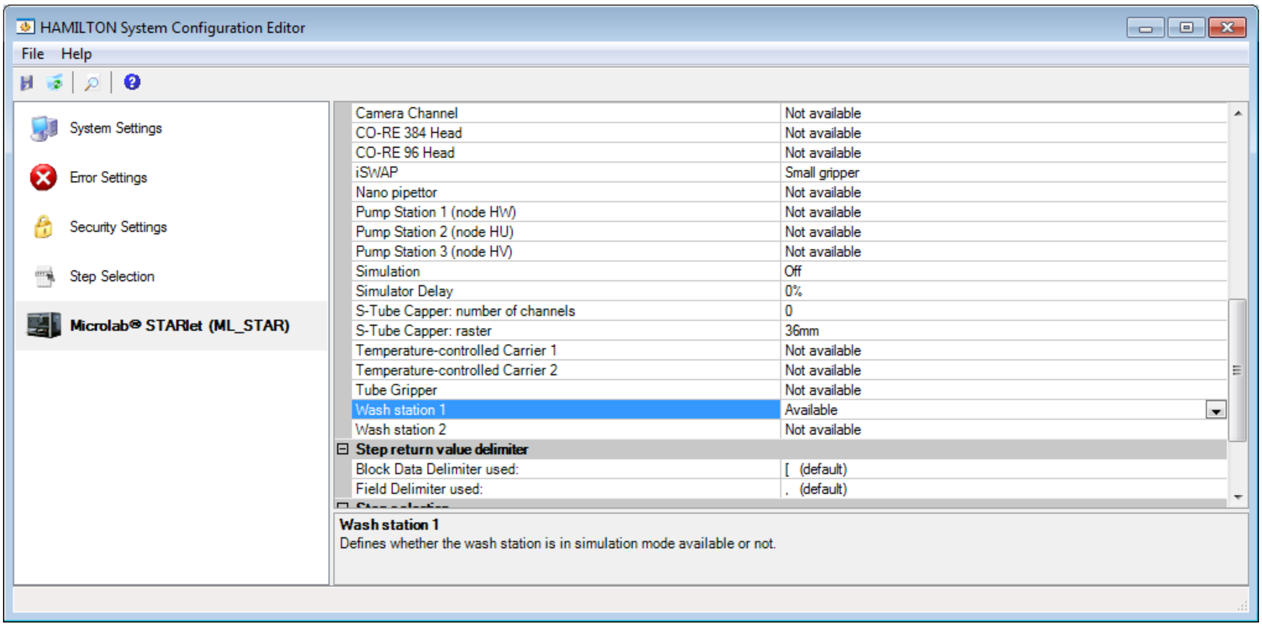
The needle wash step is embedded in the "**Needle Eject**" Command.

ML_STAR Smart Steps		
Command	**Icon**	**Action Performed**
1000µl Needle Wash Settings		Sets Wash Parameters.
1000µl Channel Needle Pick Up		Picks up a needle.
1000µl Channel Needle Eject		Ejects the needle for (optional) washing.

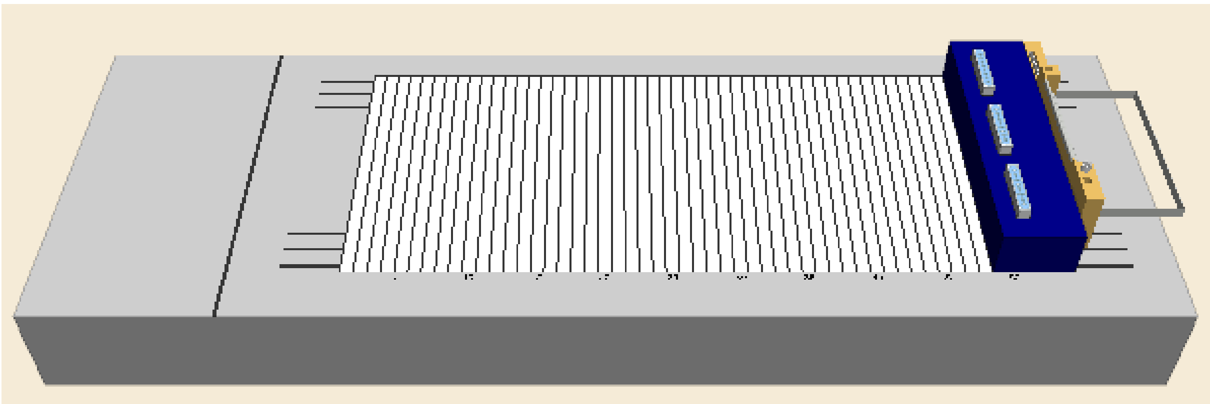
19.3 Programming the CR Needle Wash Station

Below is a simple example of a method showing how to program the CR Wash Station using the Smart Steps. This method will wash all three sets of needles in the Wash Station. This could be useful after the instrument was not used for a long time.

1. First, activate the Wash Station in the System Configuration Editor. Note that this entry will work globally for all methods. If the method does not use a Wash Station, it can also be switched off here.

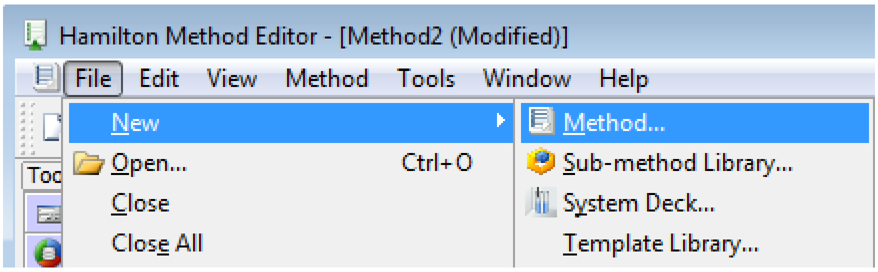


2. The method requires the following Deck Layout:

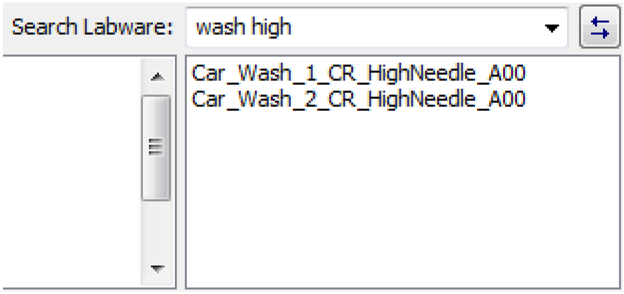


Creating the Deck Layout

1. Create a new method called DemoWashStation.med.



2. Switch to the "**Labware**" Tab and add the search string "**Wash high**" to the "**Search Labware**" Field. The system will filter out everything except the two possible wash stations. Select the "**Car_Wash_1_CR_HighNeedle_A00**" Carrier for the wash station and drag it onto the deck. Refer to the image below.

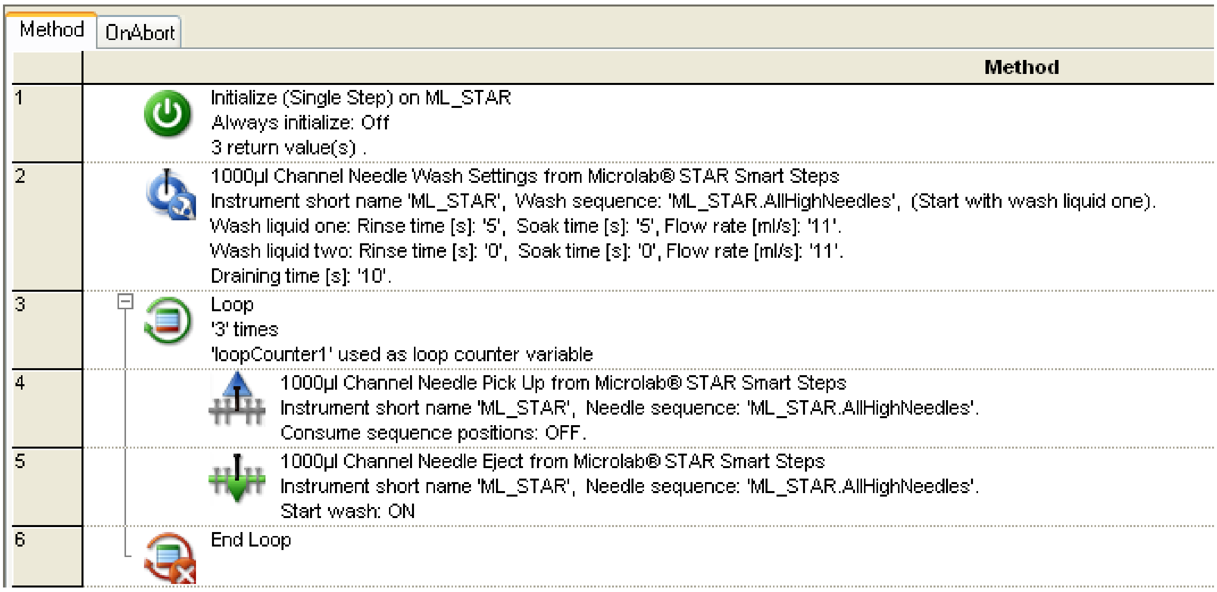


Creating the Sequences

Create one sequence over all three wash blocks and name it "**AllHighNeedles**".

Creating the Method

The resulting method should look like this:

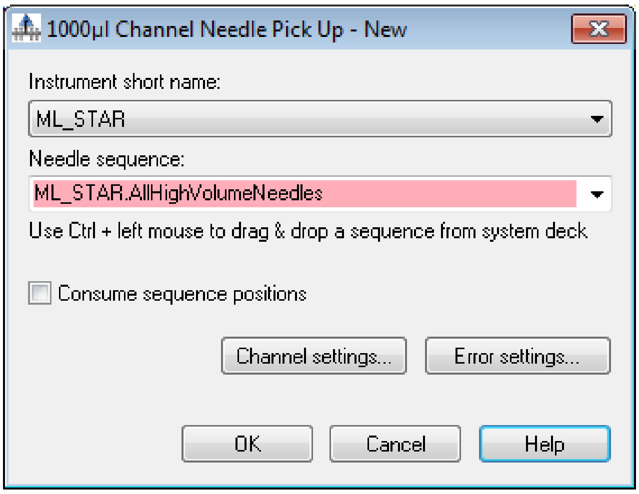


1. Define the wash settings using the “**Needle Wash Settings**” Smart Step (line 2).
2. This table shows default values of the wash parameters for 1000 µL, 300 µL and 10 µL-volume needles.

Needle	**Rinse time**	**Soak time**	**Flow rate**	**Draining time**
10 µL	5 seconds	5 seconds	16 ml/second	10 seconds
300 µL	5 seconds	5 seconds	15 ml/second	10 seconds
1000 µL	5 seconds	5 seconds	11 ml/second	10 seconds

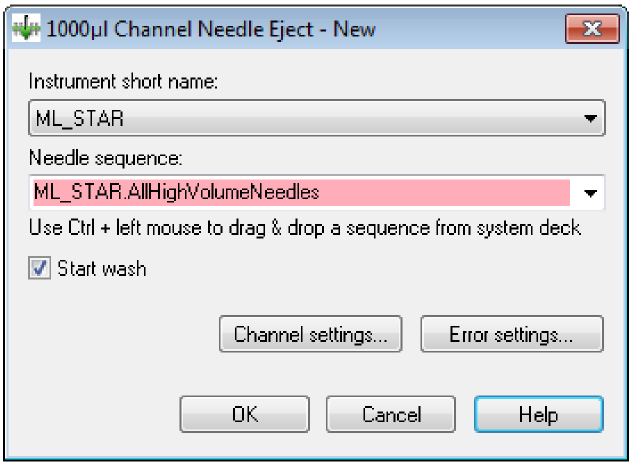
3. Select the wash sequence that shall be used for washing needles. Set all wash parameters and Click **[OK]**.
4. Create a loop of 3 iterations (or use the sequence “**AllHighNeedles**” as the limitation).

5. The next step would be to pick up the needles:

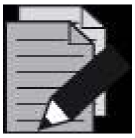


6. Please note that the sequence can be selected from the drop-down list or by clicking **[Ctrl]**. It can be “Drag-and-Dropped” directly from the Deck Layout.

7. The Needle Eject dialog box appears as shown:



8. Make sure that the "**Start wash**" Checkbox is ticked. Activating this box will close the wash modules lid right after ejecting and start the washing with the specified parameters.



NOTE

*It is recommended to use the "**Needle Pickup**" Smart Step. If the "**1000µl Channel Tip Pick Up**" Single Step is used, a "**1000µl Channel Wait for Needle Wash**" Step has to be programmed before the tip pick up step.*

With the Easy Step ASPIRATE, the needle pick up is also possible.

With the Easy Step DISPENSE, needle eject and start washing is also possible.

The error settings are similar to those of the "**Pipette**" Smart Step.

20 DC Needle Wash Station

In combination with a wash station, re-usable steel needles can be used for pipetting with the spreadable pipetting channels of the Microlab instrument, instead of using the disposable tips. The result of washing the needles depends on the wash setting.

NOTE

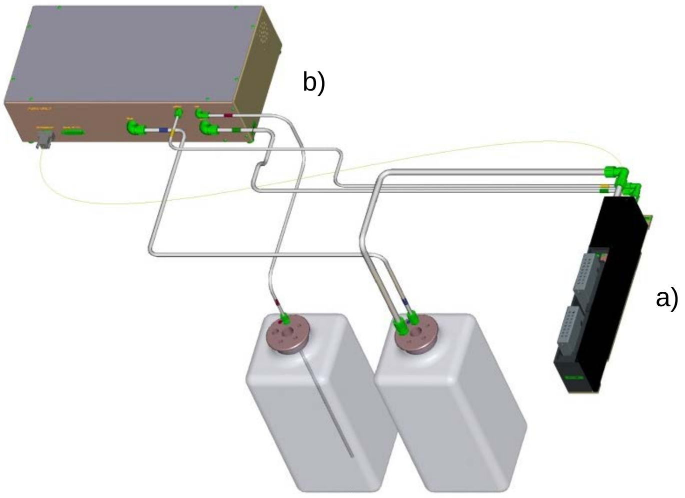
*It is recommended to use the "**Needle Pickup**" SmartStep. If the "**1000µl Channel Tip Pick Up**" Single Step is used, a "**1000µl Channel Wait for Needle Wash**" Step has to be programmed before the tip pick up step.*

With the Easy Step ASPIRATE, the needle pick up is also possible.

With the Easy Step DISPENSE, needle eject and start washing is also possible.

Sample transfers may be done using new disposable tips, while reagents, buffers, etc. may be distributed with needles.

The picture below shows a complete, hooked-up DC needle wash station:



a) DC Washer Unit

B) Pump Unit P and Tubing

C) Wash Solution Container

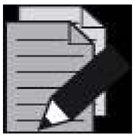
D) Waste Container

The DC needle Wash Station does not allow parallel washing and pipetting. Liquid level sensors inside the washer unit prevent flooding of the system. The washer unit recognizes if there is not enough liquid to fill the wash chamber.

The DC needle wash station consists of 32 positions for needles. Two different sets of needles can be used simultaneously. All types of needles (10 µL, 300 µL and 1000 µL) can be washed.

NOTE

There is no liquid level sensor in the waste container. Always empty the waste container in refilling the wash solution container.

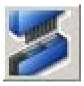
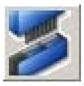


NOTE

The DC Needle Wash Station is no longer available.

20.1 Command Description

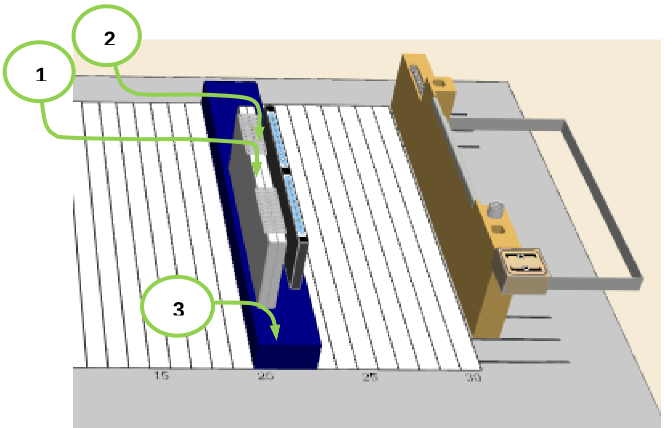
The commands for the DC needle wash station are functions of the library "**HSLMlStarDcWashstationLib**".

HSLMlStarDcWashstationLib:		
Command	**Icon**	**Action Performed**
Wash Settings		Sets wash parameters.
Needle Wash		Starts the wash step.
Needle Wash2		Starts the wash step.
Empty Fill Wash Chamber		Empties or refills the wash chamber.

20.2 Programming the DC Needle Wash Station

To create a method using needle washing with the DC needle Wash Station, follow these steps:

1. Create a new method called DC_WashstationDemo.
2. Add the template for the DC needle Wash Station: ***Car_DC_WashStation_CR_Needle_A01*** to the layout.

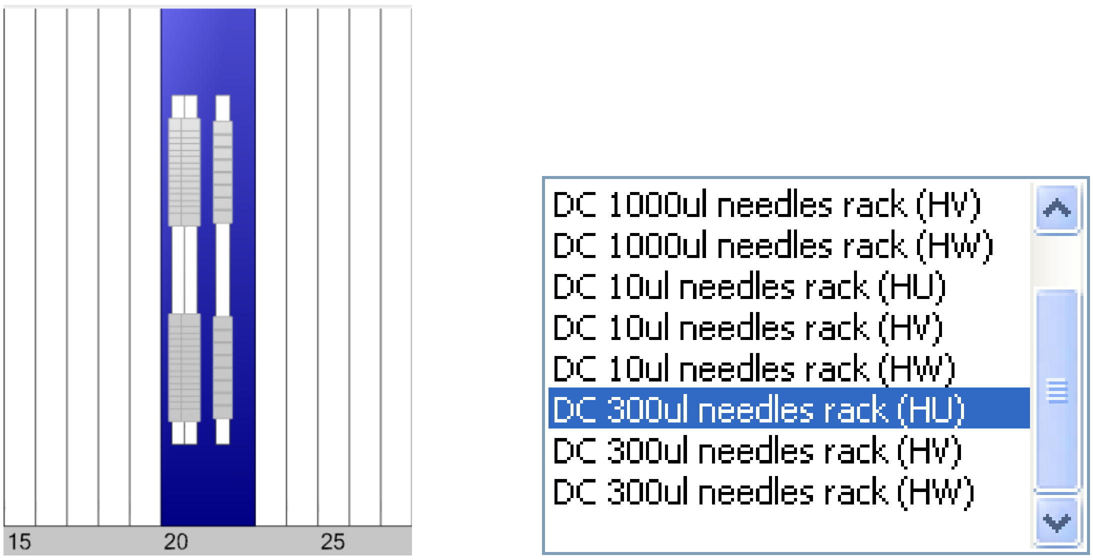


The following sequences belong to the template of the DC needle Wash Station:

Pos	Sequence Name	Use
1	waste_dc_washstation_0001	Dispensing the rest volume before the needle wash (empty tip).
2	waste_dc_washstation_0002	Dispensing the rest volume before the needle wash (empty tip).
3	washchamber_dc_washstation_0001	Washing the needles.

3. Add the needle sequence (according to the used needles and the Node ID setting of the pump unit and washer unit) to the template of the DC needle Wash Station. The needle sequence corresponds to the position (for picking up / releasing) of the needles.

   e.g. ***DC 300µl needles rack (HU)***



HU, HW and HV are the node (ID-) names of the corresponding pump unit

4. Switch to the method and add the HSLMlStarDcWashstationLib library.
5. Program a similar method like for the CR Wash Station. Use the help file of the HSL Library.

6. Recommended wash settings and typical carry-over:

300 µL needle **(Asp./disp 100 µL fluorescein solution 2g/100ml PBS; wash solution deionized water)**		**Result**	
Wash volume: Wash cycles: Mix cycles: Soak time: Immersion depth:	300 µL 2 3 0 sec 5 mm	Duration: Water consumption: Carryover:	45 sec 125 ml > 4.0 x 10^{-5}
Wash volume: Wash cycles: Mix cycles: Soak time: Immersion depth:	300 µL 3 3 0 sec 5 mm	Duration: Water consumption: Carryover:	70 sec 250 ml > 3.0 x 10^{-6}

21 Wash Station 96

The Wash Station 96 is an optional device for washing 96 disposable tips in parallel. The tips are washed both outside and inside at the same time: on the outside in the wash chamber of the wash station 96 and on the inside by aspiration/dispense cycles with the CO-RE 96 Probe Head. The result of washing the tips depends on the wash setting.

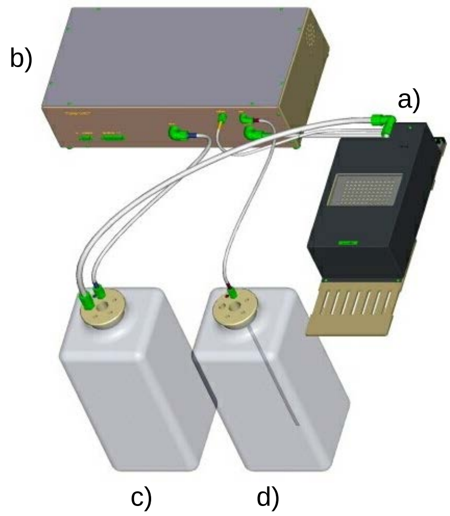
NOTE

If carry-over is not acceptable for the application, use new disposable tips for each aspiration step.

The Wash Station 96 is no longer available. It was replaced by the wash station 96/384.

Sample transfers may be done using new disposable tips, while reagents, buffers, etc. may be distributed with washed tips.

The picture below shows a complete, fully connected Wash Station 96:



a) Washer unit 96

b) Pump unit P and tubing

c) Waste container

d) Wash solution container

Liquid level sensors inside the washer unit prevent flooding of the system. On the other hand, the washer unit recognizes if there is not enough liquid to fill the wash chamber.

NOTE

There is no liquid level sensor in the waste container. In case of refilling the wash solution container, always empty the waste container.

When re-using washed tips, pipetting precision may decrease by a factor of 3.

21.1 Command Description

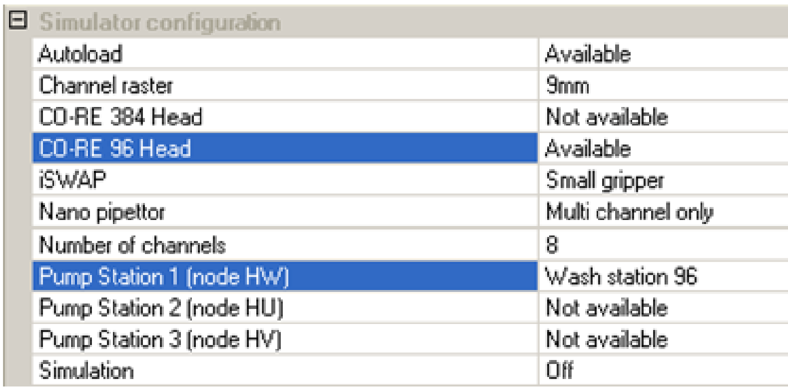
The table below gives a brief overview of the available commands for the Wash Station 96. Another possibility to wash the tips is given by the "**CO-RE 96 Head Dispense**" Basic Step Command. In the Basic dispense step, the wash function via the "**Wash tips after dispense**" Parameter can be enabled. The wash parameter can be set through the [**Customize…]** Button.

Liquid Handling		
CO-RE 96 Probe Head		
Command	**Icon**	**Action Performed**
CO-RE 96 Head Wash		Wash tips using the default settings.
CO-RE 96 Head Empty Washer		Empty washer.

21.2 Programming the Wash Station 96

The following is a simple example on how to program the Wash Station 96 using Single Steps.

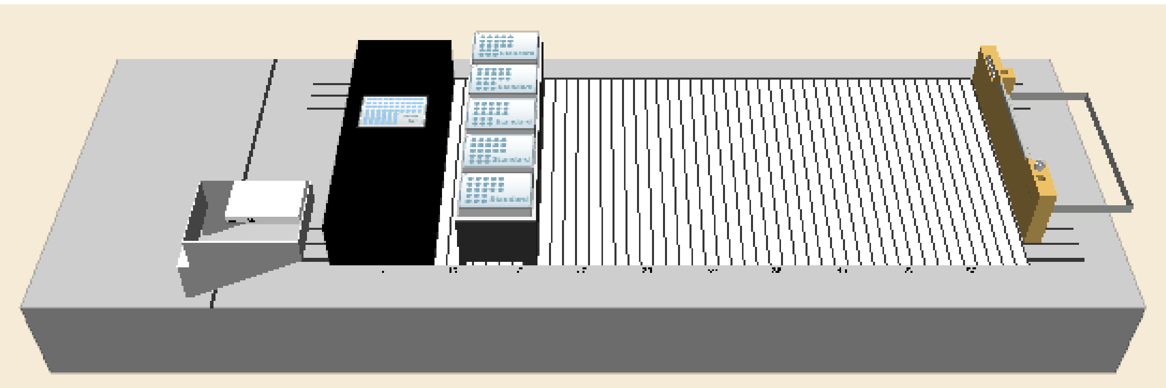
Please note that the CO-RE 96 Probe Head and one of the Pump Stations in the System have to be activated in the Configuration Editor.



Simulator configuration	
Autoload	Available
Channel raster	9mm
CO-RE 384 Head	Not available
CO-RE 96 Head	Available
iSWAP	Small gripper
Nano pipettor	Multi channel only
Number of channels	8
Pump Station 1 (node HW)	Wash station 96
Pump Station 2 (node HU)	Not available
Pump Station 3 (node HV)	Not available
Simulation	Off

Creating the Deck Layout

The method requires the following Deck Layout:

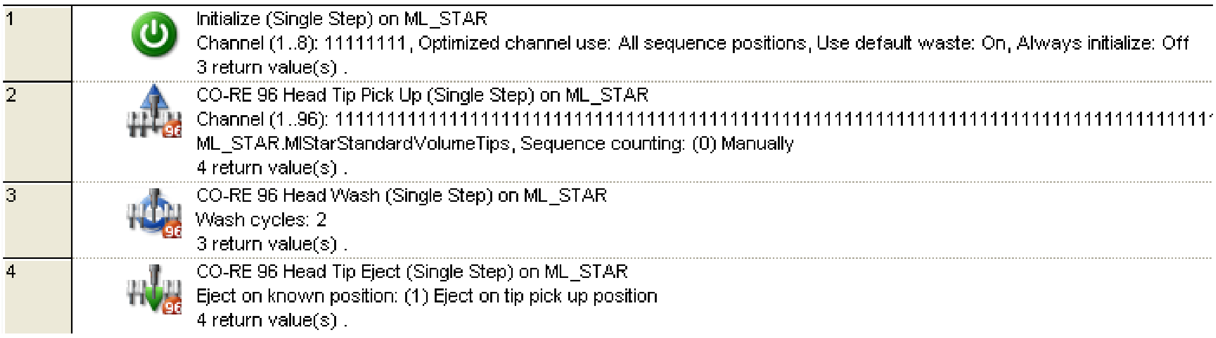


Add a Wash Station, slide waste and a Standard Tip Carrier to the deck. Use the "**Search Labware**" Field to add the following labware:

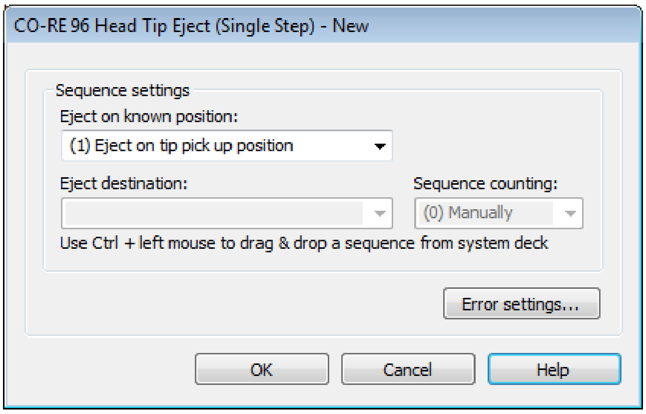
- "**CORE_HU_96WashStation_A00**"
- "**Core_96SlideWaste**" from "**Microlab STAR Wastes**"
- "**TIP_CAR_480_ST_A00**"

Creating the Method

The resulting method should look like the image shown below.



1. Pick up tips from the Tip Carrier. Set "**Manually**" for sequence counting to eject the tips to the pickup position (Step 2).
2. In line 3, use the "**CO-RE 96 Head Wash**" Wash Step without modifying the default values. These values guarantee and are the basis for good wash results.
3. Eject the tips. Use "**Eject on tip pick up position**" to eject the tips back into the Tip Carrier.

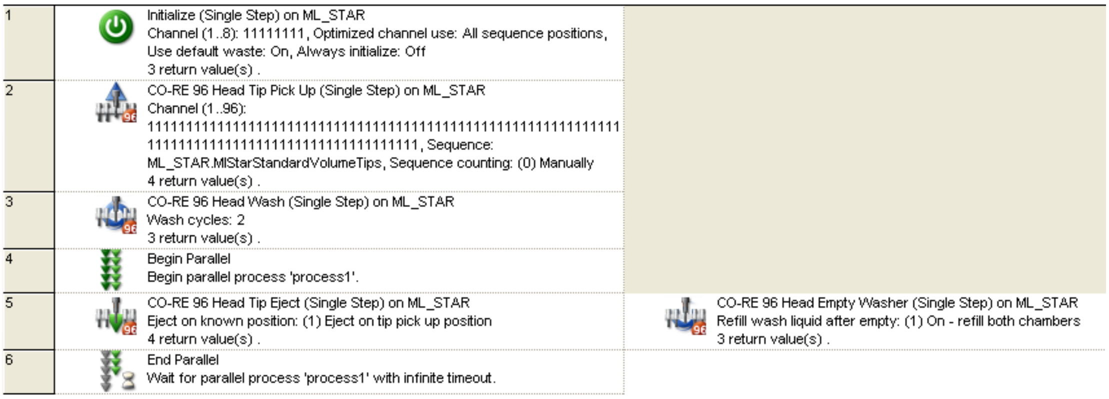


NOTE

*The method can be accelerated when programming the last "**Refill wash chamber**" Command parallel to the succeeding method steps.*

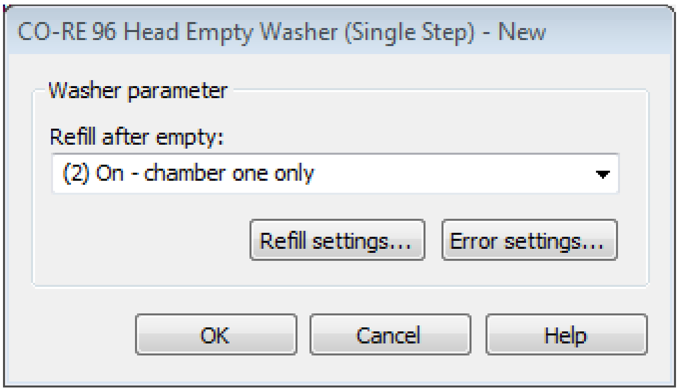
Creating a Modified Method

The method above can be modified to speed up the wash step. The last refilling command for the wash chamber has to be programmed in parallel to the tip eject step. The resulting modified method should look like the image shown below.



Use the “**Begin Parallel**” and “**End Parallel**” Steps to do this.

In the CO-RE 96 Head empty washer step, set the washer parameters to “**(2) On – chamber one only**” to have the maximum speed up.



22 Wash Station 96/384

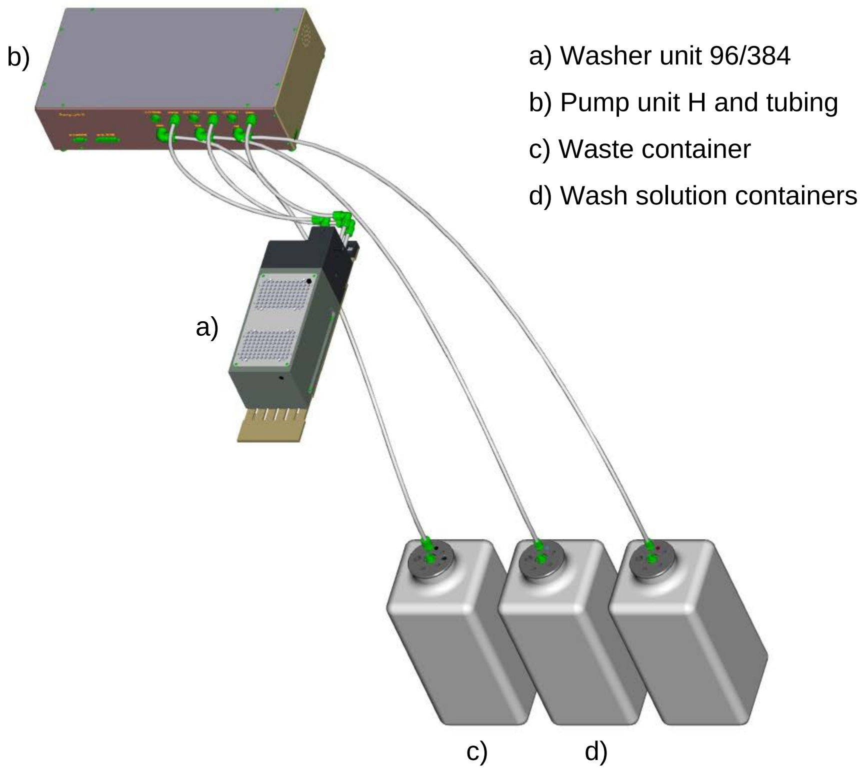
The Wash Station 96/384 is an optional device for washing 96/384 disposable tips in parallel. The tips are washed both outside and inside at the same time: on the outside in the wash chambers of the Wash Station and on the inside by aspiration/dispense cycles using the CO-RE 96 Probe Head / TADM or the CO-RE 384 Probe Head. The result of washing the tips depends on the wash setting.

NOTE

If carry-over is not acceptable for the application, use new disposable tips for each aspiration step.

Sample transfers can be done using new disposable tips, while reagents, buffers, etc. can be distributed with washed tips.

The picture below shows a complete, hooked-up Wash Station 96/384:



Liquid level sensors inside the washer unit prevent flooding of the system. The washer unit recognizes if there is not enough liquid to fill the wash chamber.

NOTE

Always empty the waste container when refilling the wash solution container. When re-using washed tips, pipetting precision may increase by a factor of 3.

22.1 Command Description

The table below gives a brief overview of the available commands for the wash station 96/384. Another possibility to wash the tips is given by the basic step commands "**CO-RE 96 Head Dispense**" / "**CO-RE 384 Head Dispense**". In the basic dispense step, the wash function can be enabled via the "**Wash tips after dispense**" Parameter. The wash parameter can be set through the **[Customize…]** Button.

Liquid Handling		
CO-RE 96 Probe Head		
Command	**Icon**	**Action Performed**
CO-RE 96 Head Wash		Wash tips.
CO-RE 96 Head Empty Washer		Empty washer.

Liquid Handling		
CO-RE 384 Probe Head		
Command	**Icon**	**Action Performed**
CO-RE 384 Head Wash		Wash tips.
CO-RE 384 Head Empty Washer		Empty washer.

ATTENTION

The CO-RE 384 Probe Head performs a blow out after the wash steps. The software automatically uses the tip picking up position (tip rack). Therefore, do not reload the tip rack of the pick-up position if performing any wash steps.

22.2 Programming the Wash Station 96/384

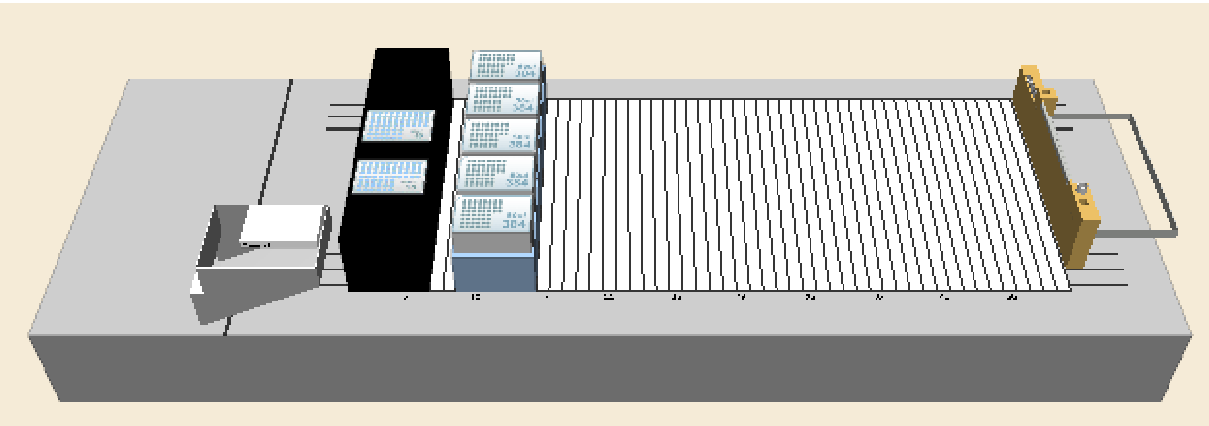
The following is a simple example on how to program the Wash Station 96/384 using the Single Steps.

Activate the CO-RE 384 Probe Head and the Pump Station 2 in the Configuration Editor:

Simulator configuration	
Autoload	Available
Channel raster	9mm
CO-RE 384 Head	Available
CO-RE 96 Head	Not available
iSWAP	Small gripper
Nano pipettor	Multi channel only
Number of channels	8
Pump Station 1 (node HW)	Not available
Pump Station 2 (node HU)	Dual chamber wash station 96/384
Pump Station 3 (node HV)	Not available
Simulation	Off

Creating the Deck Layout

The method requires the following Deck Layout:



Use the "**Search Labware**" Field to add the following labware to the deck:

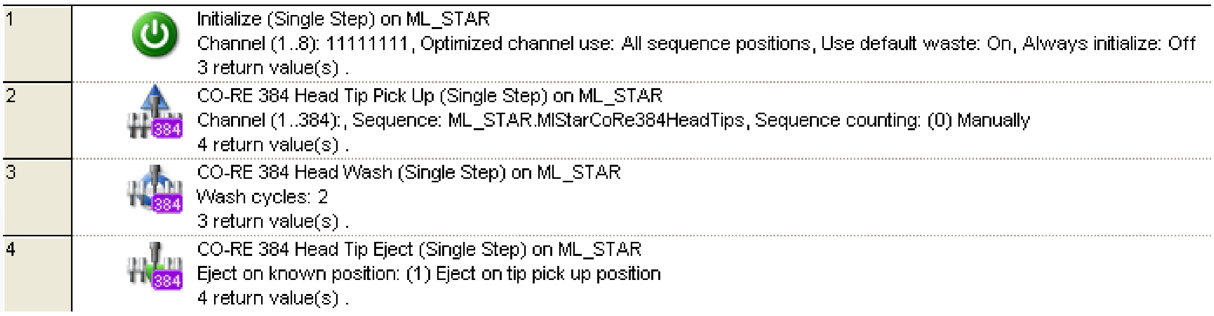
- "**CORE384DualWashStation_HU_A00**"
- "**Core 384 Slide Waste**"
- "**TIP384_CAR_1920_50ul_A00**"

Creating the Sequences

In this example, the default sequences are being used.

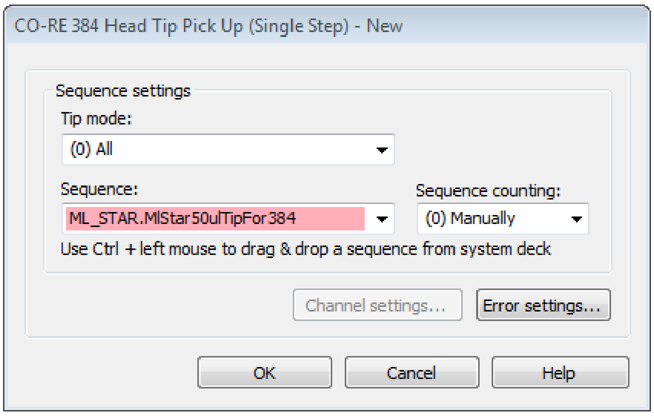
Creating the Method

The resulting method should look like the image below:

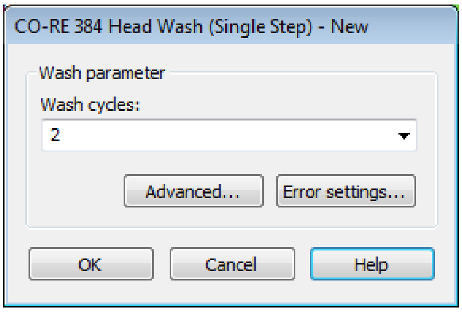


The step-in line 1 initializes the instrument.

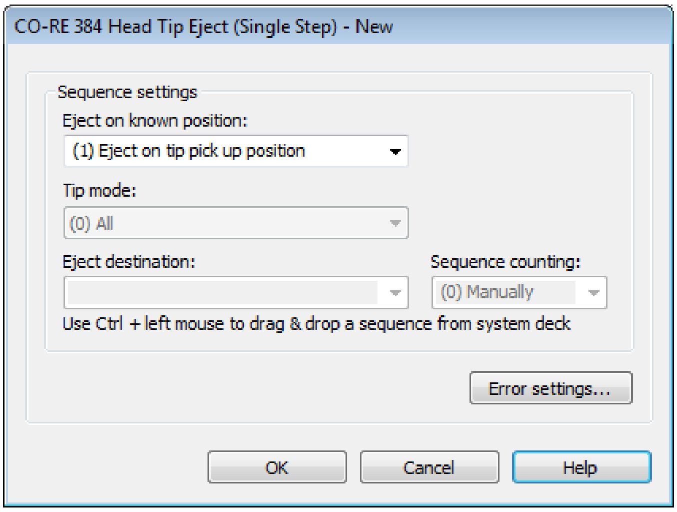
In step 2, the tips are picked up from the tip carrier (sequence counting = “**Manually**”).



1. Use the “**CO-RE 384 Head Wash**” Wash Step without modifying the default values. These values guarantee and are the basis for good wash results.



2. Eject the tips. Use “**Eject on tip pick up position**” to eject the tips back into the Tip Carrier.



NOTE

Always empty the waste container when refilling the wash solution container. When re-using washed tips, pipetting precision may increase by a factor of 3.

23 Temperature Controlled Carrier (TCC)

The temperature-controlled carrier (TCC) is a device for heating and cooling of micro plates. It has four positions for micro plates which are all of the same temperature. The TCC can heat micro plates up to 60°C or cool them down to 22°C below ambient temperature.

A maximum of two TCCs can be placed on one Microlab instrument and controlled by the VENUS Software.

Activate the Temperature Controlled Carrier in the System Configuration Editor (Microlab STAR Tab).

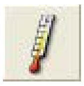
Simulator configuration	
Autoload	Available
Channel raster	9mm
CO-RE 384 Head	Not available
CO-RE 96 Head	Not available
iSWAP	Small gripper
Nano pipettor	Multi channel only
Number of channels	8
Pump Station 1 (node HW)	Not available
Pump Station 2 (node HU)	Not available
Pump Station 3 (node HV)	Not available
Simulation	Off
Simulator Delay	0%
Temperature-controlled Carrier 1	Available
Temperature-controlled Carrier 2	Not available
Wash station 1	Available
Wash station 2	Not available

The method is also using the sample tracker to store the pipetting data to a file. It can be activated in the System Configuration Editor (System Tab) as well.

Sample tracking settings	
Flag secondary Vessels	Disabled
Sample Tracking	On
Vector Database	On
Vector Database connection	Database 'HamiltonVectorDB' on server

23.1 Command Description

The table below provides a brief overview of the available commands for the TCC.

ML_STAR		
Preparation		
Command	**Icon**	**Action Performed**
Set Carrier Temperature		Sets the temperature of a TCC.
Get Carrier Temperature		Retrieves the temperature of a TCC.

Typical times (at 40% rel. humidity) to heat and cool a Microplate are ($T_{ambient} \approx 20°C$):

$T_{ambient}$ to 60°C:	20 min
60 °C to $T_{ambient}$:	20 min
$T_{ambient}$ to 4°C:	15 min
4°C to 60°C:	25 min

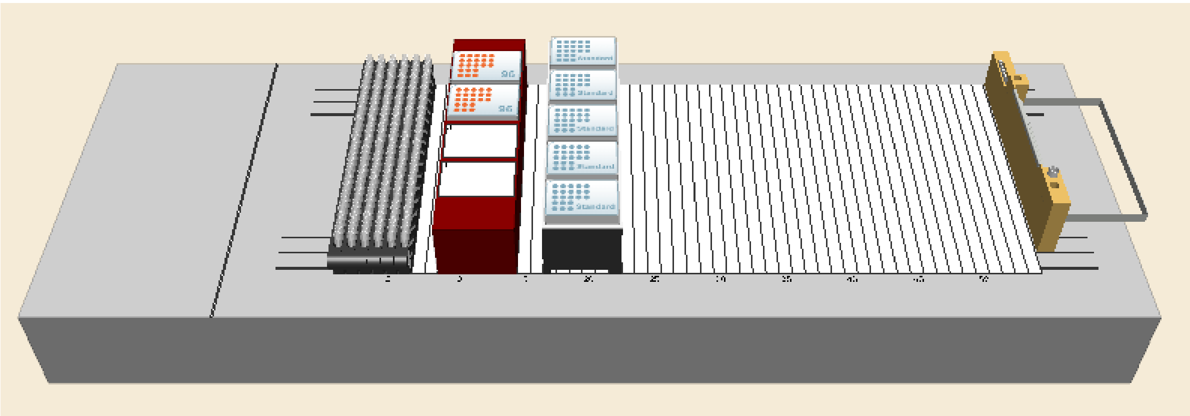
23.2 Programming the TCC / Sample Tracking

The following example makes use of the TCC and shows the additional functionality of the Microlab instrument user software. It can be found under the name "**Example**".

Creating the Deck Layout

Create the Deck Layout holding the following:

- Tip carrier (TIP_CAR_480_ST_A00).
- 6 tube carriers (SMP_CAR_32_12x75_A00) for 32 sample tubes each.
- Temperature-controlled Carrier (Car_TCC_1).
- Two standard Nunc plates (Nun_96_Fl_L) on the TCC.



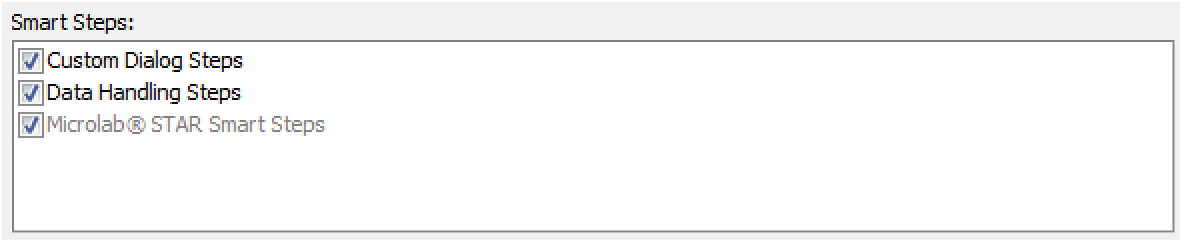
23.3 Creating the Sequences

Under the "**Sequences**" Tab, select the "**Channels**" Stamp Tool and create the following:

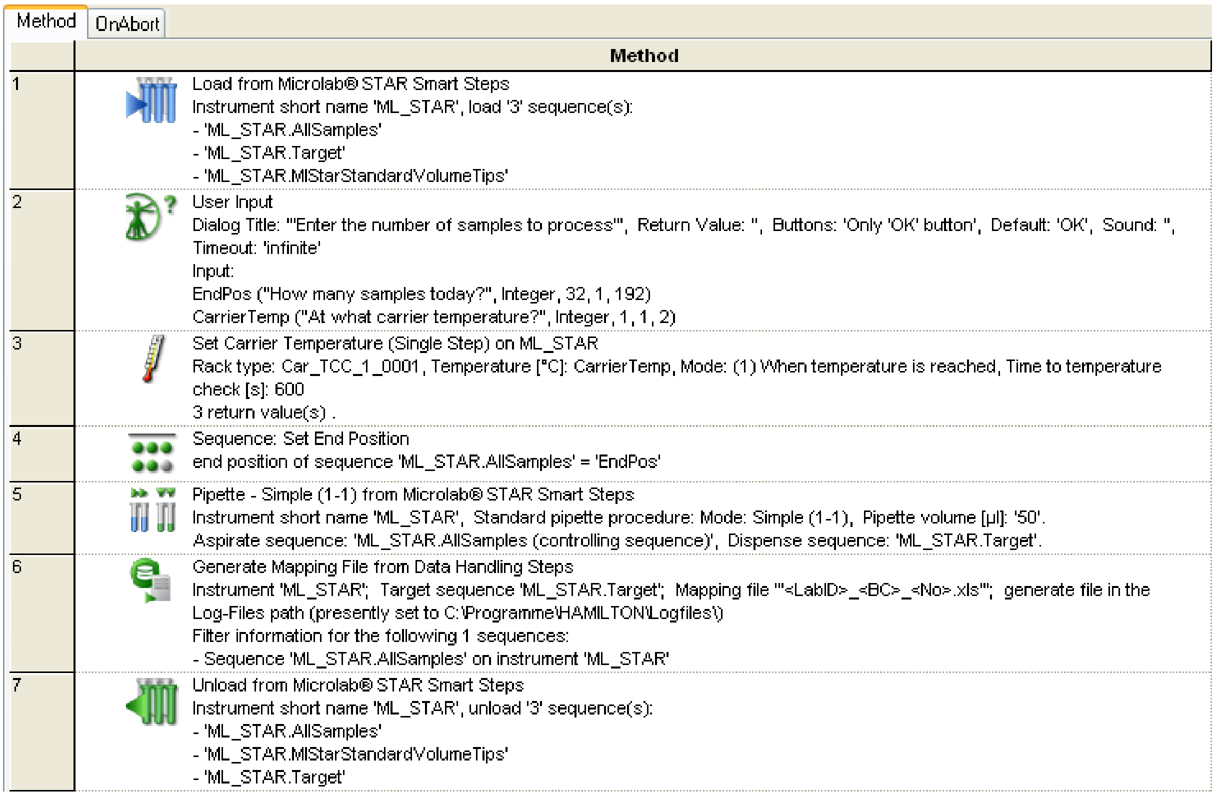
- A sequence over all six tube carriers named "**AllSamples**".
- A sequence over the two plates named 'Target'.

Creating the Method

In order to use the "**Generate Mapping File**" Step, which will create an output file with source / target / volume information, the "**Data Handling Steps**" Option has to be activated in "**Method → Instruments and Smart Steps**"



The complete method should look like the image presented below, following the first "**Initialize**" Step.

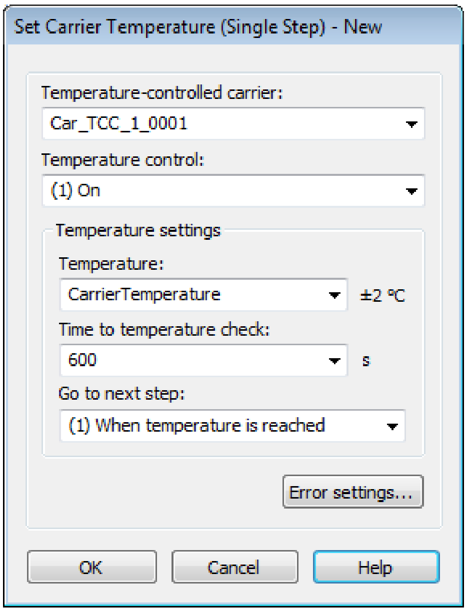


Method Analysis

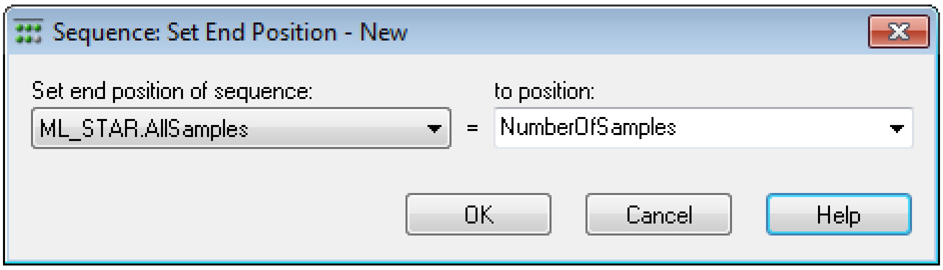
A look on how the method is constructed.

1. First, all carriers are loaded using the "**Load**" Smart Step:
2. A user input is used in line 2 to get both the number of samples to process and the temperature of the TCC to set. Do not forget to enclose the text in quotation marks; otherwise, the system will interpret it as a new variable.

3. Step 3 will set the temperature of the TCC.

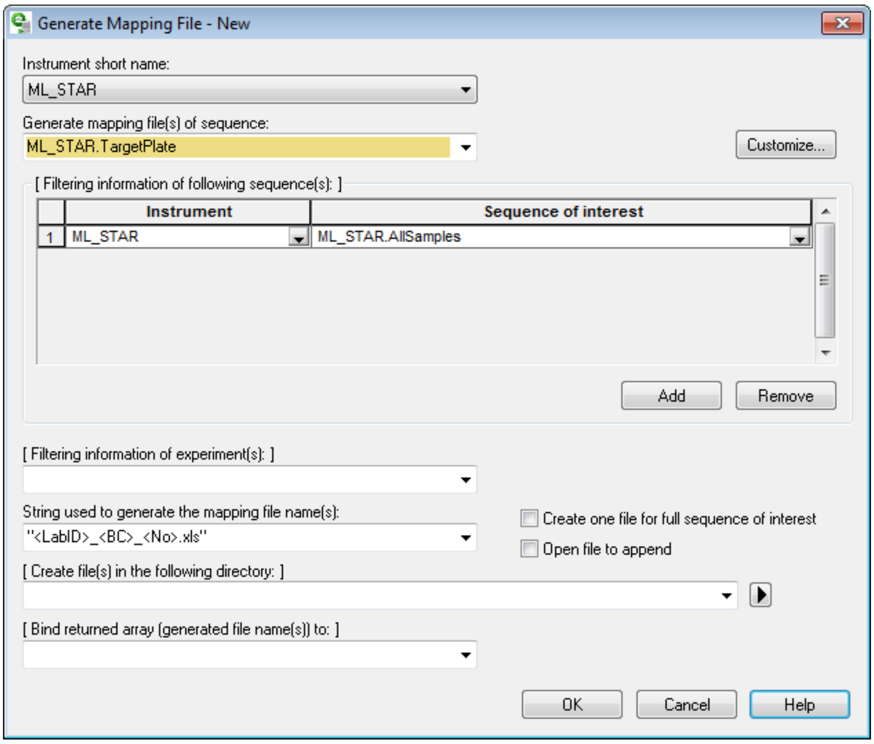


4. The carrier temperature is not set in absolute. It is held in the variable depending on the user’s input.
5. Note that this step will continue when the temperature is reached. This means that it will wait for 600 sec and then check if the temperature was reached. For detailed information, refer to the steps in the help file.
6. In step 4, the end position of the AllSamples sequence is set to the number the user has specified. Since there is a limit in the user’s input, the number cannot be set wrong.

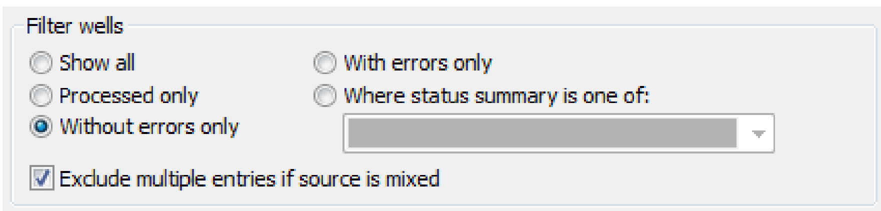


7. The “**Pipette Simple**” Step will bring the liquid from the tubes into the plate. Depending on the number of samples, either only one plate or both plates are used.
8. In step 6 a “**Report Mapping File**” is generated. In this file, all the liquid transfers of selectable sequences are reported. Refer to Section 11.9.2 Generating a Mapping File for more information.
9. Enter a sequence in the “**Generate mapping file(s) of sequence**” Field. The mapping file will contain all information about the liquid that was pipetted into the sequence. Using “**Sequences of interest**”, it is possible to select which source sequences shall be taken into account.
10. In the “**Sequences of interest**”, selecting sequences where the liquid comes from is specified. In this case, the samples are coming from the AllSample sequences. If there was a buffer or reagent sequence as well, insert those sequences too. This is to see where all liquid comes from.

11. The filename is created by a constant string, but it is more useful to use information contained in the database by place holders. The default file name is built using the labware ID, the barcode and an auto-incrementing number.
12. The path where to store the mapping file(s) has to be specified. If left blank like in the example, the data will be stored to the logfiles directory.
13. If a customized path is entered, be sure the path exists before the mapping file has to be generated.



14. The output file will be filtered to contain only the error free processed samples.
15. To do so, click the **[Customize…]** Button.



16. To filter and keep the errorless ones only, select the **[Without errors only]** Radio Button.
17. In the Sorting field, specify the sorting direction to "**Sort records by column**".

18. As seen below, it is possible to customize the columns. The standard and constant columns are displayed on the left side; either the CAT Heater Shaker or TELESHAKE. Renaming the column name on the right side can be done.

Sorting

- (•) Sort records by column (A1, B1, C1...)
- ( ) Sort records by row (A1, A2, A3...)

		Columns to be exported	Column name
1	☑	Record ID	"RecordId"
2	☐	Target Rack Barcode	"TRackBC"
3	☑	Target Labware ID	"TLabwareId"
4	☑	Target Position ID	"TPositionId"
5	☐	Target Position Barcode	"TPositionBC"
6	☑	Target Status Summary	"TStatusSummary"
7	☑	Target Status Summary Description	"TSumStateDescription"
8	☑	Target Volume	"TVolume"
9	☑	Source Rack Barcode	"SRackBC"
10	☐	Source Labware ID	"SLabwareId"
11	☐	Source Position ID	"SPositionId"
12	☐	Source Position Barcode	"SPositionBC"
13	☑	Action Timestamp	"ActionDateTime"
14	☑	User name	"UserName"

19. The resulting file is shown below. If customized column names are specified, all columns are marked with a "**T**" refers to the Target Sequence while all columns beginning with "**S**" refers to the Source Sequence.

	A	B	C	D	E	F	G	H	I
1	**RecordIc**	**TLabwareId**	**TPositionId**	**TStatusSummary**	**TSumStateDescription**	**TVolume**	**SRackBC**	**ActionDateTime**	**UserName**
2	1	Nun_96_Fl_Lb_0001	A1	0	Correct pipetting	150	CarrierBC	2008-03-30 11:42:20	chdanuser
3	2	Nun_96_Fl_Lb_0001	B1	0	Correct pipetting	150	CarrierBC	2008-03-30 11:42:20	chdanuser
4	3	Nun_96_Fl_Lb_0001	C1	0	Correct pipetting	150	CarrierBC	2008-03-30 11:42:20	chdanuser
5	4	Nun_96_Fl_Lb_0001	D1	0	Correct pipetting	150	CarrierBC	2008-03-30 11:42:20	chdanuser
6	5	Nun_96_Fl_Lb_0001	E1	0	Correct pipetting	150	CarrierBC	2008-03-30 11:42:20	chdanuser
7	6	Nun_96_Fl_Lb_0001	F1	0	Correct pipetting	150	CarrierBC	2008-03-30 11:42:20	chdanuser
8	7	Nun_96_Fl_Lb_0001	G1	0	Correct pipetting	150	CarrierBC	2008-03-30 11:42:20	chdanuser
9	8	Nun_96_Fl_Lb_0001	H1	0	Correct pipetting	150	CarrierBC	2008-03-30 11:42:20	chdanuser
10	9	Nun_96_Fl_Lb_0001	A2	0	Correct pipetting	150	CarrierBC	2008-03-30 11:42:24	chdanuser

24 Basic Vacuum System (BVS) / Crystal Vacuum System (CVS)

The CVS (Crystal Vacuum System) consists of a seven-track-wide carrier with a vacuum manifold, a park position for the manifold top and two plate positions. The rear micro plate position can be replaced by a Hamilton Heater Shaker.

The vacuum is generated with a pump, for example from Vacuubrand, which can be controlled by the VENUS Software.

The manifold top can be handled either by the iSWAP or the CO-RE Gripper. If the iSWAP are used for these movements, four tracks next to the BVS / CVS carrier must be empty, usually on the right side.

For more details about the CVS, refer to the corresponding Microlab instrument Operator's Manual.

NOTE

BVS and CVS are similar and need the same programming steps but different labware. If it is not certain which labware is at hand, teach the position inside the active position to avoid step losses during plate transports.

24.1 Command Description

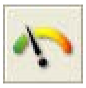
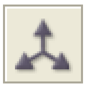
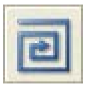
The commands for the BVS / CVS are functions of the additional library "**HSLVacuuBrandPump.hsl**". To make the library functions available, install the library through "**Tools → Hamilton Support Software...**".

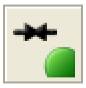
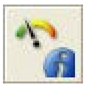
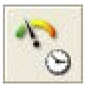
The following commands are integrated in the "**HSLVacuuBrandPump**" Library:

HSLVacuuBrandPump
- HSLStarBVSLib::BVSAbort
- HSLStarBVSLib::BVSGetAmbientPressure
- HSLStarBVSLib::BVSGetSimulationMode
- HSLStarBVSLib::BVSInitialize
- HSLStarBVSLib::BVSSetSimulationMode
- HSLStarBVSLib::BVSTerminate
- HSLStarBVSLib::BVSTrack
- HSLStarBVSLib::BVSVacuum
- HSLStarBVSLib::BVSVacuumTrack
- HSLVacuuBrandPump::Initialize
- HSLVacuuBrandPump::OpenAirAdmittanceVa
- HSLVacuuBrandPump::ReqActualPressure
- HSLVacuuBrandPump::StartPressureControl
- HSLVacuuBrandPump::StopPumpImmediatly
- HSLVacuuBrandPump::Terminate
- HSLVacuuBrandPump::WaitForPumpStopped

The commands control only the VacuuBrand vacuum pump of the BVS / CVS - the vacuum pump itself and the air admittance valve. The controlling of the vacuum system runs in the background, independently from the method. This means that there is no need to create a parallel task. The only thing to do is to set the transport steps to and from the BVS / CVS and the transport of the manifold.

There are two libraries combined under one library tab: The old HSLVacuuBrandPump library (Low Level Steps) and the new HSLStarBVSLib (High Level Steps).

HSLStarBVSLib: (recommended – HIGH LEVEL FUNCTIONS)		
Command	**Icon**	**Action Performed**
BVSAbort		This function is used to stop all pump units and shut down their connections in an abort handler.
BVSGetAmbientPressure		Returns the ambient pressure measured with the specified pump unit.
BVSGetSimulationMode		Returns whether simulation mode is set for the specified BVS or not.
BVSInitialize		This function initializes the connection to the specified BVS.
BVSSetSimulationMode		Sets the specified BVS to simulation mode.
BVSTerminate		This function closes the connection to the specified BVS.
BVSTrack		Tracks a BVS volume move to the vector database.
BVSVacuum		Runs the vacuum process on the specified BVS.
BVSVacuumTrack		Runs the vacuum process on the specified BVS. The volume move is tracked to the vector database.

HSLVacuuBrandPump: (older – LOW LEVEL FUNCTIONS)		
Command	**Icon**	**Action Performed**
Initialize		Initializes one of up to four pumps on a RS232 COM port.
Terminate		Closes connection of a selected pump.
Request actual pressure		Requests pressure of a selected pump.
Open Air Admittance Valve		Opens air-bleed valve at a selected pump.
Start Pressure Control		Starts pressure control for a desired duration and, after timeout, open the air-bleed valve for a desired time if necessary.
Wait For Pump Stopped		Waits for the termination of the "**Start Pressure Control**" Command.
Stop Pump Immediate		Stops a running "**Start Pressure Control**" Command immediately. The air-bleed valve will be opened. This command is provided for emergency cases (e.g. error handling).

24.2 Integration of BVS / CVS

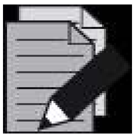
The labware for the CVS Carrier can be downloaded from the Resource Center**.**

The labware for the BVS / CVS Carrier is saved in the folder "**Microlab STAR CVS**".

Three definitions are predefined for BVS / CVS Carriers:

- with no shaker
- with one shaker on position 1
- with one shaker on position 2

No labware is preloaded on the vacuum system carriers.



NOTE

Three different versions of vacuum systems exist. All versions need the same programming steps, but different labware. If it is not certain which labware is at hand, teach the position inside the active position to avoid step losses during plate transports.

The z-position within the active position of the vacuum system is shown with different levels. See image below.

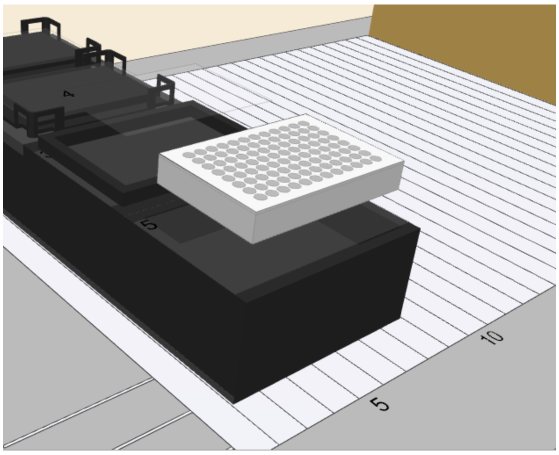


Plate on position 7

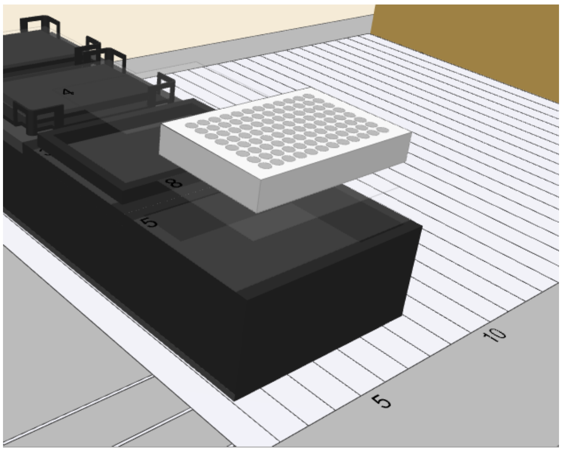
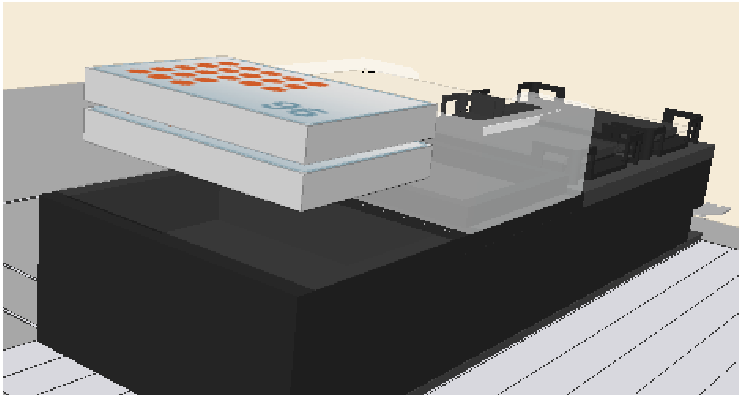
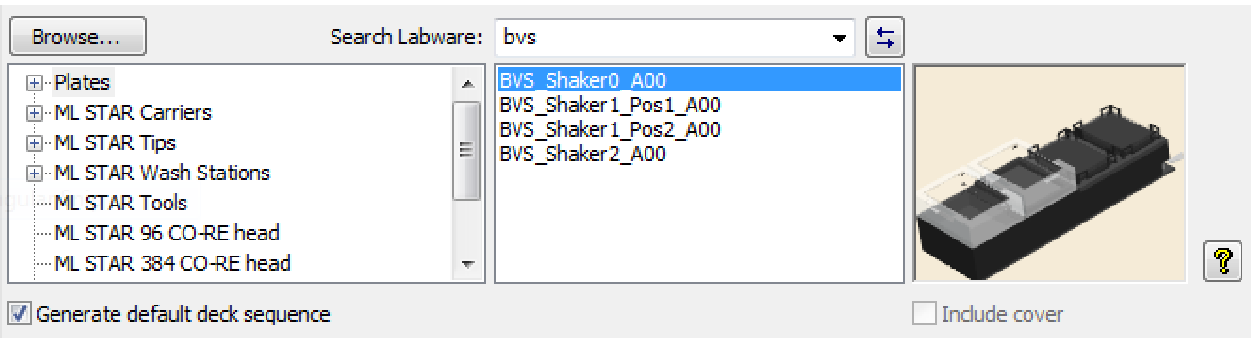


Plate on position 8

Use the zooming and rotating functions to position the labware in a correct level.



Use the "**Search Labware**" Text Field to display all BVS Carriers. Select the desired one.



Creating the Deck Layout

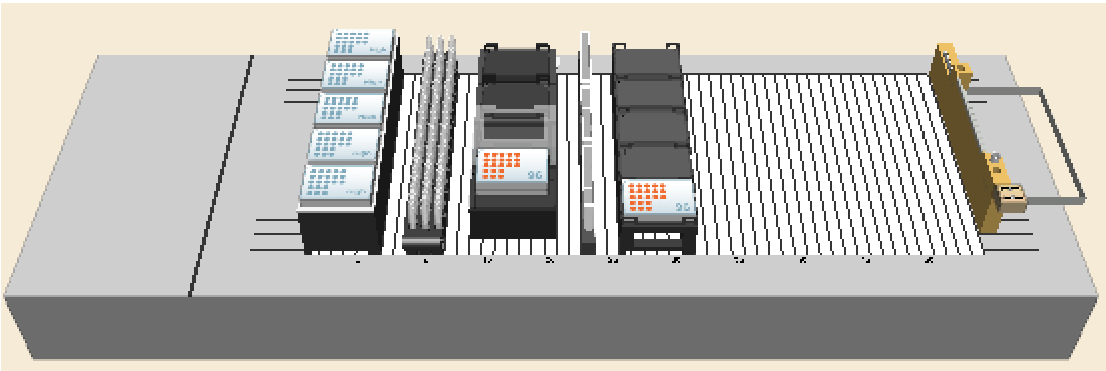
Add the following labware to the Deck Layout:

- A tip carrier “**TIP_CAR_480_HT_A00**” on the deck
- Three sample tube carriers “**SMP_CAR_32_12x100_A00**”
- The BVS vacuum system “**BVS_Shaker0_A00**” to the deck
- A reagent carrier “**RGT_CAR_3R_A01_0001**” (troughs named “**Reagent1**”, “**Reagent2**”)
- A plate carrier for Deep Well Plates “**PLT_CAR_L5AC_A00**”
- The ”**COREGripTool_AtWaste_1000ul**” on the waste block

The following plates are needed:

- Corning_96_Filter on position 7 (Top) of the BVS. Name it ‘FilterPlateActive’
- Cos_96_RD in position 8 of the BVS on the ‘Bottom’ position, named ‘TargetPlateBVS’
- The same Cos_96_RD in position 5 of the Plate Carrier, named ‘TargetPlateCarrier’

The deck should like the image presented below.



Creating the Sequences

In addition to the default sequences, only one sequence over all three sample Carriers named “**AllSamples**” is needed.

Creating the Method

The method for the BVS / CVS should look like the image shown below:

1	Comment <Init both vacuum system and instrument>
2	Initialize (Single Step) on ML_STAR Channel (1..8): 11111111, Optimized channel use: All sequence positions, Use default waste: On, Always initialize: Off 3 return value(s) .
3	BVSInitialize of HSLVacuuBrandPump HSLStarBVSLib::BVSInitialize(1, 1)
4	Comment <Pipette from tubes into the filterplate>
5	Pipette - Simple (1-1) from Microlab® STAR Smart Steps Instrument short name 'ML_STAR', Standard pipette procedure: Mode: Simple (1-1), Pipette volume [µl]: '100'. Aspirate sequence: 'ML_STAR.AllSamples (controlling sequence)', Dispense sequence: 'ML_STAR.FilterPlateActive'.
6	Comment <Aliquote reagent from Reagent 1 onto filterplate>
7	Pipette - Aliquot from Microlab® STAR Smart Steps Instrument short name 'ML_STAR', Aliquot procedure, Volume per well [µl]: '100'. Aspirate sequence: 'ML_STAR.Reagent1', Dispense sequence: 'ML_STAR.FilterPlateActive (controlling sequence)'.
8	Comment <Activate vacuum pump to wash the samples>
9	BVSVacuum of HSLVacuuBrandPump HSLStarBVSLib::BVSVacuum(1, 300, 300, 1, 200, ReachedPressureOfBVS)
10	Comment <Move filter plate and Manifold to PARK position>
11	CO-RE Grip Transport on ML_STAR Transport labware from 'ML_STAR.bvs_manifoldtop_a00_0002' to 'ML_STAR.bvs_manifoldtop_a00_0001' 1 return value(s) .
12	Comment <Move target plate from carrier into the lower position of BVS>
13	CO-RE Grip Transport on ML_STAR Transport labware from 'ML_STAR.TargetPlateCarrier' to 'ML_STAR.TargetPlateBVS' 1 return value(s) .
14	Comment <Move filter plate and Manifold to ACTIVE position>
15	CO-RE Grip Transport on ML_STAR Transport labware from 'ML_STAR.bvs_manifoldtop_a00_0001' to 'ML_STAR.bvs_manifoldtop_a00_0002' 1 return value(s) .
16	Comment <Aliquote reagent from Reagent2 onto filterplate>
17	Pipette - Aliquot from Microlab® STAR Smart Steps Instrument short name 'ML_STAR', Aliquot procedure, Volume per well [µl]: '100'. Aspirate sequence: 'ML_STAR.Reagent2', Dispense sequence: 'ML_STAR.FilterPlateActive (controlling sequence)'.
18	Comment <Activate vacuum pump to solve the samples and bring it into the target plate>
19	BVSVacuum of HSLVacuuBrandPump HSLStarBVSLib::BVSVacuum(1, 300, 300, 1, 200, ReachedPressureOfBVS)
20	Comment <Move filter plate and Manifold to PARK position>
21	CO-RE Grip Transport on ML_STAR Transport labware from 'ML_STAR.bvs_manifoldtop_a00_0002' to 'ML_STAR.bvs_manifoldtop_a00_0001' 1 return value(s) .
22	Comment <Move target plate from the lower position of BVS back to the carrier>
23	CO-RE Grip Transport on ML_STAR Transport labware from 'ML_STAR.TargetPlateBVS' to 'ML_STAR.TargetPlateCarrier' 1 return value(s) .
24	BVSTerminate of HSLVacuuBrandPump HSLStarBVSLib::BVSTerminate(1)

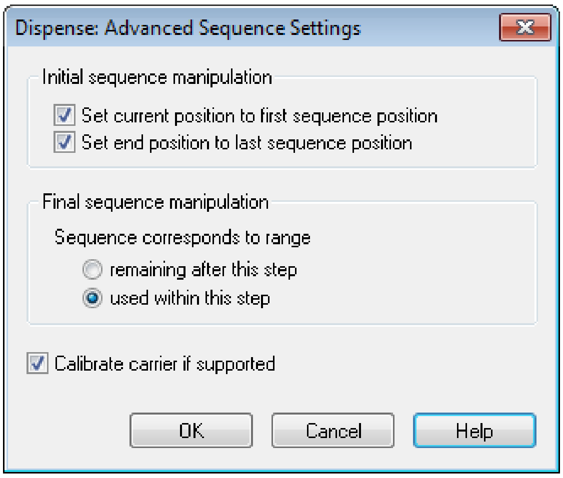
Method Analysis

Step 1: Initialization of the instrument and the VacuuBrand pump.

Step 2: Pipette samples from tubes into the filter plate on the manifold.

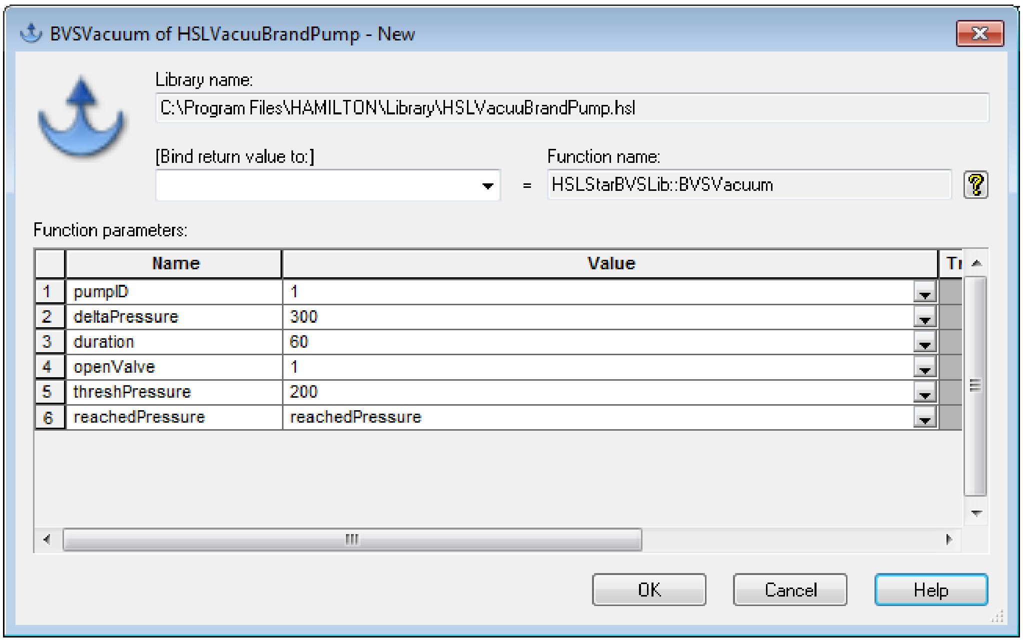
Make sure that the dispensing details settings in step 5 of the Smart Step are set as shown below.

The final sequence manipulation is now used to reset the filter plate sequence back to the start point (current = 1).



Step 4: Aliquote reagent 1 over all filter plate positions.

Step 5: Activate the vacuum pump to wash the samples.



PumpID: The id of the vacuum pump. Having more than one pump, count up from 1.

DeltaPressure: The negative pressure in millibar, calculated from ambient pressure.

Duration: The time of the vacuum action in seconds.

OpenValve**:** 1 opens the valve after the duration, 0 leaves the valve closed so that the pressure is kept in the vacuum chamber.

ThreshPressure: This is the limit at which the system will indicate the vacuum action as OK (e.g. ambient pressure was read 900 mbar, delta pressure is set to 300 (that means down to 600 mbar) and the threshPressure is set to 200 (that means 700 mbar). If the system is able to reach the threshPressure, the action will be stated as OK.

ReachedPressure**:** Here, a variable that holds the value of the deepest pressure reached during this run (to write in the trace file or for your own decisions within the method) can be inserted.

For the steps that follow, refer to the method presented in the preceding two pages

Step 6: moves the manifold with the filter plate to the PARK position of the BVS using CO-RE Grip.

Step 7: moves the target plate from the plate carrier into the lower position of BVS, with the CO-RE Grip.

Step 8: moves the manifold with the filter plate back to the ACTIVE position of the BVS.

Step 9: aliquots the solving reagent from the ReagentCarrier onto the filter plate. This is used to solve the samples from the filter so they can be transferred to the target plate underneath the filter plate. Copy step 4 and change the aspirate position.

Step 10: Vacuum action to bring the samples from the filter plate in the TargetPlate. Copy step 5 and change the duration to 300 seconds.

Steps 11 and 12 move the manifold to the park position (copy step 6) and bring the target plate back to the plate carrier.

Step 13: will finally terminate the connection between the vacuum pump and PC.

NOTE

The vacuum settings (pressure and duration) are dependent on the filter plates and the application.

NOTE

*For transports of the BVS/CVS manifold top, the "**Grip width**" and "**Grip height**" has to be defined correctly, especially when working with the CO-RE gripper. Otherwise, pipetting channels may be damaged.*

24.3 Activating BVS / CVS Maintenance

Activation of BVS / CVS maintenance requires the source files "**HslStarLineMaintMetConst.hs_**" and "**HslStarLineMaintMetConst.stp**". These files can be found in the **…\HAMILTON\Library** Folder.

The files can be modified individually by following the instructions below:

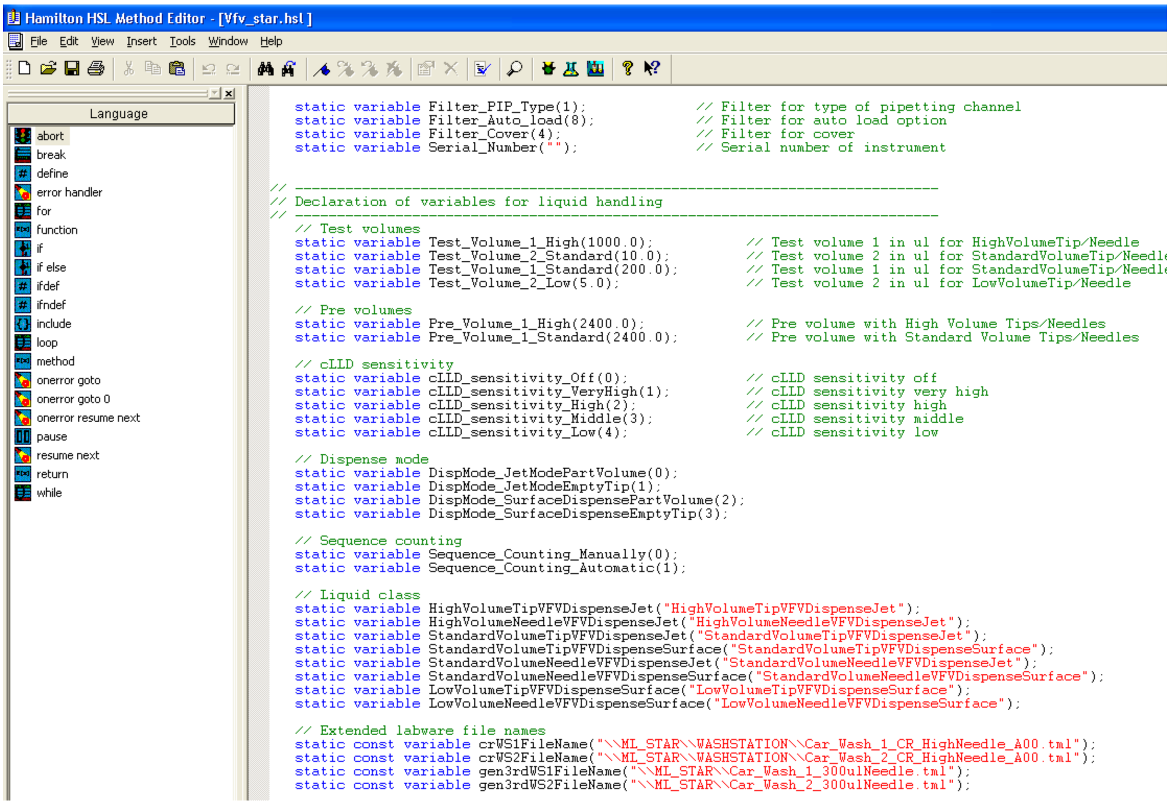
1. Make a copy of the "**HslStarLineMaintMetConst.hs_**" HSL file.
2. Modify the value of the constant "**isBVSInstalled(hslFalse);**" to "**isBVSInstalled(hslTrue);**".
3. Save the file.
4. After checking the syntax, the same file with an extension of "**.stp**" will be attained.

5. Rename the original files to something similar to "**old_ HslStarLineMaintMetConst.hs_**" and "**old_ HslStarLineMaintMetConst.stp**".
6. Finally rename the files that have been generated to the original names.

25 The HSL Method Editor

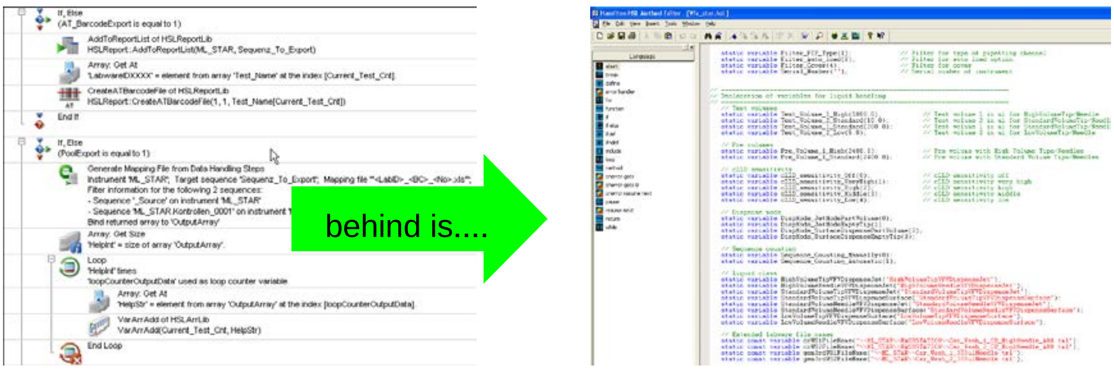
HSL or Hamilton Standard Language is the underlying programming language that the VENUS Software is operated from. HSL is a C/C++ style programming language. To use HSL, the HSL Method Editor must be started. It is a separate program that is located in "**C:\Program Files\Hamilton\Bin\HxHslMetEd.exe**" (depending on the language/installation settings).

The Editor is an acclaimed and outstanding text editor. An hsl file can be written using a text editor however the checksum lines have to be checked personally.



Before an HSL file is run, it has to be compiled (translated) into a basic computer language. This is done by the "**MICROLAB STAR Run**" Program, which should already be familiar.

Unlike a C/C++ program, the file is compiled just before it is run. This is done every time it is run. The Hamilton Method Editor is actually a code generator that creates an HSL code simultaneously**.** The Run Engine (compiler) runs the HSL file, not the '.med' file.



*.med *.hsl

25.1 HSL File Types

HSL is usually used to create libraries. There are actually 2 HSL file types that may be encountered:

Files with a file extension of “**.hsl**” are known as header files while files with an extension of “**.hs_**” are known as Source Files or Implementation Files.

Typically, header files will contain only declarations of functions and variables while source files contain the actual code that is in the function (known as the function implementation).

There are three main reasons to split into header and source files:

1. It allows another person to look only at the header file and know what the functions do. It is not necessary to analyze they work.
2. Sometimes there is a need to create additional functions, for the functions in the header to work but, may not want others to see them. Anything declared in the implementation file is not visible when the declaration file is included.
3. It enhances the speed of the VENUS Method Editor.

25.2 HSL Syntax

Most lines of code must end with a semicolon (;). Generally, a new line is started after a semicolon. But this does not need to be done (just makes the code easier to read).

The space between lines is known as whitespace and has no function.

Use of a double forward slash (//) indicates that the following characters in this line are comments only. A forward slash followed by a star (/*) together with the star followed by a forward slash (*/) indicates a comment block which can be separated over multiple lines. This will be interpreted as comment text and is not executed at runtime.

25.3 Code Block

A code block contains the actual coding in the HSL files. The following can be seen here:

- Variable declaration and calculation
- Conditional processing (loops and if, else statements)
- Function calls
- A Variety of other things

Code blocks always begin and end with curly brackets { … }.

25.4 Keywords

Keywords are reserved words that perform specific actions in HSL such as declaring variables and functions. It is forbidden to name anything else with the same name as the keyword, e.g. it is not possible to name a variable 'abort' within HSL.

A list of keywords is shown in the table below:

Keywords		
abort	function	pragma
break	global	private
char	goto	quit
const	if	return
device	ifdef	resume
debug	ifndef	sequence
define	include	short
dialog	lock	static
echo	long	struct
else	loop	string
endif	method	synchronized
error	namespace	timer
exit	next	unlock
event	object	variable
file	once	void
float	onerror	while
for	pause	__filename__

25.5 Operators

Operators represent logical operations such as addition and subtraction. Operators can be used in variable calculations or in conditional processing (if, else statements) to compare data and make decisions. HSL has a full range of operators, including arithmetic, logical, bit-wise, and assignment operators. Listed below are the different operators.

<table>
<tr><th>Symbol</th><th>Operator</th><th>Associativity</th></tr>
<tr><td>||</td><td>logical OR</td><td>Left</td></tr>
<tr><td>&&</td><td>logical AND</td><td>Left</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>|</td><td>bitwise OR</td><td>Left</td></tr>
<tr><td>&</td><td>bitwise AND</td><td>Left</td></tr>
<tr><td><</td><td>less than</td><td>Left</td></tr>
<tr><td><=</td><td>less than or equal to</td><td>Left</td></tr>
<tr><td>==</td><td>Equal</td><td>Left</td></tr>
<tr><td>!=</td><td>not equal</td><td>Left</td></tr>
<tr><td>>=</td><td>greater than or equal to</td><td>Left</td></tr>
<tr><td>></td><td>greater than</td><td>Left</td></tr>
</table>

Symbol	Operator	Associativity
%	remainder	Left
+	add	Left
-	subtract	Left
*	multiply	Left
/	divide	Left
!	not	Right
^	power	Right
-	unary minus operator	Right
++	increment	Left
--	decrement	Left
=	assignment	Right

25.6 Variables and Objects

Variables are the basic components of a source code. They are used to store/represent information. A variable can be named according to preferences, except for a predefined HSL keyword or a function. It is possible to perform calculations with variables just like in the graphical Method Editor.

There are different types of variables. The most common one only holds a value, e.g. a pipetting volume. But since variables are objects, these objects could also store other information more than just a value. A variable can also hold a sequence, a file, etc.

25.6.1 Variable Declaration

To declare a variable, use this code:

- Variable myVariable;

This will create a new variable named myVariable, and assigns zero (0) as the default value to it.

To initialize a variable with a specific value, use the following syntax:

- Variable myVariable(5); // Now, the variable holds the integer value 5
- Variable myVariable("**Hello**"); // this variable holds the string value Hello
- Variable myVariable(hslTrue); // boolean declaration – true or false

Variables always need to be declared before being used at the start of the code block. They cannot be declared in the middle of the code.

25.6.2 Variable Scope

As stated previously, variables need to be declared at the beginning of a code block. These variables are local to that code block. In other words, they only exist in that section of code. There are a few other options however:

- *Global variables* are declared at the beginning of a source file with the keyword ‘global’. These variables can be used all throughout the code – even in a different library or namespace.
- *Static variables* are similar to a global variable but are global only to that certain library/namespace. They are declared using the keyword ‘static’.

All other variables are local to the code block in which they are declared.

25.6.3 Array Declaration

Arrays are used to access a list of objects through the array name and the array index. An array can be of any sort of object (e.g. array of variables, array of files, array of sequences, etc…).

To declare an array:

1. Start by using the keyword of the type of the array (i.e. variable, file, sequence, etc…)
2. Specify a name to it, just like a variable.
3. Add a set of square brackets, [ ] right after the name.
4. An empty set of brackets creates an empty, undeclared length array.
5. If a number is enclosed in brackets, [5] it sets the size of the array.
6. Unlike in the graphical Method Editor, arrays in HSL are zero based, not one based.

A few examples:

- Variable ArrayOfVariables[ ]; // an empty array
- Variable ArrayOfVariables[100]; // an array with 100 elements
- Sequence ArrayOfSequences[ ]; // an empty array of sequences

25.6.4 If / Else Statements

HSL has the ability to compare things and make decisions. This is done with ‘If, Else’ statements. These are used in the exact same manner as they are in the graphical Method Editor, which should already be familiar.

They can be nested within each other to check multiple conditions.

They are more powerful than they are in the graphical Method Editor since multiple conditions at the same time can be checked. This eliminates the need for multiple nested ‘If, Else’ statements.

‘If’ and ‘If, Else’ statements will look like shown below.

```
If(var1 == 2)
var2 = 5;
Else
var2 = 10;
```

This example starts a new code block for multiple instructions

```
If(var1 == 2)
{
var2 = 5;
var3 = 10;
}
```

Checking multiple conditions can also be as follows

```
If(var1 == 5 && var2 == 10)
{
DoSomething();
}
Else
{
DoSomethingElse();
}
```

The above checks if both expressions are true. It is also possible to check other conditions such as ‘Or’ expressions. See the HSL Operators to know which operators can be used.

25.7 Loops

Just like in the graphical Method Editor, loops can be used in HSL to do certain things multiple times.

There are 2 types of loops in HSL:

‘for’ loops: These are used to do something a predetermined number of times.

‘while’ loops: These are used to do something while an expression is true.

The expression is similar to the ‘if, else’ statements.

There is also a ‘break’ command to break a loop when a certain condition is met.

For loop statements look like this:

```
for(i = 0; i <= 100; i++)
{
/*…do something…*/
}
```

The ‘i’ here is a loop variable.

It is originally assigned a value. In this example it is declared with a 0 value.

The second section of the *for* statement instructs the loop to keep going until i is greater than 100.

The last part of the statement increments the loop variable by 1 every time the loop is executed.

While loops look like this:

```
while(myVar < 2000)
{
/*…do something…*/
}
```

The expression in the while loop is just like the expressions that are used in ‘If, Else’ statements. Usually the code block of the loop performs some action to change the result of the expression.

25.8 Functions

Functions are pieces of code that perform repetitive tasks. Functions are analogous to sub-methods in graphical Method Editor files. Functions can (but don’t have to) accept variables and other objects as inputs. Similarly, functions can return a value of some sort (variable, sequence, file, etc.).

A function call looks like the following:

myFunction(); // This function does not return anything

var1 = myFunction(); // this function returns a variable

var1 = myFunction(5); // this function returns a variable and has a variable as an input

25.8.1 Function Declaration

A function declaration is just a statement allowing the compiler to know what functions are in existence. A function implementation is the actual code of the function. This tells the compiler how the function actually works.

The declaration and implementation can be done at the same time but is usually split into separate parts. The declaration can be seen in the header file (.hsl) while the implementation can be seen in the source file (.hs_).

A function declaration looks like this:

```
function myFunction(variable var1) variable
```

The parts of the declaration are as follows:

function: This tells the compiler that this is a function.

myFunction: This is the name of the function (however wanted).

variable: This first one is telling the compiler that ‘myFunction’ takes in one variable called ‘var1’. There can be multiple type declarations here!

variable: The second variable declaration is telling the compiler and the programmer that this function returns a variable type object.

The variable declarations can be replaced with any type of object, e.g. sequence, file, etc.…

A function does not have to return a variable. Some functions return other objects such as sequences or files while other functions can return nothing at all (this is marked by the keyword void). Sometimes an "**&**" can be seen after the word variable in the list of passed in variables/objects:

```
function myFunction(variable& var1) void
```

This ampersand is telling the compiler that the variable that is passed in is also passed out with a value that has potentially changed. This is called passing a variable by reference.

If there is no ampersand then the variable that is passed in is not changed by the function.

25.9 Namespaces

A namespace is a way to protect the functions of the library/sourcecode. This is a way to distinguish where a function comes from. For instance, when using 2 libraries and they both contain a function called 'Init', the compiler would not be able to distinguish between the 2 and would produce an error. If a namespace is used for each library, the compiler gets information about the origin of the function.

Namespaces are declared using the 'namespace' keyword.

All the code belonging to the namespace has to be enclosed by opening and closing curly brackets.

25.10 Using Libraries in HSL

Just like in the graphical Method Editor, other HSL libraries can be used in the code in order to access all of the functions that have already been created. Libraries are always included at the beginning of the source file. The preprocessor is used to include these files.

The preprocessor is a 'macro' that runs before the source code is compiled. This allows the compiler to get instructions on what is needed for the code to compile correctly.

25.10.1 Preprocessor Syntax

Preprocessor commands are always prefixed with a '#' sign.

These are usually seen at the beginning of an HSL file but also sometimes at the end.

Typically, in HSL 2 types of preprocessor commands can be seen:

Include commands. These are used to tell the compiler that some features from another HSL file (library) are used.

Include commands look like this:

#include "**HSLMthLib.hsl**"

Inclusion guards. These are used to tell the compiler to check whether a particular library is already included through some other bit of source code. This is important because trying to include the same library more than once will cause an error.

Inclusion guards look like this:

```
#ifndef __HSLMthLib_hsl__
#define __HSLMthLib_hsl__    1
/* ….
…. hsl code which implements the MthLib
…. */
#endif
```

25.10.2 HSL Runtime Inclusion Guard

In every library there is a special inclusion guard that looks like the following:

- *#ifndef HSL_RUNTIME*
- *code here*
- *#endif*
- *#ifdef HSL_RUNTIME*
- *code here*
- *#endif*

This is an inclusion guard that is checking to see if it is programming in VENUS or if the run engine is running. The library header will be presented in the '#ifndef' section and the implementation will be presented in the '#ifdef' section. It is located in these sections so that the actual commands are not compiled while programming in VENUS. This makes the editor tools faster.

25.10.3 Add Bitmaps

Adding those little icons to the library commands (seen while programming in VENUS) is easy. First, draw or download a bitmap image. The dimensions need to be either 16 x 16 or 32 x 32 pixels. Save the bitmap image with the same name as the library in the library directory. VENUS will do the rest of the steps and automatically load the icon when the library is included.

To create a separate picture for a particular function, save the bitmap image with the library name and the function name separated with a period (*libraryname.functionname.bmp*).

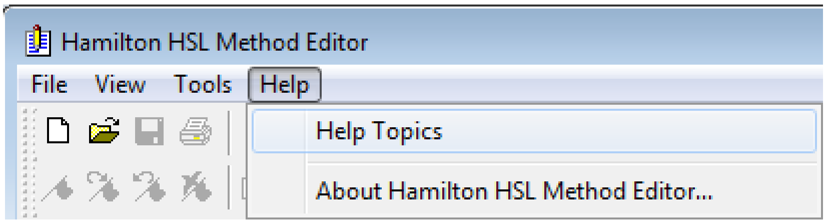
25.11 Add a Help File

Adding a help file for the library is just as easy as adding an icon. VENUS Help Files are CHM or HLP file structures. Only an appropriate editor is needed to create CHM or HLP files. A number of free editors can be found on the web.

Either create a file using an editor or write it in word and then use the CHM Editor to translate it. Save the help file with the same name as the library in the library folder just like bitmaps. It is highly recommended to create help files for a library. No one else will be able to support it otherwise.

25.12 VENUS Help Function

To get more information about the HSL Editor, refer to the VENUS Help Function:



It is located in the "**Help Menu → Help Topics**" and provides fundamental information in three categories:

- HSL Method overview / Keyboard shortcuts / Modifying a method
- All available menu options (File, Edit, View, Insert, Tools, Window, Help)
- An HSL reference guide to cover all aspects of HSL programming

26 Status Light

This section provides all necessary information about the status bar on the top front of the instrument.

The status bar provides information about:

- Instrument Status
- System Run Progress State
- Instrument Status Information with Progress Information (if activated)

NOTE

The optional Status Light Tower provides the same information.

26.1 Set Status Light (Single Step)

This step sets the status light to the given status.

Use this step to control the status light for dialogs programmed in the method.

ATTENTION

Make sure to program a Reset after the Set status has finished, e.g. “Wait for user input: Set”, show dialog, and after the dialog gets closed “Wait for user input: Reset”.

26.1.1 Step return values

This step returns 4 result values:

- Result value 3
  contains block data for the requested port.

Block Data	Description	Data Type
Num	Always set to 1	long
StepData	Contains the light status set.	long

- Result value 4

Description	Data Type
Contains the event set.	long

26.1.2 Dialog

The Set Status Light dialog enables parameters to be defined for the Set Status Light step.

- **Status**:
  Defines the status.
  The values for setting the variables are displayed in parentheses in the dialog.

  Possible settings:

  - (1) Error
  - (2) Wait for user input

- **Event**:
  Defines the event.
  The values for setting the variables are displayed in parentheses in the dialog.

  Possible settings:

  - (1) Set
  - (2) Reset

26.2 Status Light Progress (Single Step)

This step sets the current progress on the status light.

Use this step to control the progress for the running method.

NOTE

You can activate / deactivate the status light progress in the System Configuration Editor under the Microlab instrument settings.

26.2.1 Step return values

This step returns 3 result values:

- Result value 3
  contains block data for the requested port.

Block Data	Description	Data Type
Num	Always set to 1	long
StepData	Contains the light status set.	long

26.2.2 Dialog

The Status Light Progress dialog enables parameters to be defined for the Status Light Progress step.

- **Progress**:
  Defines the progress of the method.
  The values for setting the variables are displayed in parentheses in the dialog.

  Possible settings:

  - (1) 20 Percent
  - (2) 40 Percent
  - (3) 60 Percent
  - (4) 80 Percent
  - (5) 100 Percent

27 Third-Party Equipment

There is a wide range of Microlab instrument integrated devices for many applications to choose from.

The commands for these devices are functions of the additional libraries. These libraries are not part of the basic software package and need to be ordered in addition.

In order to make the library functions available, install the library into the folder *"**...\Hamilton\Library**"*.

To make the commands available, there is a need to link the library to the method. Add the "***.hsl**" Library File to the method through the "**Method → Libraries...**" Menu (refer to Section 11.12 Using Libraries).

For software drivers, please consult a local Hamilton Representative. Third-party equipment which can be integrated includes, but is not limited to the following:

- Readers (plate readers for fluorescence intensity, fluorescence polarization, time-resolved fluorescence, luminescence, UV/VIS absorbance)
- Plate Washers
- Heaters / Shakers
- Centrifuges
- Incubators and Plate Hotels
- Thermal Cyclers
- Plate Sealers
- Sample Drying Stations

28 Appendix

28.1 Glossary

Term	Definition
ADC	Anti-Droplet Control to prevent drops while pipetting highly volatile solvents.
Adjustment	Detailed positional setting for the hardware.
Air Displacement Tip	Hamilton CO-RE disposable tip
Aliquot	Aliquots are identical small volumes of liquid.
Aspirate	Procedure to draw up liquid into a pipetting device.
Autoload	Hardware assembly that enables automatic loading of the Microlab STAR. It consists of a loading head movable in Y direction, which draws the carriers into the Microlab STAR and can read the barcodes on them.
AutoLoad Tray	Hardware unit. The carriers can be placed on it and held outside the Microlab STAR. The loading tray is attached to the Microlab STAR, to support the automatic loading and unloading process.
Barcode Mask	The barcode mask defines the basic structure of a barcode. It is a pattern to which a barcode must conform. The assignment of a specific Labware item can be done in this manner. The barcode mask can require a barcode to contain specific strings at fixed positions. It can also contain wildcards.
Barcode Reader	Device for reading sample/plate Barcodes. Part of the Autoload Option.
Carrier	Unit for loading plates, tubes and tips on the Microlab instrument deck. Loading process is carried out by the Autoload unit.
Container	A container defines a tube, vessel or a single well of a plate.
Container Identification	Barcode for the identification of a container. Servers for a unique identification of a vessel, e.g. a sample test tube.
Deck	The work surface of the Microlab instrument. It presents at the same time the greatest possible area, cf. *Work Area*. The deck is divided into tracks. Carrier placement is defined by the tracks, as long as they are in the operating range of the pipetting area.
Deck Layout	A collection of labware placed upon a deck.
Dispense	To distribute quantities of liquid from a pipetting device.
Docking Station	The long bar at the back of the Microlab STAR instrument for guiding the cables and the tubing for accessories.

Term	Definition
Firmware	Lower-Level program code that is carried out on the processors of the Microlab Instrument
Front Cover	Protective covering for the Microlab Instrument, featuring a hinged front window made of transparent Acrylic glass. With this option and assembly, the work surface of the Microlab instrument is covered in such a way that it is shielded from user intervention and other outside influences (such as dust). At the same time, it protects the user from the movements of the Microlab instrument.
Hardware Error	Type of error that is caused by a technical problem with the hardware
HSL	Hamilton Standard Language
HHS	Hamilton Heater Shaker. Unit to heat of shake micro plates in SBS format.
HSB	Heater Shaker Box. Interface unit which is needed if more than two HHS are being used.
Instrument	Hardware of the Microlab instrument (mechanics, electronics, and firmware)
Instrument Steps	The commands made available by the firmware for controlling the Microlab instrument.
Labware	Refers to movable items to be placed on the Microlab instrument deck, such as carriers, containers, or racks.
LIMS	Higher level data processing system, generally known as Laboratory Information Management System, also LMS.
Liquid	Includes all kinds of liquids, among which are included reagents, controls, standards, wash fluids.
LLD (Liquid Level Detection)	Detection of liquid surface which may be achieved either by pressure or capacitive signal detection.
Loading/Unloading	The process by which plate, tube and tip carriers are brought on and off the Microlab instrument deck. This is automatically performed via Autoload unit.
MAD (Monitored Air Displacement)	Aspiration monitoring Feature. During the aspiration process, the pressure within the pipetting channel is measured in real time.
Method	The method contains all instruction that must be executed during a run.
Microlab instrument Software	Software to run the Microlab instrument.

Term	Definition
MTP (Microtiter Plate)	In general, a Microtiter Plate (micro plate) is assumed to have 96 wells (8 x 12) 9 mm wide. There are also plates with 384 wells (16 x 24 / 4.5 mm), or others with a different size
Pause	Interruption of processing. The current processing steps are completed.
Pipetting	Transfer of liquids from one container to another.
Pipetting Arm	Assembly equipped with the pipetting device and/or plate handler.
Pipetting Channel	Hardware including the function of picking up a tip aspirating, dispensing, tip eject, Liquid Level Detection and the Y/Z-movements
Pooling	Pipetting of different liquids in one well; 1, 2, 3…to n and n to 1, 2, 3...
Processing Step	Defines what must be carried out on the Microlab instrument, as well as the location it must be carried out and possible interaction with other system components or labware. The action is defined in accordance with the methods, the loading and the tasks.
Rack	Group of containers, as DWP, MTP, etc.
Rack Identification	Barcode for rack identification
Random Access	Means that every pipetting channel can access any position anywhere on the work area.
Run	Execution of the processing steps defined in the method with the aim of processing one or more liquids and containers (e.g. MTP). The run is a series of timed commands, in order to carry out processing on the Microlab instrument according to the processing plan.
Run Abort	Cancelled run by the user or by the Microlab instrument
Run Visualization	Visualization of the current run, reporting the status of the Microlab instrument.
Sample	Refers to a liquid in a unique identified container which is to be processed.
Sequence	A list that specifies the order of execution for an action on the Microlab instrument. Generally, two sequences are required: A source- and a target sequence (e.g., for pipetting from a source to a target sequence).
Stacker	Storage unit for racks
Side Touch	Pipetting on the wall of a container of its center to prevent droplets.

Term	Definition
TADM	Total Aspiration and Dispensing Monitoring. The pressure inside each individual pipetting channel is monitored, during aspiration and dispensing.
Tip	Disposable tip for pipetting
Tip Rack	Frame that holds the tips.
Tip Waste	Container for ejected tips.
Touch-Off	Type of dispensing where the tip approaches the bottom of the empty container so close as to allow the dispensed droplet to have simultaneous contact with the tip and the container bottom.
Trace	Record of the status during processing
Tube	A container for liquid, usually having a circular cross-section, and a cylindrical length section.
User	User of the software. Access rights for different types of users can be defined, such as operators, laboratory manager, etc.
Waste Container	A device on the Microlab instrument deck to collect used disposable tips.
Well	The individual container of a MTP or DWP.
Well Type	Geometrical shape of the well, such as U, V or flat.
Wick Side of Container	Type of dispensing where tip touches the side of the container and thus releases the droplet. This function is not available for the Microlab instruments.
Work Area	The area of the Microlab instrument to which access is provided during the processing. Elements to be pipetted or handled can be placed in this area.
Worklist	Information according to which a method is to be executed on the Microlab instrument. A worklist may contain different parameters e.g. Pipetting volume, heating temperature, shaking speed, etc.

HAMILTON

Web: www.hamiltoncompany.com
USA: 800-648-5950
Europe: +41-58-610-10-10

Hamilton Americas & Pacific Rim
470 Energy Way
Reno, Nevada 89502 USA
Tel: +1-775-858-300
Fax: +1-775-856-7259
sales@hamiltoncompany.com

Hamilton Europe, Asia, & Africa
Via Crusch 8
CH-7402 Bonaduz, GR, Switzerland
Tel: +41-58-610-10-10
Fax: +41-58-610-00-10
contact@hamilton.ch

To find a representative in your area, please visit hamiltonrobotics.com/contacts.