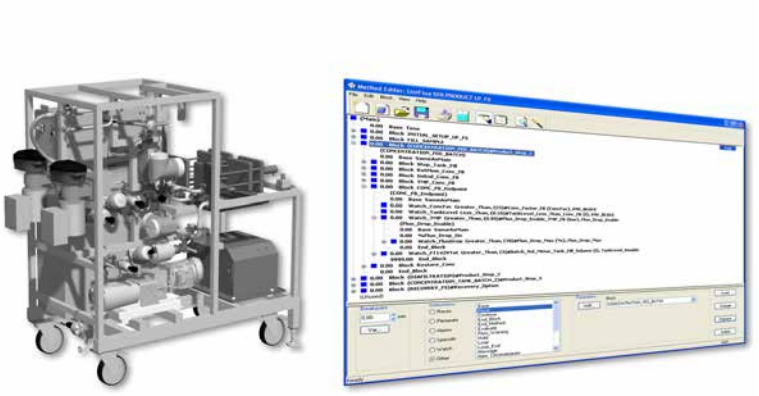
cytiva

UniFlux

Method Templates User Guide



cytiva.com

Table of Contents

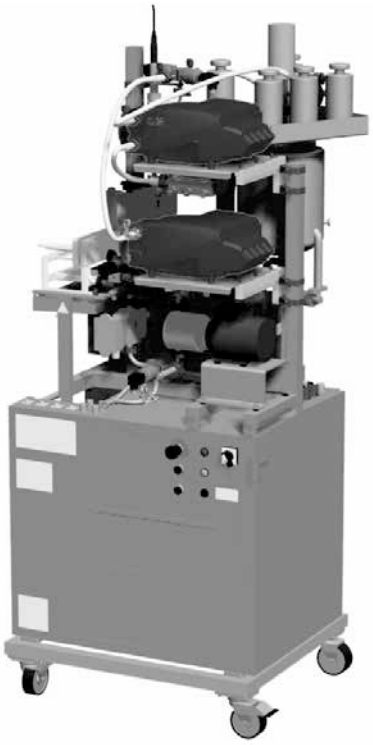
1 Introduction

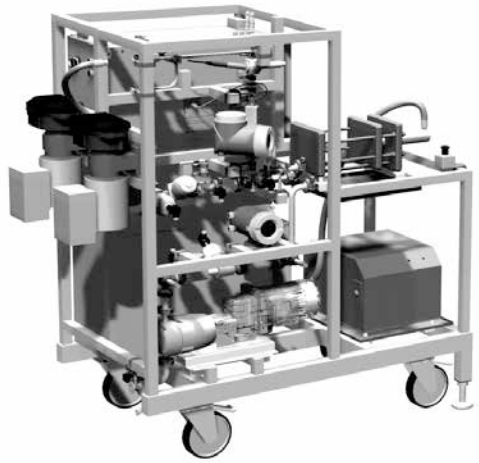
UniFlux systems

The UniFlux™ series is a standard line of cross flow filtration systems. The systems are intended for pilot to production scales biological separations. The systems are configured to operate hollow fiber cartridges suited for microfiltration applications (e.g., cell clarification/harvesting), or cassettes/hollow fibers for ultrafiltration applications (e.g., protein concentration and diafiltration in downstream unit operations).

UniFlux system range

The UniFlux systems are available in four sizes and are capable of processing batches of < 5 to > 10000 liters of feed material, see table below.

System	Flow range Typical use
UniFlux 10 	0.5 to 10 l/min Pilot scale production

System	Flow range Typical use
UniFlux 30 	3 to 60 l/min Small scale production
UniFlux 120 	12 to 120 l/min Medium scale production

System	Flow range Typical use
UniFlux 400 (filters not shown) 	40 to 400 l/min Large scale production

UNICORN

UNICORN™ software provides full automation with data logging capabilities of the entire cross flow process. UNICORN is 21 CFR Part 11 compliant, satisfying requirements for electronic records and signatures to meet increasing regulatory demands.

Method templates

UniFlux method templates offer a starting point for method creation. ***Templates*** are basic methods that can be used as a starting point for developing customized methods. The method variables in a suitable ***Template*** are adjusted to create a method for the current application.

2 Safety

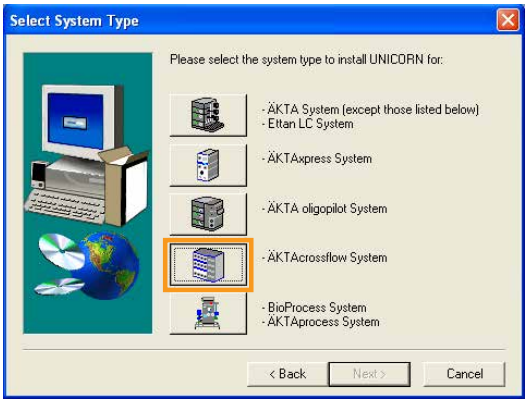
UniFlux Operating Instructions for your system describes how to use UniFlux systems safely. Read the safety instructions in this manual before installing, using or maintaining the system.

3 Installation

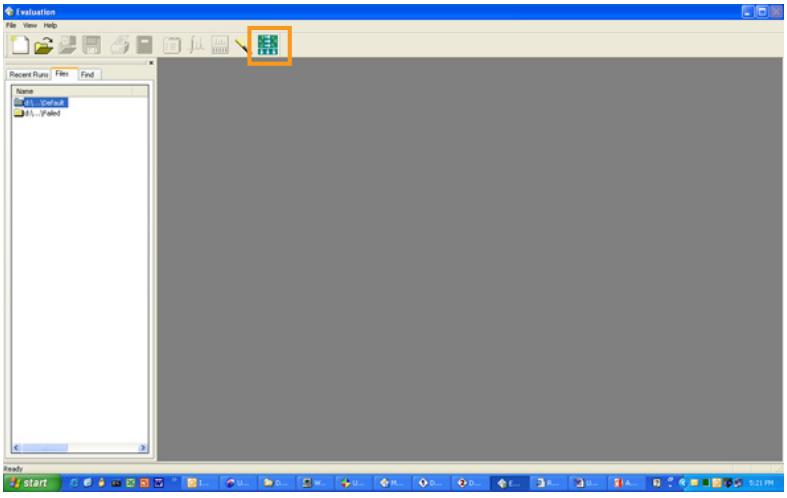
In this chapter

3.1 UNICORN

UNICORN installation requires the selection of the system type in order to install the appropriate features. For UniFlux systems select system type ***ÄKTAcrossflow*** in order to install the evaluation wizard for filtration. See the installation instructions in *UNICORN Administration and Technical Manual* for more details. UniFlux method templates require UNICORN 5.11 or higher.

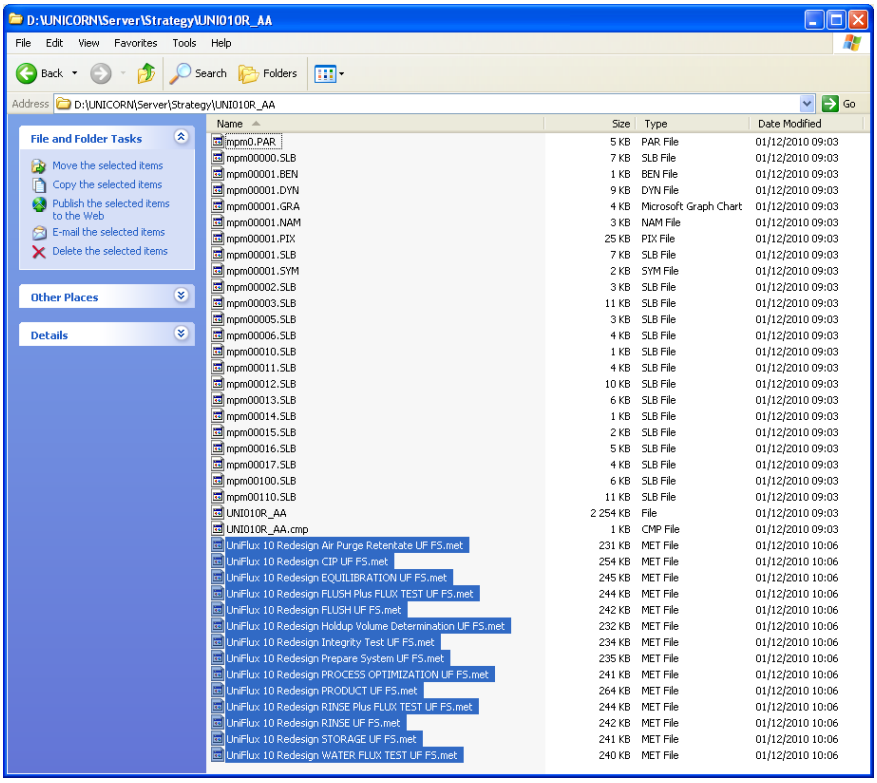


Confirm that the ***Membrane System Evaluation*** icon is displayed in the UNICORN ***Evaluation*** module. This icon shows that the evaluation wizard for filtration is installed.



3.2 Method templates

UniFlux method templates are delivered with file extension `*.met`. To install the method templates, copy and paste the templates into the strategy folder for the UniFlux system. The path to the strategy folders is `[drive]\UNICORN\Server\Strategy`.



Tip: *If desired, the templates can be renamed. For example, the "UF FS" designation can be removed.*

4 Method templates

In this chapter

4.1 Overview

How to use the method templates

Via ***Instant Run*** in ***System Control*** the user can immediately start a run from a selected method template. Migration of process parameters from development systems such as ÄKTAcrossflow™ are easily entered. Alternatively, users can amend the templates to match standard operating procedures (SOP) or master batch records (MBR).

Available method templates

A complete set of templates is available for the UniFlux systems 10, 30, 120, and 400, and for the following filter types: Ultrafiltration Flat Sheet Cassette (UF FS), Ultrafiltration Hollow Fiber (UF HF), and Microfiltration Hollow Fiber (MF HF). Each template set contains utility methods, non product methods, and a product method. UniFlux 10 UF templates also offer process optimization methods.

The table below gives an overview of the available templates.

Method templates	UniFlux 10 System			UniFlux 30, 120, 400 Systems		
	UF FS	UF HF	MF HF	UF FS	UF HF	MF HF
UTILITY METHODS						
Holdup Volume Determination	X	X	X	X	X	X
Air Purge Retentate	X	X	X	X	X	X
Integrity Test	X	X		X	X	
Prepare System	X	X	X	X	X	X
NON PRODUCT METHODS						
WATER FLUX TEST	X	X		X	X	
RINSE	X	X	X	X	X	X
RINSE Plus FLUX TEST	X	X		X	X	
FLUSH	X	X	X	X	X	X
FLUSH Plus FLUX TEST	X	X		X	X	
STORAGE	X	X	X	X	X	X
EQUILIBRATION	X	X	X	X	X	X
CIP	X	X	X	X	X	X
PRODUCT METHODS						
PRODUCT	X	X	X	X	X	X
PROCESS OPTIMIZATION	X	X				

System components

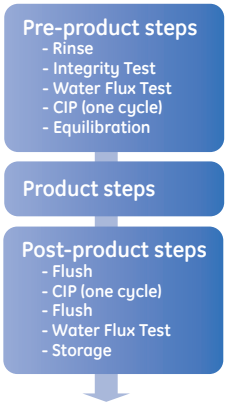
Templates are configured for systems with the following components:

- ***TransferFeedPump***
- ***PermeatePump***
- ***FeedTank***
- ***UV***

The components are selected in ***System Setup***. For systems that do not have all of these components, the template may require updating for syntax and/or function. For an example, see section *Example of syntax error, on page 30*.

Suggested sequence

At pilot scale a workflow sequence may not yet be defined. A suggested sequence for ultrafiltration application is shown below. For microfiltration applications, eliminate the ***Integrity Test***. The sequence of the steps may need to be adjusted depending on the application. Make sure to test all steps for each particular application.



4.2 Utility methods

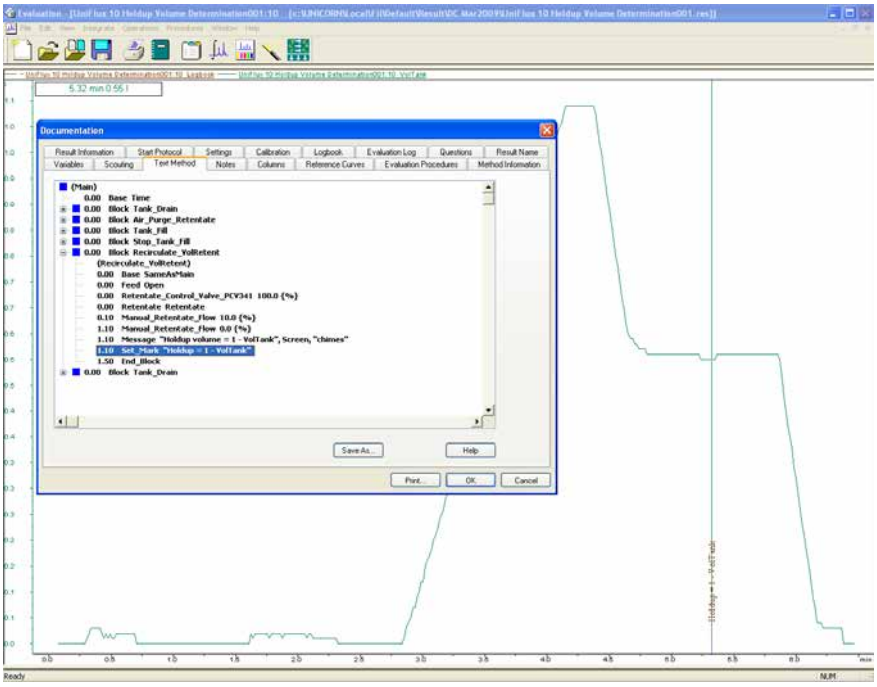
Available utility methods

Each UniFlux method templates set includes four utility methods:

- ***Holdup Volume Determination***
- ***Air Purge Retentate***
- ***Integrity Test***
- ***Prepare System***

Holdup Volume Determination

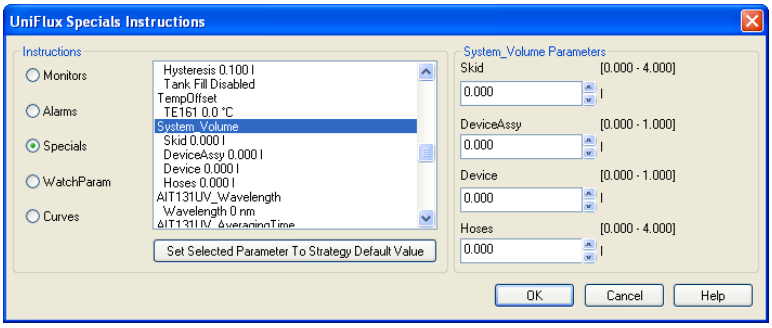
This method empirically determines the total recirculation holdup volume including retentate piping and filter. All liquid is removed from the recirculation loop via air purge. A set volume of liquid is transferred into the tank at 50% ***TransferFeedPump***. The default volume is 1 liter for UniFlux 10 and 10 liters for the larger systems. The liquid is recirculated for one minute. A ***Set_Mark*** is issued. The user calculates the difference between the ***Tank Fill Volume*** and the ***VolTank*** reading. The difference is the holdup volume.



The holdup volume is necessary for correct ***VolRetent*** data. This is critical for concentration factor and diafiltration factor calculations as these instructions latch onto the ***VolRetent*** value.

Edit the holdup volume

Follow the instruction below to enter correct holdup volume.

Step	Action
1	In UNICORN ***System Control*** module, select ***System*** →***Settings***. *Result:* The ***UniFlux Instructions*** dialog opens.
2	In the ***UniFlux Instructions*** dialog, select ***Specials*** and ***System Volume***. 
3	In the ***Skid*** field, enter the correct holdup volume. Click ***OK***.

Air Purge Retentate

This method uses the system air supply to remove liquid from the retentate in two steps:

- Air purge to feed drain (block ***Purge_Through_Feed_Drain***)
- Air purge to tank followed by tank drain (block ***Purge_Ret_Return_To_Tank*** followed by ***Tank_Drain***)

Air flow is generated by the instruction ***Integrity_Test_Pressure***. Default ***Integrity_Test_Pressure*** is 0.5 bar. Default air flow time is 0.5 minutes.

Air_Purge_Retentate blocks are included in all product method templates. Purging of the retentate lines can be performed prior to sample fill and/or during recovery to maximize yield. If the blocks are not wanted, they can be removed from the method.

Integrity Test

This method is used for ultrafiltration applications. A test of the system integrity is performed, followed by a test of the membrane integrity. The system integrity test is performed with the feed, retentate, and permeate valves closed. The membrane integrity test is performed with the permeate valve open for four minutes by default. A ***Set_Mark*** is executed and then used in the ***Evaluation*** module to measure the air flow curve value. Default ***Integrity Test Pressure*** is 1 bar.

Consult membrane manufacturer's protocol for air integrity test. Ideally, when membranes are stacked together in the holder, test each membrane individually to avoid false positives. Also, for multiple membranes stacked together in the holder, multiply the manufacturer's air flow specification by the number of membranes.

Prepare System

The method is used to rinse both the retentate and permeate flow paths. This removes the previous liquid in the system and fills it with the current liquid to be used with the filter. Each valve drain point is rinsed. Default values are 0.5 minutes for retentate drain rinse and 1 minute for permeate rinse at 10% ***Manual_Feed_Flow***.

In the method, default tank level for UniFlux 10 is 2 liters and default tank level for UniFlux 30, 120, and 400 is 20 liters. Default consumption for UniFlux 10 is 20 liters. However, tanks for UniFlux systems 30, 120, and 400 are supplied separately and are available in different volumes. Therefore the working tank volumes and consumption must be determined empirically for each configuration. Make sure to update the instructions based on current tank volume, see *Update tank volume, on page 29*.

The block ***Prepare_System*** is included in all non product method templates.

4.3 Non product methods

Available non product methods

UniFlux method templates set includes eight non product methods:

- ***WATER FLUX TEST***
- ***RINSE***
- ***RINSE Plus WATER FLUX TEST***
- ***FLUSH***
- ***FLUSH Plus WATER FLUX TEST***
- ***STORAGE***
- ***EQUILIBRATION***
- ***CIP***

Tip: ***RINSE***, ***FLUSH*** *and* ***STORAGE*** *are similar methods with different nomenclature respective of their function.* ***RINSE*** *differs in that after a default rinse volume has passed through the permeate of the filter a* ***Hold_Until*** *instruction follows that holds the method until the conductivity is* ***Less_Than*** *a set value. The purpose for the separate methods is to allow for ease of use when used individually or when combining steps into a larger method.*

General information

All non product methods contain the block ***Prepare_System***, see *Prepare System, on page 15*.

In all non product methods a ***Set_Mark*** (e.g.,***"WATER FLUX TEST"***) indicates the start of the method. The ***Set_Mark*** is inserted into the chromatogram and into the logbook during a method run.

WATER FLUX TEST

Measures the water flux at a single, user specified, test value. The ***Set_Eval_Mark*** instruction is used to allow UNICORN to automatically pull the data set into the evaluation protocol for normalized water flux comparisons at 25ºC and 1 bar.

Water Flux testing can be run on multiple cassettes installed in the holder. However, a true ***Water Flux Test*** result for each cassette must be run on each filter individually.

RINSE

Removes the glycerine (or other) solution from the filter, which was used by the manufacturer for storage. It can also be used to rinse a membrane that has been in storage solution, such as 0.1 M NaOH.

RINSE Plus FLUX TEST

Combines the two functions ***RINSE*** and ***WATER FLUX TEST*** for time and buffer savings.

FLUSH

Removes residual feed, contaminants, and insoluble substances from the filter before cleaning. It can also be used to flush the filter with water until residual solution (i.e., CIP solution) is removed to an acceptably low level.

FLUSH Plus FLUX TEST

Combines the two functions ***FLUSH*** and ***WATER FLUX TEST*** for time and buffer savings.

STORAGE

Equilibrates the filter in a solution that is chemically compatible with the filter for long term storage, such as 0.1 M NaOH.

EQUILIBRATION

Equilibrates the filter and system with a buffer having similar pH and ionic strength to the feed solution. The equilibration continues until any of the specified conditions is met. The ***EQUILIBRATION*** block counts off of permeate volume and starts with the sub block ***Totalizer_Reset*** that resets the permeate volume.

Endpoint conditions for EQUILIBRATION

Any of two blocks can be used to specify the conditions; ***EQ_Endpoint_Conditions_L*** or ***EQ_Endpoint_Conditions_G***. ***EQ_Endpoint_Conditions_L*** is included in the template as default. "***L***" stands for ***Less_Than***, and "***G***" stands for ***Greater_Than***. These refer to the test for the ***Watch*** conditions on conductivity, pH, and UV. The blocks also include a watch instruction for permeate volume.

The table below lists the ***Watch*** conditions of the blocks ***EQ_Endpoint_Conditions_L*** and ***EQ_Endpoint_Conditions_G***.

Signal	Watch instruction	Test	Action
Permeate volume	***Watch_FT142VTot***	***Greater_Than***	***END_BLOCK***
Conductivity	***Watch_CT101***	***Less_Than***/ ***Greater_Than***	***Cond_Stable_Baseline***
pH	***Watch_AT121***	***Less_Than***/ ***Greater_Than***	***pH_Stable_Baseline***
UV	***Watch_AIT131UV***	***Less_Than***/ ***Greater_Than***	***UV_Stable_Baseline***

Any combination of the above criteria can be used. Defaults for monitor signals are such that they are at their maximums or minimums dependent on block choice of "***L***" or "***G***". Their conditions will not occur unless adjusted to reflect the process conditions. For information on ***Stable_Baseline***, see *Watch for stable baseline, on page 51*.

Standard operating procedures may request that both conductivity and pH conditions be met in order for a filter to be considered equilibrated. A method text update can be done to nest conditions for one monitor followed by the next, see *Nested Watch instructions, on page 55*.

CIP

Cleans the filter. The user must choose the appropriate cleaning solution, and the time for cleaning. While 0.5 M NaOH is common, other solutions such as detergents may be used instead of or in combination with NaOH. CIP cycles can vary between 30-60 minutes.

The cleaning is performed with the system in total recycle at a default transmembrane pressure (TMP). The method provides the possibility for up to two separate cleaning cycles with an optional water flush between each solution. The first cycle can be used for an initial clean and discard of dirty solution. The second step can be used to bring in fresh CIP solution for a longer clean time. Alternatively, two cleaning solutions can be used.

Both CIP cycles are performed with permeate valve open. Single text modification can be done to run with permeate closed, for example, in the first cycle of a two cycle CIP.

Optional blocks

The table below describes the optional blocks of the ***CIP*** method template. If an optional block is not needed, replace the block with the block ***NO_STEP*** to keep the placeholder, or delete the block from the method.

Block	Description
Tank_Fill_Prewash	Provides an option to fully fill the tank and drain prior to block ***Prepare_System***. If, during processing, a tank volume greater than the ***Prepare_System*** volume was used, this block will provide a means to prewash the tank without having to increase liquid usage during ***Prepare_System***.
Fill_Tank_Full_Plus_Hold	Fills the tank completely full, overfilling through the top outlet. Connect flex hose here to waste.
Full_Tank_Flush	Works together with block ***Fill_Tank_Full_Plus_Hold_Option***. Overfills the tank with flush liquid through the top outlet. Connect flex hose here to waste.

Block	Description
System_Flush_Btw_CIP	Rinses the flow path when two CIP cycles are chosen. Not required, especially if the same CIP solution is used for both cycles. Rinses the retentate and permeate flow paths. Each valve drain point is rinsed. Defaults are 0.5 minute for retentate drain rinse and 1 minute for permeate rinse at 10% ***Manual_Feed_Flow***.

DeltaP retentate control

All non product methods use ***DeltaP*** for retentate control. In non product methods ***DeltaP*** is a favorable choice. There is no flow meter feedback, so water is acceptable to use with the magnetic flow meter on the recirculation side. ***DeltaP*** is also favorable as it is transparent to membrane surface area. The same setpoint can be used over the surface area range. Default is 1.0 bar for UF cassettes and hollow fibers, and 0.3 bar for MF hollow fibers. Shear may be favored over ***DeltaP*** in hollow fiber applications. Update the text method accordingly. For guidance consult *Operating Handbook - Hollow fiber cartridges for membrane separations*.

TMP setpoint

All non product methods for ultrafiltration have ***TMP*** 0.5 bar setpoint by default. The non product methods for microfiltration have no ***TMP*** set (open permeate).

4.4 Product methods

Available product blocks

The default ***PRODUCT*** method templates contain three product steps. The steps are entered as block variables for easy transition from one step type to another. The available blocks are:

- ***CONCENTRATION_FED_BATCH***
- ***CONCENTRATION_TANK_BATCH_1***
- ***CONCENTRATION_TANK_BATCH_2***
- ***DIAFILTRATION***
- ***DIAFILTRATION_ONLY***

If less than three product steps are to be used, any of the steps can be replaced with the block ***NO_PRODUCT_STEP*** or the block variable can be cleared followed by the deletion of the step.

CONCENTRATION_FED_BATCH

Used as the first step in a process when the total sample volume is greater than that of the tank volume. An initial working volume is set in block ***FILL_SAMPLE***. It can be the only step in the process or followed by subsequent steps such as ***DIAFILTRATION*** and ***CONCENTRATION_TANK_BATCH***.

CONCENTRATION_TANK_BATCH_1 and CONCENTRATION_TANK_BATCH_2

Used as a step in a process when the total sample volume is less than or equal to that of the tank volume. The volume is set in block ***FILL_SAMPLE***. It can be the only step in the process or followed by subsequent steps such as ***DIAFILTRATION*** and ***CONCENTRATION_TANK_BATCH***.

DIAFILTRATION and DIAFILTRATION_ONLY

Used as a step in a process to diafilter (buffer exchange) a sample. ***DIAFILTRATION*** is to be used as a second step after concentration (***CONCENTRATION_FED_BATCH*** or ***CONCENTRATION_TANK_BATCH***). ***DIAFILTRATION_ONLY*** is to be used as a first and only step in a process (i.e., clarification).

Endpoint conditions for product methods

The endpoint conditions for the product blocks are listed in the tables below.

Any combination of the criteria can be used. Whichever condition is met first will end the product step. Defaults are set to the maximum for ***DF_X_Fct***, ***ConcFac***, and ***FIR142VTot***. Defaults for monitor signals are such that they are at their maximums or minimums. Their conditions will not occur unless adjusted to reflect the process conditions. For information on ***Stable_Baseline***, see *Watch for stable baseline, on page 51*.

Standard operating procedures may request that both conductivity and pH conditions are met in order for ***DIAFILTRATION*** to be considered complete. A method text update can be done to nest conditions for one monitor followed by the next, see *Nested Watch instructions, on page 55*.

Endpoint conditions for CONCENTRATION_FED_BATCH

The endpoint conditions for block ***CONCENTRATION_FED_BATCH*** are set in block ***CONC_FB_Endpoint***.

The available conditions are listed in the table below.

Signal	Watch instruction	Test	Action
Concentration factor	***Watch_ConcFac***	***Greater_Than***	***END_BLOCK***
Tank level	***Watch_TankLevel***	***Less_Than***	***END_BLOCK***
Transmembrane pressure	***Watch_TMP*** (See page *Watch_TMP in microfiltration methods, on page 22*)	***Greater_Than***	For ultrafiltration methods:***Flux_Drop_Enable*** For microfiltration methods: ***END_BLOCK***
Permeate volume	***Watch_FT142VTot***	***Greater_Than***	***TankLevel_Disable***

Endpoint conditions for CONCENTRATION_TANK_BATCH

The endpoint conditions for the blocks ***CONCENTRATION_TANK_BATCH_1*** and ***CONCENTRATION_TANK_BATCH_2*** are set in blocks ***CONC_TB1_Endpoint*** and ***CONC_TB2_Endpoint***.

The available conditions are listed in the table below.

Signal	Watch instruction	Test	Action
Concentration factor	***Watch_ConcFac***	***Greater_Than***	***END_BLOCK***
Tank level	***Watch_TankLevel***	***Less_Than***	***END_BLOCK***

Signal	Watch instruction	Test	Action
Transmembrane pressure	***Watch_TMP*** (See page *Watch_TMP in microfiltration methods, on page 22*)	***Greater_Than***	For ultrafiltration methods: ***Flux_Drop_Enable*** For microfiltration methods: ***END_BLOCK***
Permeate volume	***Watch_FT142VTot***	***Greater_Than***	***END_BLOCK***

Endpoint conditions for DIAFILTRATION

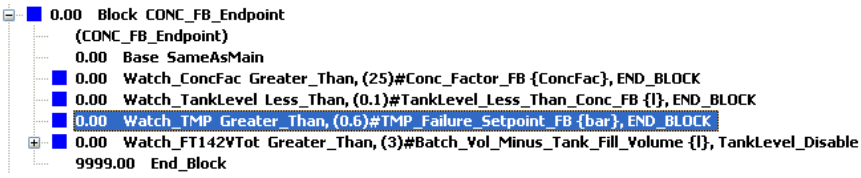
The endpoint conditions for the blocks ***DIAFILTRATION*** are set in any of the blocks ***DF_ENDPOINTS_L*** or ***DF_ENDPOINTS_G.*** '***L***' stands for ***Less_Than***, and '***G***' stands for ***Greater_Than***. These refer to the test for the ***Watch*** conditions on conductivity, pH, and UV.

The available conditions are listed in the table below.

Signal	Watch instruction	Test	Action
Diafiltration factor	***Watch_DF_X_Fct***	***Greater_Than***	***END_BLOCK***
Permeate volume	***Watch_FT142VTot***	***Greater_Than***	***END_BLOCK***
Conductivity	***Watch_CT101***	***Less_Than/ Greater_Than***	***Cond_Stable_Baseline***
pH	***Watch_AT121***	***Less_Than/ Greater_Than***	***pH_Stable_Baseline***
UV	***Watch_AIT131UV***	***Less_Than/ Greater_Than***	***UV_Stable_Baseline***

Watch_TMP in microfiltration methods

Each concentration block of microfiltration methods contains a ***Watch_TMP*** instruction with the corresponding variable ***TMP_Failure_Setpoint***, see below.

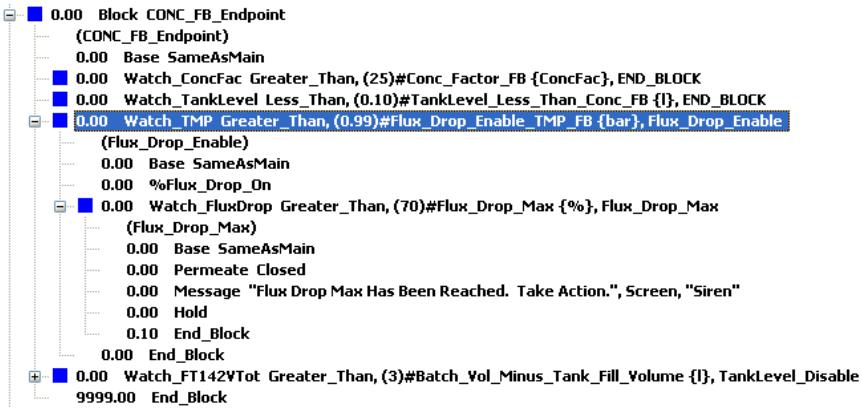


When the transmembrane pressure (TMP) exceeds the ***TMP_Failure_Setpoint***, the condition is met. The default setpoint is 0.1 bar less than the ***TMP_Alarm*** value. The default action is to execute the instruction ***End_Block***. The action can also be created as a sub block.

Note: *In cases of excessive fouling of the filter, this function prevents against damage of the filter and loss of product.*

Watch_TMP in ultrafiltration methods

Each concentration block of ultrafiltration methods contains a ***Watch_TMP*** instruction with the corresponding variable ***Flux_Drop_Enable_TMP***, see below.



When the transmembrane pressure (TMP) exceeds the ***Flux_Drop_Enable_TMP_FB***, the condition is met. The default action is to execute the instruction ***Flux_Drop_Enable***. The instruction ***% Flux_Drop_On*** is executed and latches on to the current flux value.

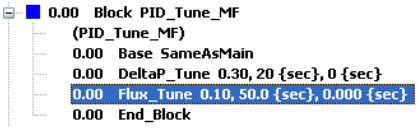
Watch_FluxDrop Greater_Than takes action if the set ***Flux_Drop_Max*** drop occurs. The default drop is 70%. The default action is that the permeate valve closes, the system is put into ***Hold*** state and a message to take action is displayed. The user can choose other actions to take place, for example execution of the instruction ***End_Block***.

Watch_ FluxDrop can be active during each step or across the process. To keep ***Watch_ FluxDrop*** active, remove ***%Flux_Drop_Off*** from block ***Restore_Conc***, and remove ***Watch_Off FluxDrop*** from block ***Watch_Off_Product***.

Note: *In cases of excessive fouling of the filter, this function prevents against damage of the filter and loss of product.*

PID instruction Flux_Tune

In product methods for microfiltration hollow fiber, the block ***INITIAL SETUP_MF_HF*** includes the block ***PID_Tune_MF*** with the ***Flux_Tune*** instruction, see illustration below.



The ***Flux_Tune*** instruction must be updated to ensure minimal oscillation when using flux for permeate control. The updated parameters should be used to baseline the process. Any update to the values requires testing for the specific process conditions including membrane surface area and sample.

Examples

The illustrations below show the result of default and updated ***Flux_Tune*** parameters.

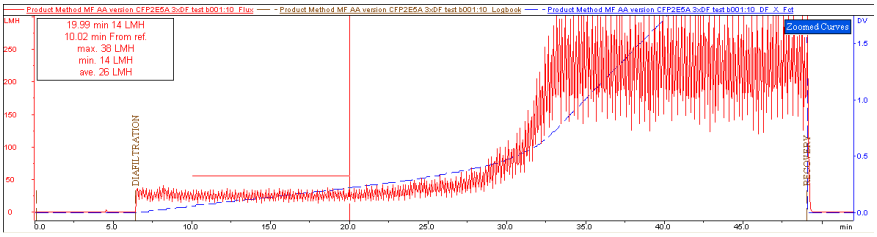


Figure 4.1: Method with default Flux_Tune parameters. A fouling event occurs due to heavy oscillation around the flux set point.

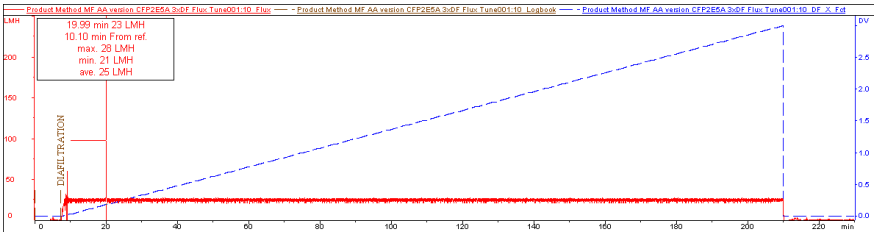


Figure 4.2: Method with updated Flux_Tune parameters (P=0.2 and I=50). The oscillation around the flux set point is less varied and no fouling occurs. The process runs to completion.

RECOVERY

The default ***PRODUCT*** and ***PROCESS OPTIMIZATION*** method templates contain a recovery block. The block is used to recover the final product from the tank and retentate flow path. The available blocks are:

- ***RECOVERY*** (default)

- ***NO RECOVERY***

The choices of sub blocks in ***RECOVERY*** are:

- ***Recovery_No_Buffer_Flush***
- ***Recovery_One_Buffer_Flush***
- ***Recovery_Two_Buffer_Flushes*** (default)

By default the air flow meter is used to push product out of the retentate flow path using block ***Air_Purge_Retentate***. If the final product is sensitive to air, the air purge may be replaced by block ***Recovery_Chase***. This block uses buffer to recover final product. Note that some dilution of the final product will occur if block ***Recovery_Chase*** is used.

For microfiltration applications, an additional recovery option, ***Recovery_Filter_Drain***, is included. This can be used when the target is in the permeate. The block ***Recovery_Filter_Drain*** allows liquid in the filter permeate chamber to be recovered via the filter drain. This block allows for drainage by gravity for 0.5 minutes.

Recovery_Flush blocks recirculate the buffer and then drains via the filter drain valve.

PROCESS OPTIMIZATION

PROCESS OPTIMIZATION methods are available for UniFlux 10 ultrafiltration (UF) applications using flat sheet cassette (FS) or hollow fiber (HF). The methods are used to determine the optimal TMP setpoint relative to sample, membrane surface area, and retentate flow rate. Plot the data in the evaluation wizard for ***Flux*** vs. ***TMP*** curves. For intensive process optimization studies consider ÄKTAcrossflow system.

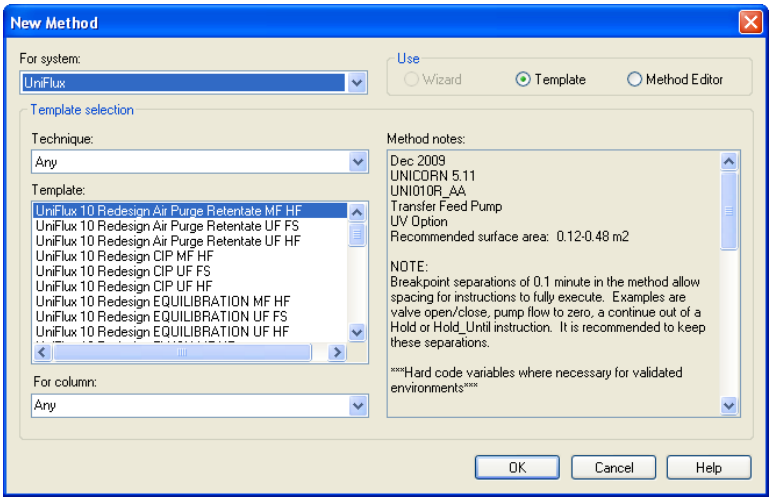
5 Edit methods

In this chapter

5.1 Working with method templates

Open a new template

Follow the instruction below to open a new template.

Step	Action
1	Open UNICORN ***Method Editor***.
2	In ***Method Editor*** module, click the ***New*** icon.  *Result:* The ***New Method*** dialog opens. A list of templates appear on the left, and the ***Method Notes*** appear on the right. 
3	In the ***New Method*** dialog: **a.** Select the ***Template*** radio button in the ***Use*** field. **b.** Select a method template from the ***Template*** list. **c.** Click the ***OK*** button.

Method Editor modes

The ***Method Editor*** interface operates in two modes. Select mode by clicking the corresponding icon.

- ***Text Instructions*** editor for entering and editing method instructions

- ***Run Setup*** for defining method properties and access method information

Select Method editor mode

If you want to open...	then...
Text Instructions	• click the ***Text Instructions*** icon,  *or* • choose ***View*** →***Text Instructions***.
Run Setup	• click the ***Run Setup*** icon,  *or* • choose ***View*** →***Run Setup***.

Tabs in Run Setup

The table below contains brief descriptions of some of the tabs of ***Run Setup***. For a detailed information, refer to *UNICORN User Reference Manual*.

Tab	Description
Start Protocol	Shows which items of the ***Run Setup*** that are displayed at the start of the run. Each template has a default ***Start Protocol***.
Questions	Displays the questions used in the method. ***Questions*** provide a means for entering run-specific information at the start of a run. For each template, default questions are included as a guide.
Variables	Lists all variables used in the method with their default values, organized by method block.
Notes	Shows the descriptive comments that form a part of the method documentation. For each template, the tab ***Method Notes*** includes a full operational description.

Update tank volume

In the method, default tank level for UniFlux 10 is 2 liters and default tank level for UniFlux 30, 120, and 400 is 20 liters. Default consumption for UniFlux 10 is 20 liters. However, tanks for UniFlux systems 30, 120, and 400 are supplied separately and are available in different volumes. Therefore the working tank volumes and consumption must be determined empirically for each configuration. Once determined, update ***TankLevel_Alarm*** in the ***Text Instructions*** box, and the other ***TankLevel*** parameters in the ***Text Instructions*** box or on the ***Variables*** tab of ***Run Setup***.

The table below lists the instructions that need to be updated based on the current tank volume.

Instruction	Block	Product template	Non product template
TankLevel_Alarm	***Alarms***, ***Tank_Fill***, ***Tank_Drain***	YES	YES
TankLevel	***Tank_Fill***	NO	YES
TankLevel (Product Start)	***FILL_SAMPLE***	YES	NO
TankLevel (Recovery Flush 1, 2)	***Recovery_Flush***	YES	NO
Hold_Until TankLevel	***Tank_Fill***, ***Tank_Drain***	YES	YES

Syntax errors

Syntax errors in a method are indicated by the symbols listed in the table below.

Marking	Description
Blue square with a red cross	Blocks that call an instruction that contains one or more invalid instructions.

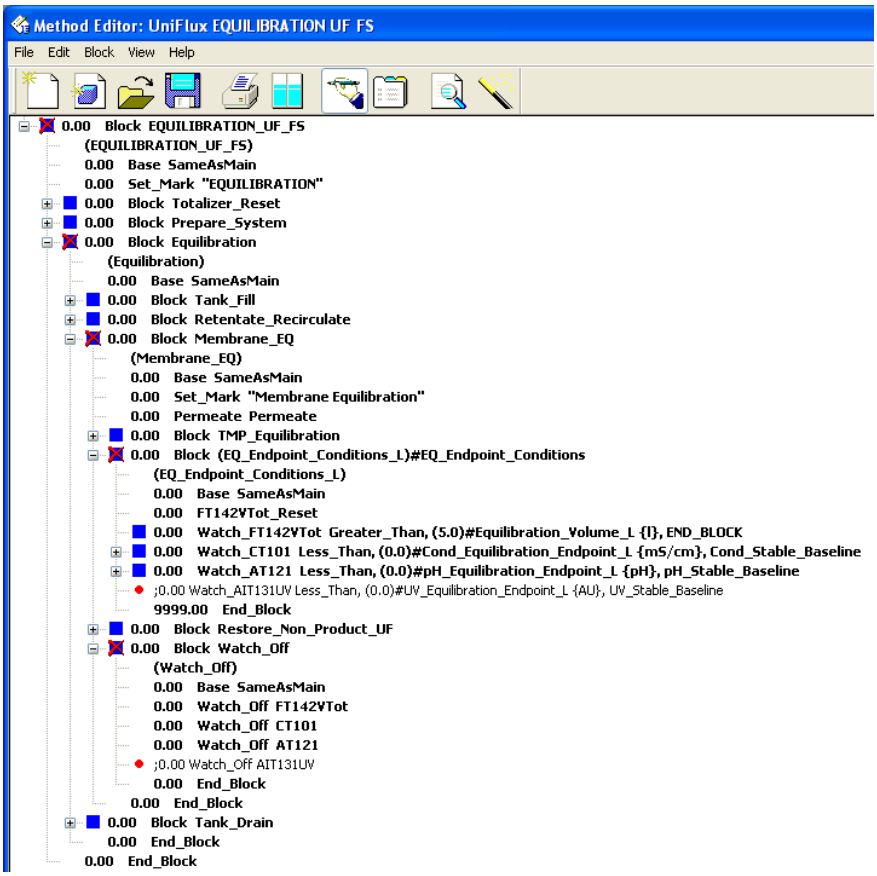
Marking	Description
Red bullet •	Instructions with invalid syntax. All such instructions must be deleted or changed before a method can be run. The instructions may be of the following types: • Calls to blocks which are not defined in the method • Instructions that apply to a different system strategy (can occur if a method is written for one system and saved for another) • Instructions for components that have not been selected in the ***System Setup***

Note: *A method cannot be run if it contains syntax errors. A message* `Unable to send run command` *will appear in* ***System Control*** *screen.*

Tip: *Check blocks in* ***<Unused>*** *for syntax errors.*

Example of syntax error

The illustration below shows an equilibration method for a UniFlux system without UV. Syntax errors due to lack of this component are represented by a red bullet for the line of text affected. These two lines need to be deleted in order to correct the syntax error.


Method Editor: UniFlux EQUILIBRATION UF FS
File Edit Block View Help
0.00 Block EQUILIBRATION_UF_FS
(EQUILIBRATION_UF_FS)
0.00 Base SameAsMain
0.00 Set_Mark "EQUILIBRATION"
0.00 Block Totalizer_Reset
0.00 Block Prepare_System
0.00 Block Equilibration
(Equilibration)
0.00 Base SameAsMain
0.00 Block Tank_Fill
0.00 Block Retentate_Recirculate
0.00 Block Membrane_EQ
(Membrane_EQ)
0.00 Base SameAsMain
0.00 Set_Mark "Membrane Equilibration"
0.00 Permeate Permeate
0.00 Block TMP_Equilibration
0.00 Block (EQ_Endpoint_Conditions_L)#EQ_Endpoint_Conditions
(EQ_Endpoint_Conditions_L)
0.00 Base SameAsMain
0.00 FT142¥Tot_Reset
0.00 Watch_FT142¥Tot Greater_Than, (5.0)#Equilibration_¥olume_L {l}, END_BLOCK
0.00 Watch_CT101 Less_Than, (0.0)#Cond_Equilibration_Endpoint_L {mS/cm}, Cond_Stable_Baseline
0.00 Watch_AT121 Less_Than, (0.0)#pH_Equilibration_Endpoint_L {pH}, pH_Stable_Baseline
;0.00 Watch_AIT131UV Less_Than, (0.0)#UV_Equilibration_Endpoint_L {AU}, UV_Stable_Baseline
9999.00 End_Block
0.00 Block Restore_Non_Product_UF
0.00 Block Watch_Off
(Watch_Off)
0.00 Base SameAsMain
0.00 Watch_Off FT142¥Tot
0.00 Watch_Off CT101
0.00 Watch_Off AT121
;0.00 Watch_Off AIT131UV
0.00 End_Block
0.00 End_Block
0.00 Block Tank_Drain
0.00 End_Block
0.00 End_Block

5.2 Working with blocks

Blocks

Methods in UNICORN are usually divided into blocks. Blocks typically represent well-defined steps in the filtration procedure, such as:

- ***Membrane Settings***
- ***FILL SAMPLE***
- ***CONCENTRATION FED BATCH***
- ***DIAFILTRATION***
- ***RECOVERY***

Each block in a method consists of a series of instructions, which request specific operations in the system. Every block must start with the ***Base*** instruction and end with the ***End_Block*** instruction, with all specific instructions in between. The instructions are executed in the order they are written until the block is finished by the ***End_Block*** instruction. At that time the method will proceed to the next block.

In the ***Text*** pane of ***Text Instructions***, blocks are marked with a blue square. Main blocks are always in all capital letters. Sub blocks are always first letter cap followed by lower case.

Two ways to add new blocks

You can add blocks to a method in two ways, using either

- the ***Instruction box*** of the ***Text Instructions*** editor,

or

- the ***New Block*** dialog box reached via the ***New Block*** icon.

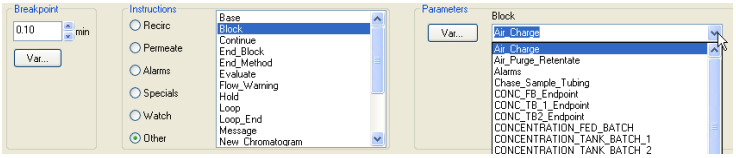
Both alternatives are described below.

Add blocks with the Instruction box

Follow the instruction below to add blocks with the ***Instruction box***.

Step	Action
1	In the ***Text*** pane of the ***Text Instructions*** editor, select the instruction or block that you want to precede the new block.
2	In the ***Instruction*** box, select ***Other*** →***Block***.
3	If you want to add a block:

Step	Action
	a. Enter a name for a new block in the ***BlockBlock*** field, or select the block to insert in the drop-down menu.



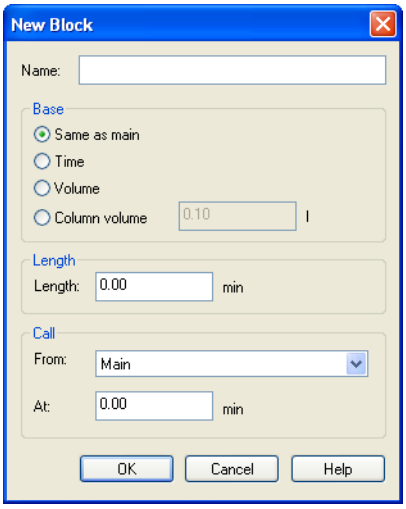
	b. Click the ***Insert*** button.
	Note:
	*A new block creates the framework for the block. The **Base** and **End_Block** instructions already exist, other instructions must be added by the user.*
	Result:
	The block is inserted after the block that was selected in step 1.

Add blocks with the New Block dialog box

Follow the instruction below to add blocks with the menu options of the ***New Block*** dialog box.

Step	Action
1	• Choose ***Block*** →***New*** in the ***Method Editor*** module, or • click the ***New Block*** icon.
	
	Result: The ***New Block*** dialog box is displayed.

Step	Action



2 Enter the relevant information in the ***New Block*** dialog box, and click ***OK***.

Result:

The new block is added to the method, and placed last of all blocks.

Tip:

The block can be placed in other positions by selecting something other than ***Main*** *in the* ***From*** *drop-down list.*

Tip:

For blocks at the same breakpoint, click and drag the block to place it in the proper sequence.

Import blocks

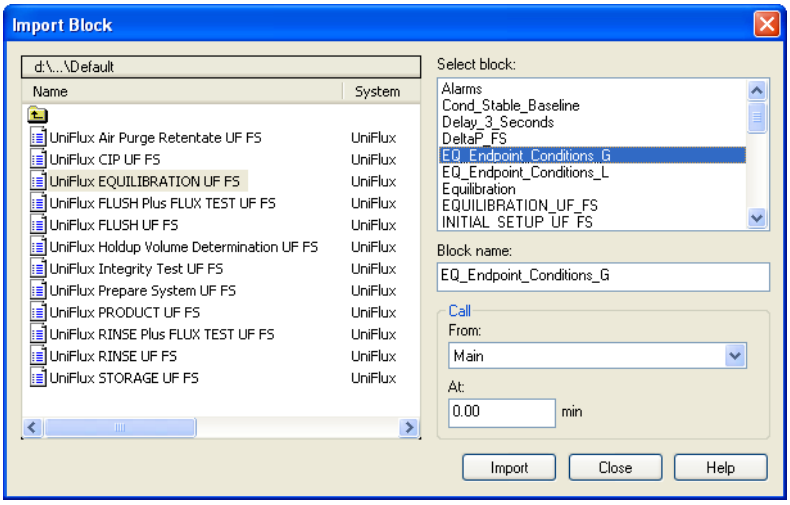
Follow the instruction below to import method blocks.

Step	Action
1	Choose ***Block*** →***Import Block As*** in the ***Method Editor***.

Step	Action

Result:

The ***Import Block*** dialog is displayed.



2

a. Select the method from which you want to import a block.

b. Select the block to import.

Result:

The name of the selected block is displayed in the ***Block name*** field.

3

In the ***Call*** field, do the following:

a. On the ***From*** drop-down list, select a block into which the block will be imported.

b. In the ***At*** field, select the breakpoint value for the block to be imported.

4

Click the ***Import*** button.

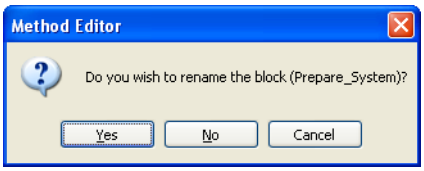
Result:

If the selected block contains sub blocks, the user will be asked if the sub blocks are to be imported as well. If imported sub block names already exist in the current method, the imported blocks will be renamed to 'Renamed0' with an ascending number for each additional block. Rename the blocks after the import is completed. After renaming, clean up renamed blocks in unused. Use the edit function to search for 'Renamed' to ensure all are deleted.

Step	Action
	Note: *The imported block cannot have the same name as an existing block in the method. If the default name is not allowed for this reason, the* ***Import*** *button will be gray and locked. If this occurs, change the name of the imported block in the* ***Block name*** *field so that the* ***Import*** *button becomes available.* ***Tip:*** *When importing multiple blocks it may be easier to import them all into* ***<Unused>*** *and insert them into the desired locations from the* ***Text Instructions*** *box, see* *Add blocks with the Instruction box, on page 32**.*
5	**a.** Repeat steps 2 and 3 if needed. **b.** Click the ***Close*** button.

Copy blocks from within an opened method

Follow the instruction below to copy blocks from within an open method.

Step	Action
1	**a.** Right-click the block you want to copy. **b.** Choose ***Copy***.
2	**a.** Right-click the instruction line just above the point where you want the block to be pasted. **b.** Choose ***Paste***. *Result:* A dialog box asks if you wish to rename the pasted block. 
3	Choose from the following options: • Click ***No***. *Result:*

Step	Action
	If the block to be copied contains sub blocks, the user will also be prompted with the choice to rename the sub blocks. The block is then inserted with the same breakpoint value as the block or instruction selected for point of insertion. The new block will be identical to the copied block. All changes made to either blocks will change both blocks. • Click ***Yes***. *Result:* The user is asked to enter a new block name in the ***Block Name Definition*** dialog that opens. The new block is then inserted in the method with the same breakpoint value as the block or instruction selected for point of insertion. The new block has the same instructions as the copied block but has the new block name and is not linked to the copied block.

Move a block within a method

Follow the instruction below to move a block within a method.

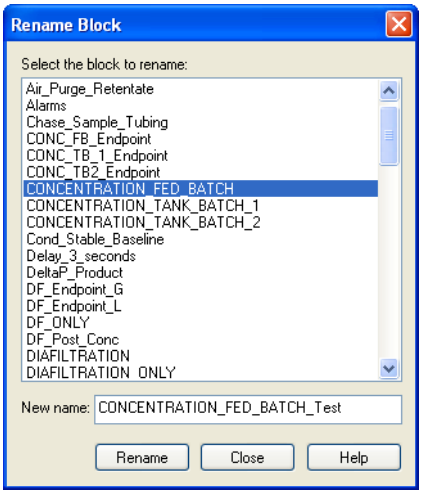
Step	Action
1	**a.** Right-click the block you want to move. **b.** Choose ***Cut***.
2	**a.** Right-click the instruction line just above the point where you want the block to be pasted. **b.** Choose ***Paste***. *Result:* The block is now removed from its original breakpoint and pasted at the new breakpoint. The pasted block is inserted with the same breakpoint value as the block or instruction selected for point of insertion.

Rename blocks

Follow the instruction below to rename blocks.

Step	Action
1	• Choose ***Block*** →***Rename Block*** in the ***Method Editor*** module, or • right-click the block you want to rename in the ***Text*** pane and select ***Rename***. *Result:* The ***Rename Block*** dialog box is displayed.

Step	Action



Note:

*By default, the block that is currently selected in the **Text** pane is automatically selected in the dialog.*

2 Enter the new name in the ***New name*** field and click ***Rename***.

3 **a.** If needed, repeat step 2 for other blocks.

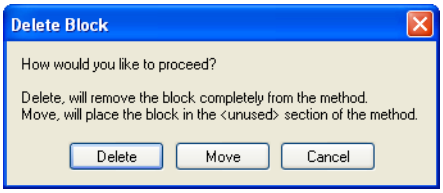
b. Click ***Close***.

Note: If the block you renamed is called in a ***Block*** or ***Watch*** instruction, the block name in these instructions will be changed automatically.

Delete or move blocks from a method

Follow the instruction below to delete blocks completely from the method, or to place blocks in the ***<Unused>*** section of the method.

Step	Action
1	Right-click a block and choose ***Delete*** from the shortcut menu.

Step	Action
	Result: The ***Delete Block*** dialog opens. 
2	Choose from the following options: • Click ***Delete***. *Result:* The block is totally removed from the method. If the block is called several times in the method, all the blocks will be deleted. Blocks deleted in this fashion cannot be called again in the method. ***Note:*** *If the block contains sub-blocks, another dialog box is displayed, asking you if you want to delete the sub-blocks as well.* • Click ***Move***. *Result:* The block is deleted from the method and transferred to the ***<Unused>*** section of the method. If the block is called several times in the method, however, only the row with the block currently marked in the ***Text*** pane will be deleted. In this case, the block will not be placed in the ***<Unused>*** section (since the block is still used in the method). Blocks deleted in this fashion can be called again in the method.

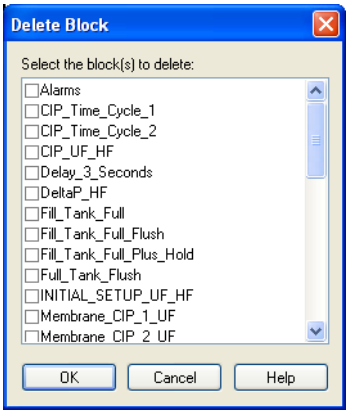
Note: *Any block or watch instruction that calls a deleted block remains in the method, but is shown with a red bullet beside the instruction line to indicate that it is no longer valid. A method cannot be run if it contains invalid instructions, even if they are located in* ***<Unused>*** *.*

Delete blocks using the Delete Block command

Follow the instruction below to delete a block using the ***Block*** →***Delete Block*** command:

Step	Action
1	Select the menu command ***Block*** →***Delete Block*** in the ***Method Editor***.

Step	Action
	Result: The ***Delete Block*** dialog box is displayed with all blocks listed in alphabetical order.



2	Select the blocks you want to delete and click ***OK***.
3	Click ***Yes*** to confirm. *Result:* The selected blocks are deleted from the method.

Note: *Any block or watch instruction that calls a deleted block remains in the method, but is shown with a red bullet beside the instruction line to indicate that it is no longer valid. A method cannot be run if it contains invalid instructions, even if they are located in* ***<Unused>*** *.*

Restore deleted blocks

If a block is deleted by mistake, it can be restored by importing it from the saved version of the method provided that the method with the block deleted has not been saved. For instructions, see *Import blocks, on page 34*.

5.3 Working with text instructions

Base instruction

Every method block must start with a ***Base*** instruction, defining the base for calculating breakpoints. Different blocks in the same method may be written on different bases.

Method blocks for filtration are written in one of two bases:

- time in minutes (min)
- volume in liters (l)

Most blocks for UniFlux are in base time.

Tip: *For all blocks other than the main block, the base may also be defined as* ***SameAsMain****, which means that the block will inherit the base defined in the main block.*

What base should I use?

Use the base which most closely suits the purpose of the block. Time is recommended as the base for most steps in a run. In some situations, it may be more suitable to use volume base (filter equilibration volume) for individual blocks.

Note: *Be careful when changing the base for an existing method. Changing between time and volume base can affect the relative duration of steps in the method if different steps use different flow rates. In some cases a change to volume base may stop the progress of the method if there is no active flow.*

Change an instruction

Follow the instruction below to change an instruction in the ***Text*** pane ***Text Instructions***.

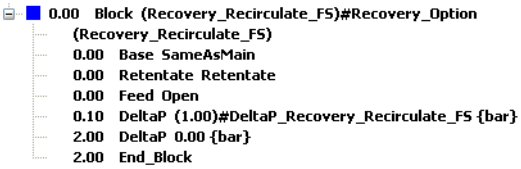
Step	Action
1	Select the instruction. *Result:* The instruction with its current parameters is displayed in the ***Instruction box***.
2	Make the required changes to the breakpoint or parameters, *or* select a new instruction in the ***Instruction Box***.
3	Click the ***Change*** button, *or* the ***Replace*** button.

Step	Action
	Note: *The **Change** and **Replace** buttons are equivalent unless changes are made to a breakpoint. See below.*

Breakpoints

Each instruction in a method block is issued at a specified breakpoint according to the base instruction. The first instruction in a block is always at breakpoint 0.00, and all other breakpoints are counted from this point.

For example, the ***DeltaP*** instruction below will set the ***DeltaP*** to 1.00 bar at 0.1 minutes after the start of the block. At 2.00 minutes the ***DeltaP*** will be set to 0.00 bar (turned off).



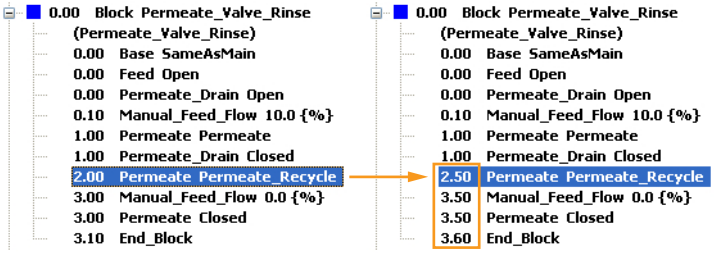
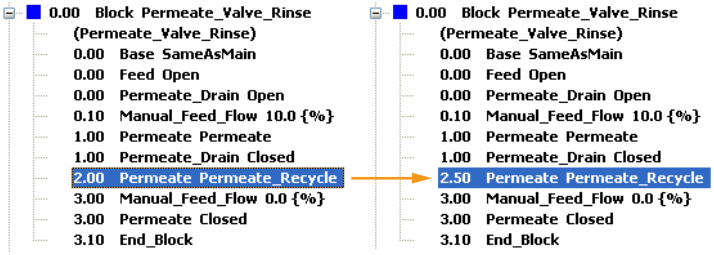
Breakpoint separations

Breakpoint separations of 0.1 minute in the method allow spacing for instructions to fully execute. Examples are valve open/close, pump flow to zero, a continue out of a ***Hold*** or ***Hold_Until*** instruction. It is recommended to keep these separations. In the following cases, breakpoint separations are required:

- A set of instructions does not execute properly all the time – intermittent failure.
- An instruction is not executed; it is "passed over".

Update breakpoints

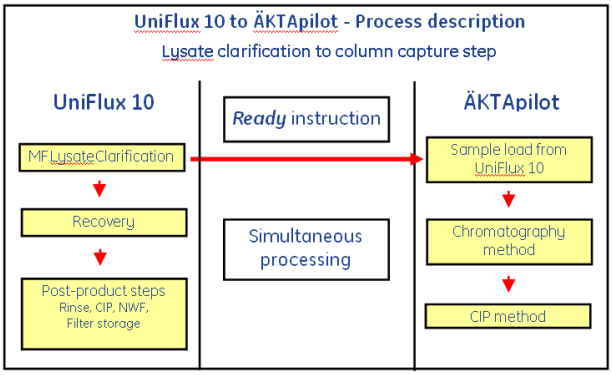
The table below describes the difference in function between the ***Change*** button and the ***Replace*** button when you change breakpoints.

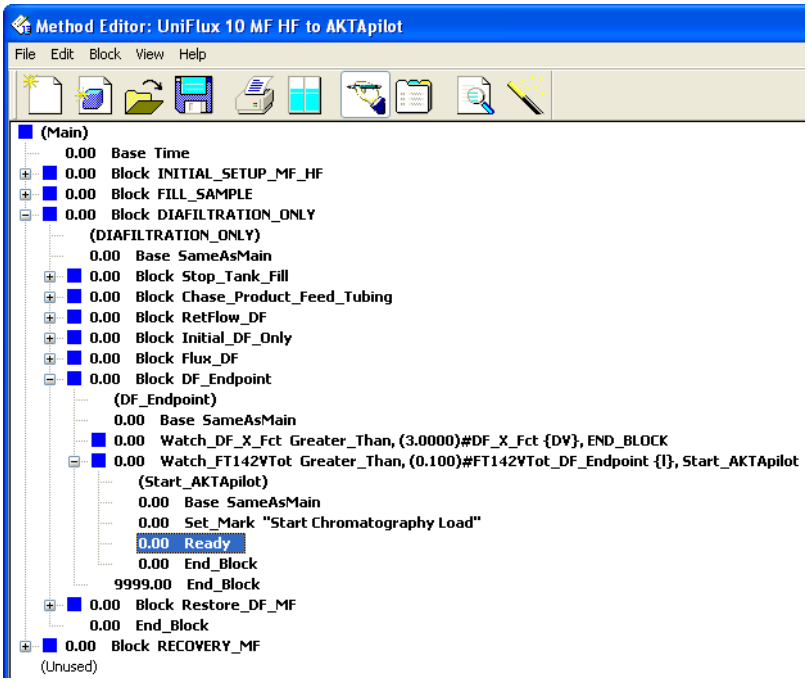
Button	Function
Change	This button shifts all subsequent instructions in the block according to the change in the breakpoint. ***Change*** does not affect the relative order of instructions in the method. You cannot change the breakpoint of an instruction to earlier than the nearest previous breakpoint in a block. The illustration below shows an example where instruction ***Permeate Permeate_Recycle*** is changed from breakpoint 2.00 to 2.50 minutes using the ***Change*** button. All instructions following this are also updated by 0.50 minutes. 
Replace	This button moves the selected instruction but does not change the breakpoint of any other instruction. ***Replace*** can change the relative order of instructions in the method. The illustration below shows an example where instruction ***Permeate Permeate_Recycle*** is changed from breakpoint 2.00 to 2.50 minutes using the ***Replace*** button. No other instruction is affected by this action. 

Ready instruction

The ***Ready*** instruction signals that a next step in a ***MethodQueue*** may start, provided that the next step runs on a different idle wet unit. This instruction provides more efficient method linking allowing both units to be in a run mode simultaneously. This functionality can be used to protect sensitive proteins across unit operations.

In the example below lysate clarification on UniFlux 10 is followed by column capture steps on ÄKTApilot.





5.4 Working with variables

Variables

All method templates have variables. Each variable indicates an option for user choice. Variables allow one method to be used for runs under a variety of conditions, such as varying load volumes. Variables can be assigned to most instruction parameters including breakpoints. Some variables are block variables, that is, the user selects the block to be included in the method from a number of available blocks.

Note: *In validated environments the use of variables may be minimized or eliminated once the final method has been verified*

Two types of variables

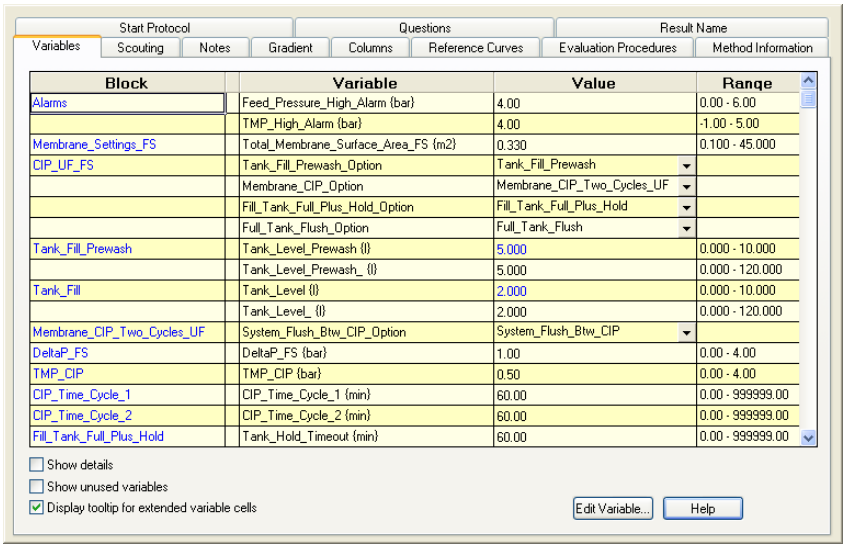
There are two types of variables that can be assigned to instructions, see the table below.

Variable type	Description
Parameter	Defines the function of the instruction such as valve positions, retentate flow rate, TMP, recovery volume, or the block to be executed.
Breakpoint	Defines when the instruction occurs and how long it lasts.

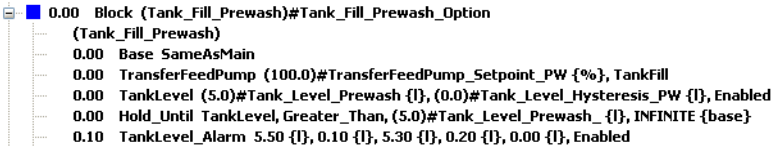
Identify variables

Variables in a method can be identified in several ways, see below.

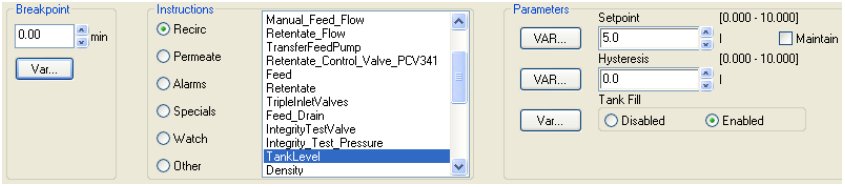
- All variables are listed on the ***Variables*** tab of the ***Run Setup***, grouped according to the block in which they appear.



- In the ***Text*** pane in ***Text Instructions***, the current value is given in parentheses followed by the variable name that is preceded by "#" sign.



- In the ***Instructions*** field of the ***Instruction box***, the ***VAR*** button beside the parameter field is displayed in capital letters, that is, ***VAR*** not ***Var.***



Change variable values

To change default variable values, you can either

- edit the instruction or block in the ***Instruction box*** of ***Text Instructions***

or

- change the value in the ***Variables*** tab of ***Run Setup***

or

- change the value in the ***Variables*** dialog of the ***System Control*** module immediately before the start of a run without using the ***Method editor***

Variable names

Variables are defined with names, which can be explicit descriptions of the variable function, for example, ***Total_Membrane_Surface_Area_FS*** (FS = Flat Sheet cassette). Suitable choice of variable names can make the method easier to read and understand, and also help the operator in setting variable values at the start of a run.

The variable names can be up to 32 characters long and the following characters can be used:

- Letters (A-Z)
- Digits (0-9)
- The underscore character (_)

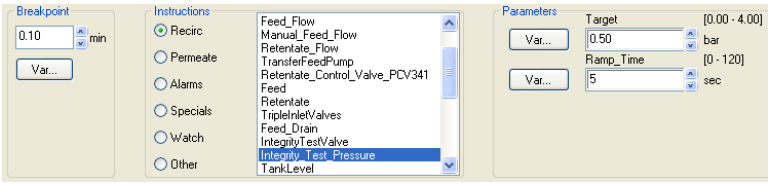
The case of letters is retained, but not significant. The names ***Flow_Rate*** and ***FLOW_RATE*** are treated as identical.

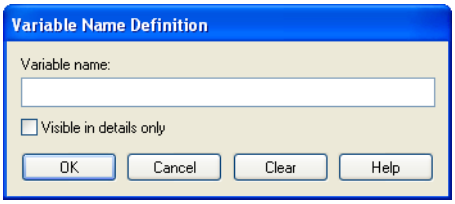
Define variables

Only one variable that affects block length may be defined within each block. However, any number of parameters may be defined as variables within a block, up to a total of 64 variables for the entire method. Each parameter defined as a variable is also assigned a default value, which is used if no changes are made to variable values at the start of a run.

Follow the instruction below to define a new variable.

Step	Action
1	Select the instruction where you want to define the variable in the ***Text*** pane of ***Text Instructions***. *Result:* The parameters for the instruction are shown in the ***Instruction box***.
2	**a.** Locate the breakpoint or the parameter in the ***Instruction box***. **b.** Click the ***Var*** button. *Result:* The ***Variable Name Definition*** dialog opens.



Step	Action
	
3	In the ***Variable Name Definition*** dialog: **a.** Enter a variable name. **b.** Select the ***Visible in details only*** check box if you want to set the variable as a detailed variable. Detailed variables only become visible on the ***Variables*** tab if the ***Show details*** check box is selected. **c.** Click ***OK***. *Result:* The ***Var*** button changes to ***VAR*** to confirm the new variable. The variable is displayed in the ***Text*** pane.

5.5 Working with conditional instructions

Watch instructions

Watch instructions make it possible to perform a specified action when a particular monitor signal meets a given condition. There are ***Watch*** instructions for each process monitor signal, and each ***Watch*** instruction can use various conditions to respond to a signal value. Each ***Watch*** instruction must also have an action to perform once the condition is met, and a breakpoint.

In UniFlux method templates, ***Watch*** instructions are used in the ***EQUILIBRATION*** methods to end the filter equilibration step, and in the ***PRODUCT*** methods to end concentration and diafiltration steps.

When is a Watch instruction active?

The breakpoint when the ***Watch*** instruction is issued determines when the watch begins, not when the block is activated. Once set, a ***Watch*** instruction remains active until:

- the condition is met,

or

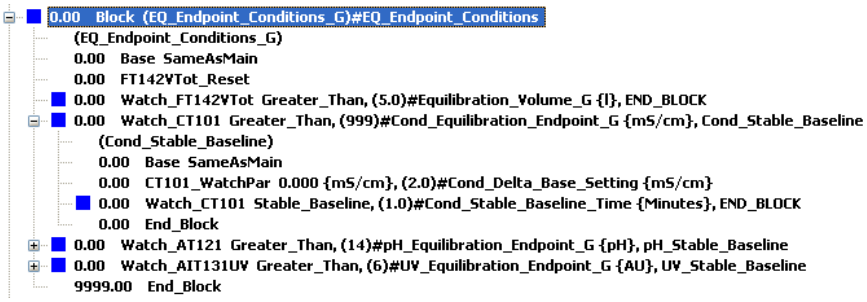
- a new ***Watch*** instruction is issued for the same monitor,

or

- the ***Watch*** is turned off by the ***Watch_off*** instruction.

Activation of Watch instructions

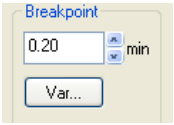
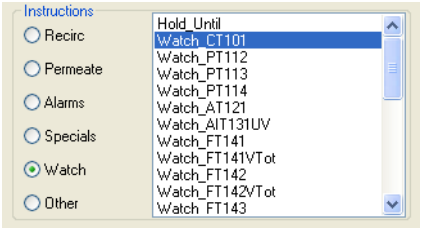
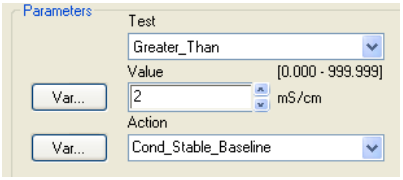
Once UNICORN executes the action of a ***Watch*** instruction, the method advances to the next line in the method. In the example below multiple ***Watch*** instructions are in the same block. They will be activated one after another. In order to keep the block ***EQ_Endpoint_Conditions_G*** from advancing the ***End_Block*** breakpoint is updated to 9999.00 minutes.



Insert a Watch instruction

Watch instructions are inserted in the ***Instruction box*** of the ***Text Instructions Editor***.

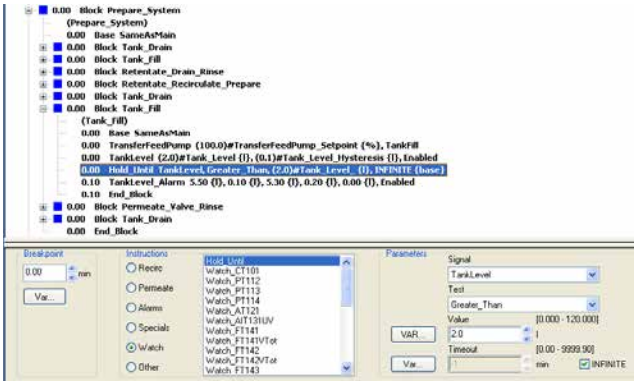
Follow the instruction below to insert a ***Watch*** instruction.

Step	Action
1	In the ***Breakpoint*** field, select the appropriate breakpoint. This determines when the watch begins. 
2	In the ***Instructions*** field: **a.** Select ***Watch***. **b.** Select a ***Watch*** instruction from the drop-down list 
3	In the ***Parameters*** field, select appropriate values under ***Test***, ***Value*** and ***Action***.  ***Note:*** *The **Test** parameters **Slope**, **Less Than** or **Valley**, and **Peak Max** are for chromatography systems. They are not used in UniFlux programming.*
4	Click ***Insert***. *Result:* The new ***Watch*** instruction is inserted into the method at the selected breakpoint.

Hold_Until instructions

The ***Hold_Until*** instruction is a special kind of ***Watch*** instruction. The method is put on hold until a specific condition is met (signal, test and value) or until the specified ***Timeout*** is reached. The method execution is then continued. Instructions at the same breakpoint as the ***Hold_Until*** instruction, but in the method placed after ***Hold_Until***, will be executed after the method execution is continued. It is recommended to space instructions after ***Hold_Until*** by 0.10 minutes to ensure robust execution every time.

In UniFlux method templates ***Hold_Until*** is used in block ***Tank_Fill***. The goal is to hold the method until the tank is filled with the volume set point. In the example below the ***Hold_Until*** instruction is used with ***Timeout*** equals ***INFINITE***.



Watch for stable baseline

The condition ***Watch Stable_Baseline*** is met if the signal does not deviate by more than ± ***BaseLine***/ ***Delta_Base*** (see below) from the baseline during the time interval specified for the watch. The baseline value is determined by the signal at the start of the watch. When the condition is met the action calls for the block to end immediately and continue with the remainder of the program. If the condition is not met, a new interval is started with a new baseline value defined by the signal level at the start of the new interval.

Default parameters for Watch Stable_Baseline

Default parameters for ***Watch Stable_Baseline*** are:

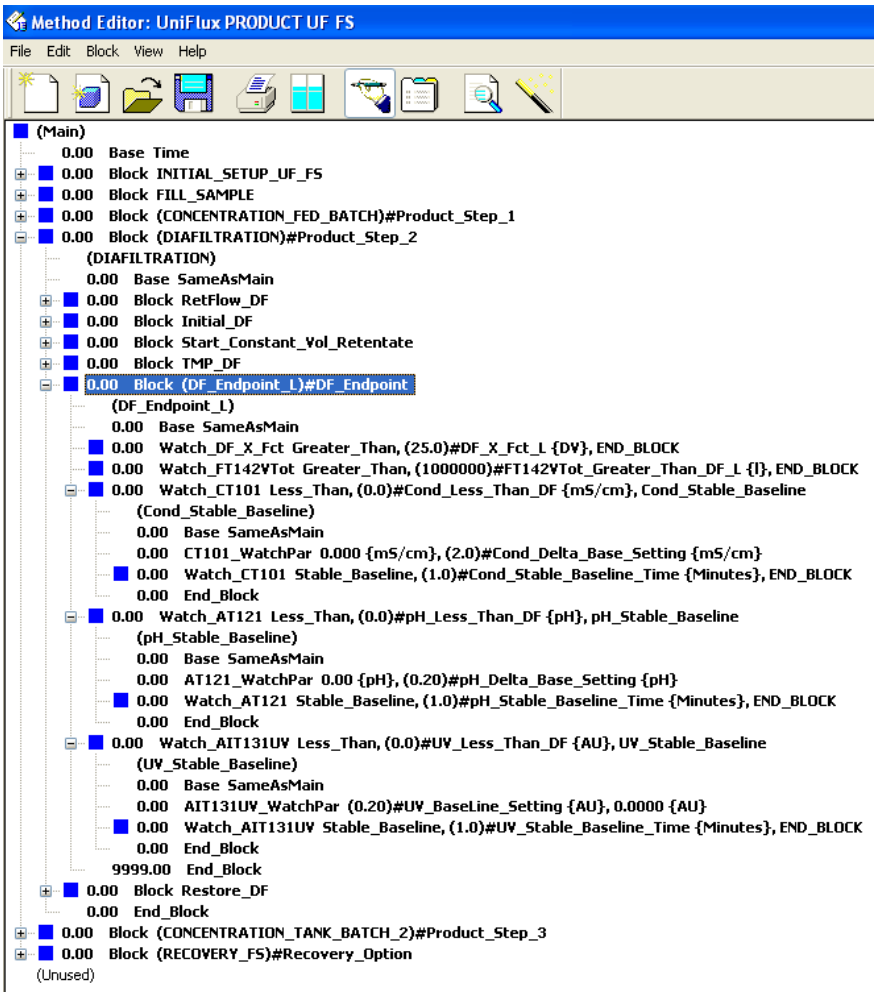
Parameter	Default value
Duration	1 minute
BaseLine/ ***Delta_Base*** for conductivity	± 2 mS/cm
BaseLine/ ***Delta_Base*** for pH	± 0.20 pH units
BaseLine/ ***Delta_Base*** for UV	± 0.20 OD

Example of Watch for stable baseline

UniFlux ***PRODUCT*** method for diafiltration includes ***Endpoint*** blocks with a series of ***Watch*** instructions. Whichever occurs first will end the current ***Product_Step***. The ***Watch*** conditions will set a threshold greater than or less than a set point. Once a ***Watch*** condition occurs, it is automatically followed by a ***Watch Stable Baseline*** instruction. Once the signal meets the criteria and is considered stable, the resulting action is to end the block. This will then end the ***DIAFILTRATION*** product step.

Block ***DF_Endpoint_L*** contains five ***Watch*** instructions (see illustration below). One is based on the diafiltration factor (***DF_X_Fct***). A second is based on the permeate volume totalizer (***FT142VTot***). The remaining three are based on ***Watch Less Than*** followed by ***Stable Baseline***. All three permeate monitors – ***Cond***, ***pH***, and ***UV*** – are monitored. The default ***Watch Less Than*** values are all at zero. These conditions will never be met unless the user updates the parameters to a value that matches a standard procedure. Once the ***Watch Less Than*** condition is met, ***Watch Stable Baseline*** is active.

In order to keep the block ***DF_Endpoint_L*** from advancing the ***End_Block*** breakpoint is updated to 9999.00 minutes.

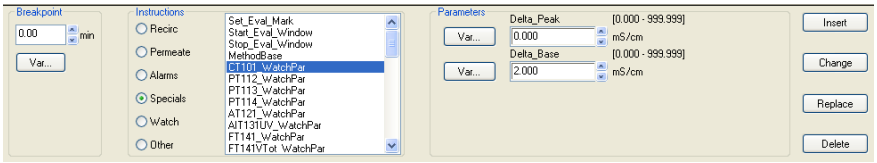


Set BaseLine/ Delta_Base

Note: *Recent strategies have updated the **BaseLine** instruction to **Delta_Base**. The function is the same.*

The parameter ***BaseLine*/ *Delta_Base*** defines the permitted variation for the ***Stable_Baseline*** condition, see *Watch for stable baseline, on page 51*). The ***BaseLine*** setting affects only the ***Stable_Baseline*** condition. ***BaseLine*** is found under the ***Specials*** instruction set. There is a ***WatchPar*** (watch parameter) for all monitors (Cond, pH, UV, Flow) as well as for other instructions.

An example is shown below.



Default parameters for BaseLine/ Delta_Base

The default parameters for ***Watch Stable_Baseline*** are:

Signal	Default *BaseLine/ Delta_Base*
Conductivity	± 2 mS/cm
pH	± 0.20 pH units
UV	± 0.20 AU

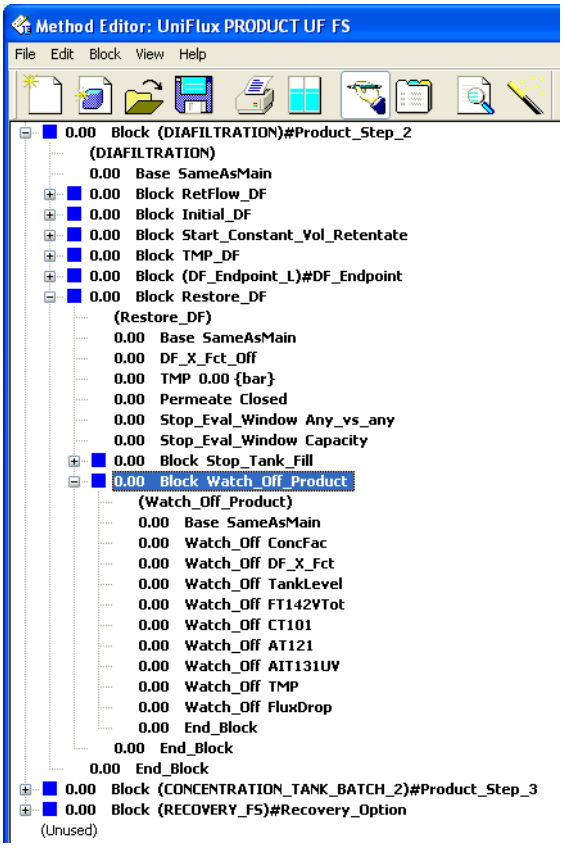
Watch Off

A ***Watch_Off*** should always be placed in the method just after the time in which the watch instruction is no longer needed. This is to prevent a ***Watch*** condition from occurring at the wrong time and thus performing the action at the wrong time.

Example of Watch Off

In UniFlux method templates, the block ***Watch_Off_Product*** includes all signals monitored during a concentration or diafiltration product step.

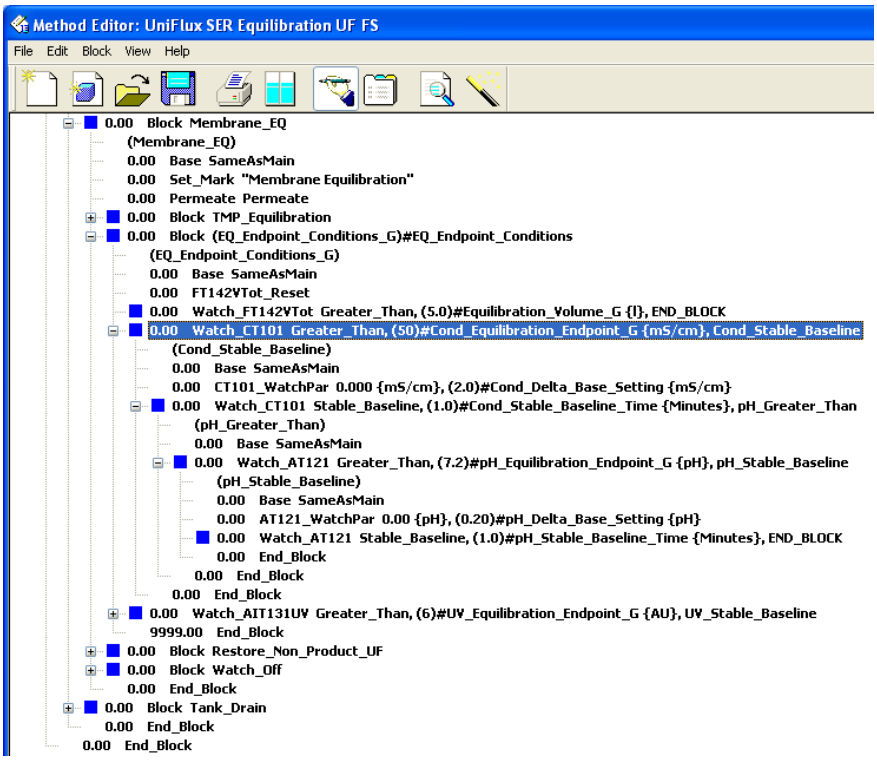
In the example below, the block ***Watch_Off_Product*** ensures that ***Watch*** conditions that are not met within the ***DIAFILTRATION*** block are inactivated before the next product step starts.



Nested Watch instructions

Inserting ***Watch*** instructions within the action block of another ***Watch*** instruction creates nested blocks. UNICORN will not read the second ***Watch*** instruction until after the condition of the first ***Watch*** instruction is met.

In the example below, UNICORN will not read the ***Watch_AT121*** instruction for a pH value greater than 7.2 before the conductivity has exceeded 50 mS/cm, and the conductivity baseline is stable.



Check the duration of a Watch instruction

In the ***Block*** pane of the ***Method Editor*** module the user can visualize the duration of ***Watch*** instructions in the method. Most often a ***Watch*** instruction is only intended to be active within a block or section of the method.

Example

The illustrations below show the ***Block*** pane of UniFlux ***EQUILIBRATION*** methods. The conditional instructions in block ***EQ_Endpoint_Conditions_L*** are shown in green text.

Fig. 5.1, on page 57 shows a method *without* the block ***Watch_Off***. The conditional instructions in block ***EQ_Endpoint_Conditions_L*** are active through the end of the method. This is not ideal.

Fig. 5.2, on page 57 shows a method *with* the block ***Watch_Off***. The conditional instructions in block ***EQ_Endpoint_Conditions_L*** are active only within this block, as intended.

Figure 5.1: EQUILIBRATION method without block Watch_Off

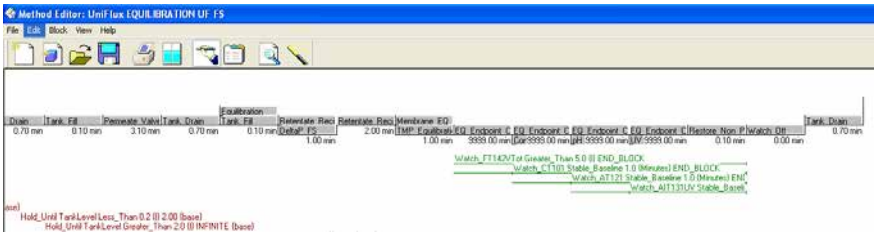


Figure 5.2: EQUILIBRATION method with block Watch_Off

5.6 Examples on updating method templates

This chapter gives example on how to update method templates.

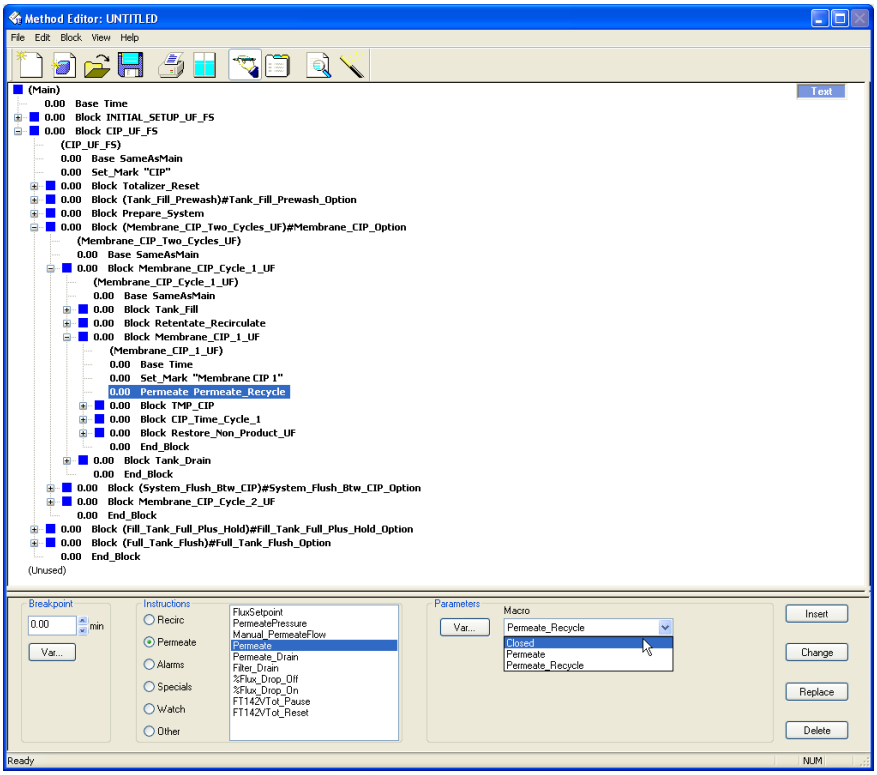
Update a CIP template

Goal

Two CIP cycles are to be used. The first cycle is for retentate recirculation only. The permeate valve shall be closed. In the second cycle the permeate shall be in recycle, as per default.

Action

In the ***Text*** pane, expand the ***Membrane_CIP_1_UF*** block. Highlight the instruction ***Permeate Permeate_Recycle***. In the ***Instruction box*** pane, choose ***Closed*** from the drop-down list. Click ***Replace***.



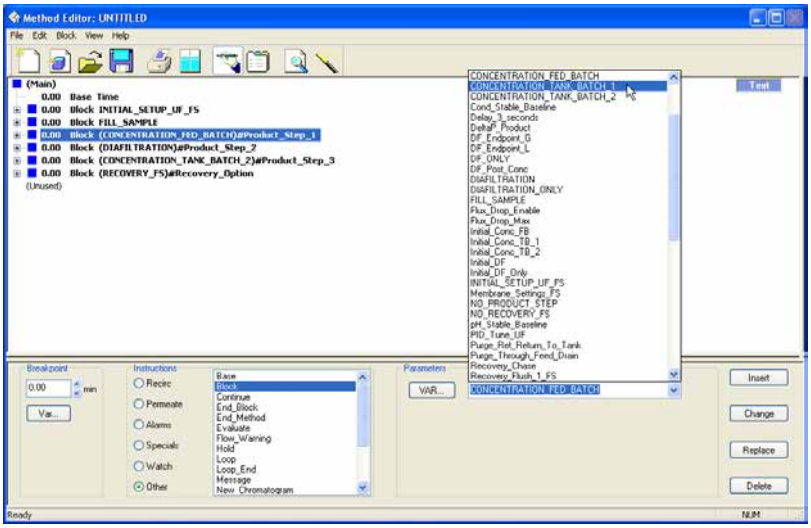
Update a PRODUCT template

Goal

The method shall contain two product steps. The first is ***CONCENTRATION_TANK_BATCH_1***. The second is ***DIAFILTRATION***. Recovery with one flush is to be used.

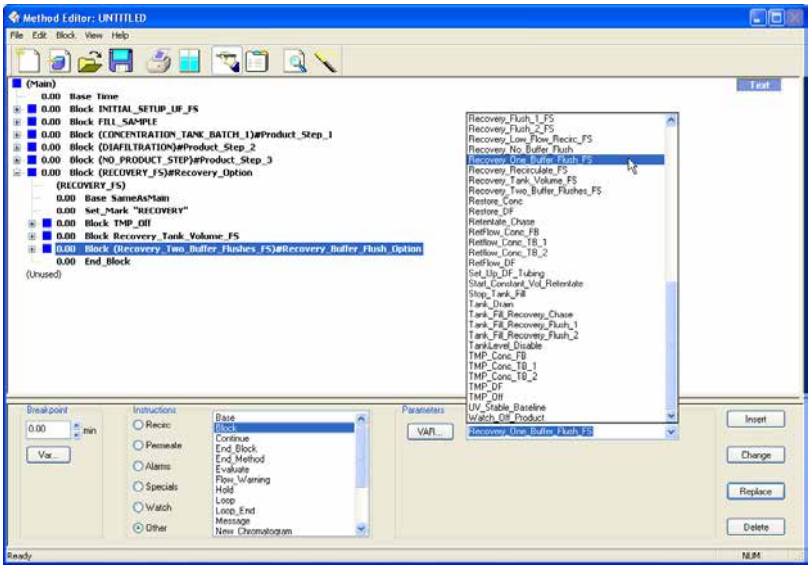
Action

- To update ***Product_Step_1***:
  1. In the ***Text*** pane, highlight ***Product_Step_1***.
  2. In the ***Instruction box*** pane, choose the ***CONCENTRATION_TANK_BATCH_1*** block from the drop-down list.



  3. Click ***Replace***.

- To remove ***Product_Step_3***:
  1. In the ***Text*** pane, highlight ***Product_Step_3***.
  2. In the ***Instruction box*** pane, choose the ***NO_PRODUCT_STEP*** block from the drop-down list and click ***Replace*** *or* click ***Delete***.
- To update the ***RECOVERY*** block:
  1. In the ***Text*** pane, expand the ***RECOVERY*** block.
  2. Highlight the block ***Recovery_Buffer_Flush_Option***.
  3. In the ***Instruction box*** pane, choose the ***Recovery_One_Buffer_Flush*** block from the drop-down list.



4. Click ***Replace***.

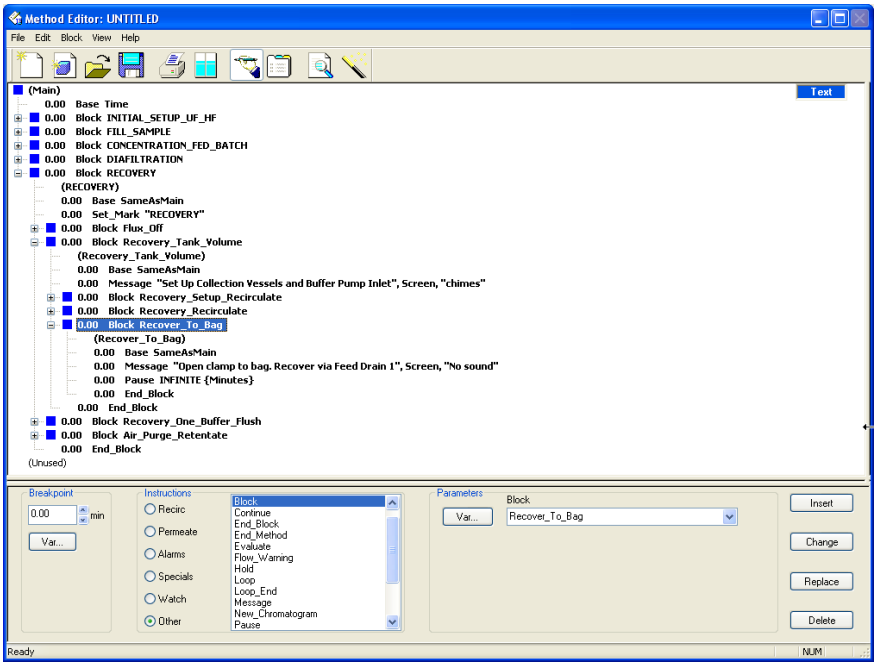
Update a RECOVERY block

Goal

The ***RECOVERY*** block shall be updated for collection of the product in a ReadyCircuit™ bag.

Action

Insert a new block to prompt the user to set up a bag for recovery via **Feed Drain**.



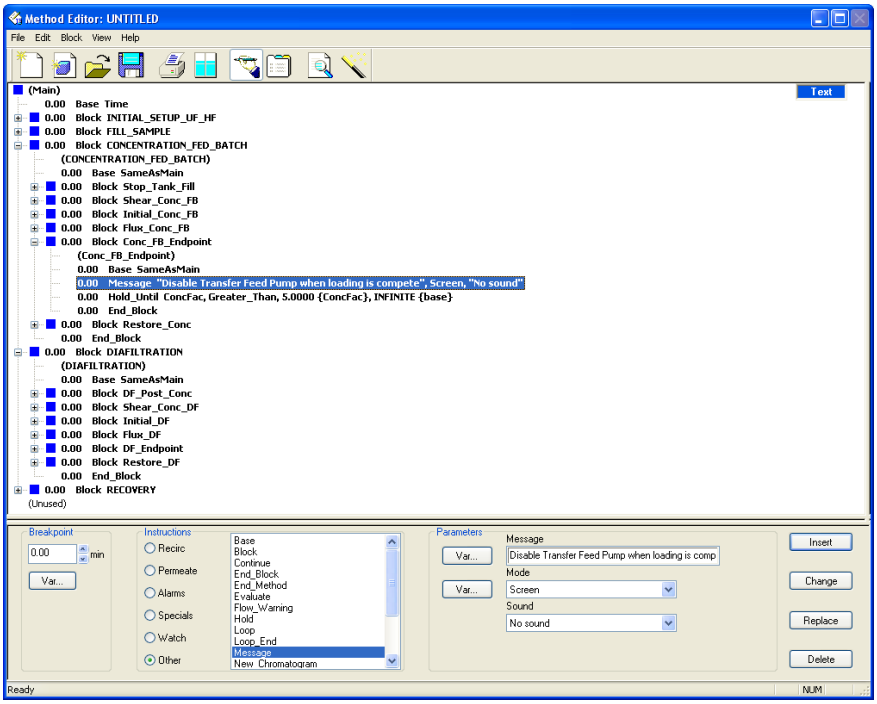
Product template without tank level control

Goal

Utilize the system automation despite the lack of a tank level input from an external scale. Use a single endpoint condition for the product steps.

Action

In the ***Text*** pane, update the method to prompt the user to monitor the tank level readout and take action. For the product step endpoint block, remove ***Watch*** instructions that are not required, replace the ***Watch*** instruction for ***ConcFac*** with a ***Hold_Until*** instruction, and update the breakpoint of the ***End_Block*** instruction to 0.00 minutes.


Method Editor: UNTITLED
File Edit Block View Help
Text
(Main)
0.00 Base Time
0.00 Block INITIAL_SETUP_UF_HF
0.00 Block FILL_SAMPLE
0.00 Block CONCENTRATION_FED_BATCH
(CONCENTRATION_FED_BATCH)
0.00 Base SameAsMain
0.00 Block Stop_Tank_Fill
0.00 Block Shear_Conc_FB
0.00 Block Initial_Conc_FB
0.00 Block Flux_Conc_FB
0.00 Block Conc_FB_Endpoint
(Conc_FB_Endpoint)
0.00 Base SameAsMain
0.00 Message "Disable Transfer Feed Pump when loading is compete", Screen, "No sound"
0.00 Hold_Until ConcFac, Greater_Than, 5.0000 {ConcFac}, INFINITE {base}
0.00 End_Block
0.00 Block Restore_Conc
0.00 End_Block
0.00 Block DIAFILTRATION
(DIAFILTRATION)
0.00 Base SameAsMain
0.00 Block DF_Post_Conc
0.00 Block Shear_Conc_DF
0.00 Block Initial_DF
0.00 Block Flux_DF
0.00 Block DF_Endpoint
0.00 Block Restore_DF
0.00 End_Block
0.00 Block RECOVERY
(Unused)
Breakpoint
0.00
min
Var...
Instructions
Recirc
Permeate
Alarms
Specials
Watch
Other
Base
Block
Continue
End_Block
End_Method
Evaluate
Flow_Warning
Hold
Loop
Loop_End
Message
New_Chromatogram
Parameters
Var...
Var...
Message
Disable Transfer Feed Pump when loading is comp
Mode
Screen
Sound
No sound
Insert
Change
Replace
Delete
Ready
NUM

6 During a run

Buttons in System Control

Button	Function
Run	The ***Run*** button opens the ***Run*** dialog, which shows all available methods. If a method is loaded, ***Run Setup*** opens.
Hold	The ***Hold*** button suspends execution of the method, while liquid is still pumped at the current flow rate and eluent concentration.
Pause	The function of the ***Pause*** button depends on the strategy. The ***Pause*** button suspends execution of the method and stops all pumps so that the system comes to a standstill.
Continue	The ***Continue*** button resumes the execution of a paused or held method.
End	The ***End*** button terminates the method execution and puts the system into an ***End*** state.

Continuing out of a paused Hold condition

During a run the user can pause an active ***Hold*** condition by clicking the ***Pause*** button. To continue out of the ***Pause*** state and resume the ***Hold*** condition, click the ***Hold*** button.

Note: *Do not click the **Continue** button. Clicking **Continue** will end the **Hold** condition and advance the method.*

Example from UniFlux method templates

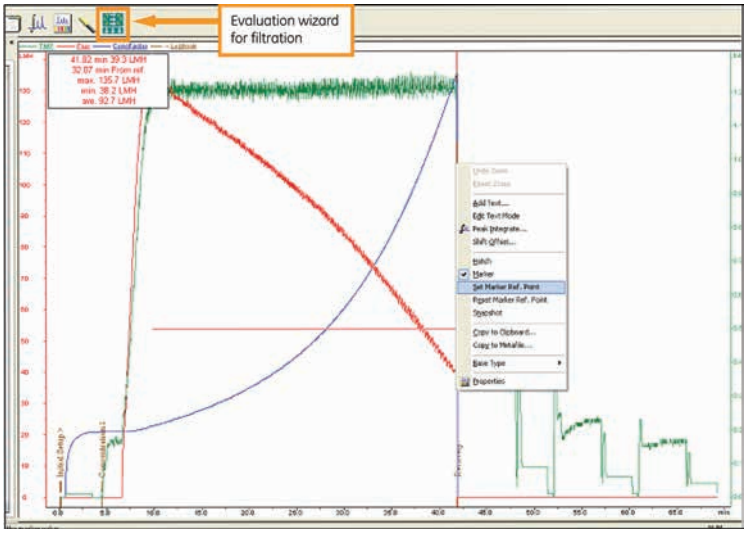
In UniFlux method templates, the ***Hold_Until*** condition is used in the blocks ***Tank_Fill*** and ***FILL_SAMPLE***. The method can enter into a ***Pause*** state if an alarm event occurs or if the user clicks the ***Pause*** button during filling of the tank. To resume the method and continue filling the tank to the desired set point, click the ***Hold*** button.

7 Evaluation

Description

UNICORN ***Evaluation*** module is installed with an evaluation wizard for filtration. Both classic evaluation operations and wizard operations can be used to gain the most information from the result file.

The ***Membrane System Evaluation*** icon in the ***Evaluation*** module shows that the evaluation wizard for filtration is installed.



Included operations

The evaluation wizard for filtration includes the following operations:

- ***Process Optimization***
- ***Normalized Water Flux***
- ***Diafiltration Time Optimization***
- ***Capacity Plots***
- ***Any vs Any***

Curve names

ÄKTAcrossflow system and UniFlux system in some cases use different names for the same curve, see the curve names in bold text in the table below.

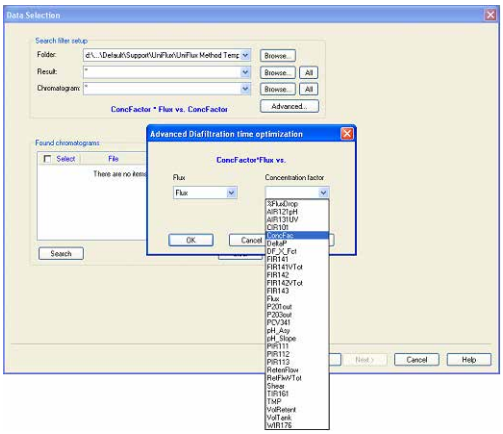
Operation	ÄKTAcrossflow curve names	UniFlux curve names
Process Optimization	Flux, TMP	Flux, TMP
Normalize Water Flux	Flux, TMP, **Temp**	Flux, TMP, **TIR161**
Diafiltration Time Optimization	**ConcFactor**, Flux	**ConcFac**, Flux

When using the evaluation wizard with result files from UniFlux systems, this curve information has to be updated prior to completing the wizard operation, see instruction below.

Update curve information

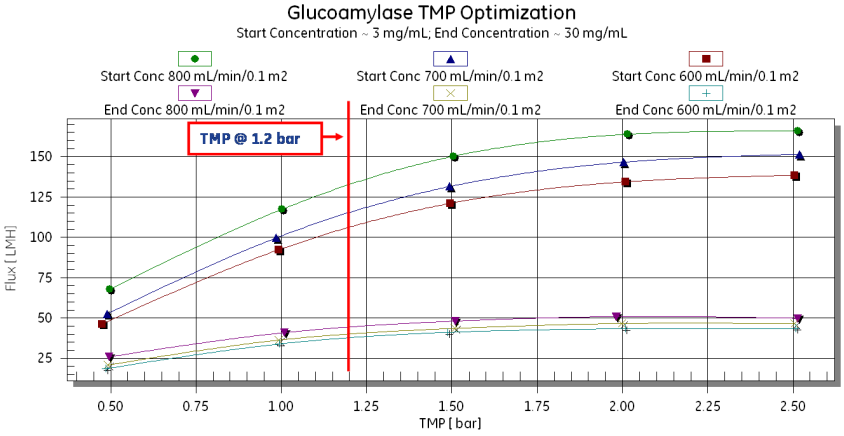
Follow the instruction below to do the necessary updates on curve information.

Step	Action
1	In the ***Evaluation*** module, click the ***Membrane System Evaluation*** icon.  *Result:* The ***Operation*** dialog opens.
2	In the ***Operation*** dialog, select one of the operations. Click ***Next***. *Result:* The ***Data Selection*** dialog opens.
3	In the ***Data Selection*** dialog, use the ***Search filter setup*** section to select the desired result file(s).
4	In the ***Search filter setup*** section, click the ***Advanced*** button. *Result:* The ***Advanced...*** dialog opens.
5	Use the drop-down list to update the curve information according to the table above.


Data Selection
Search filter setup
Folder:
Result:
Chromatogram:
Browse...
All
Advanced...
ConcFactor * Flux vs. ConcFactor
Found chromatograms
Select
File
Search
Advanced Diafiltration time optimization
ConcFactor*Flux vs.
Flux
Concentration factor
OK
Cancel
Next >
Cancel
Help

Example plot - Process Optimization

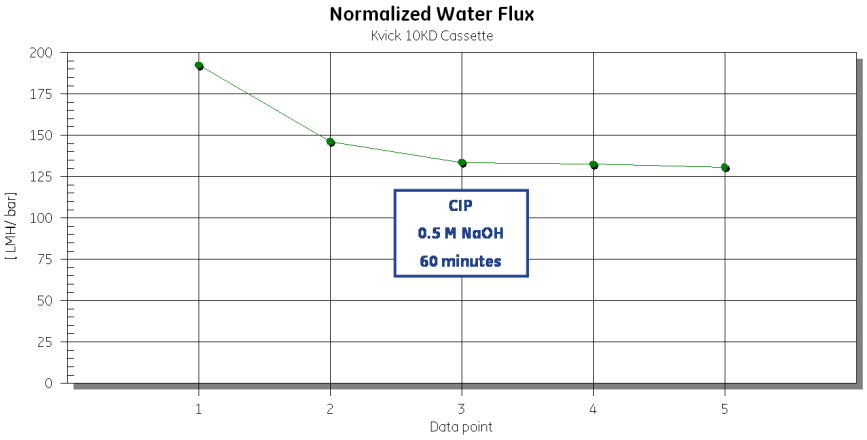
The illustration below shows a ***Process Optimization*** plot for starting and ending concentrations.



Average Flux Across The Process ~ 60 LMH

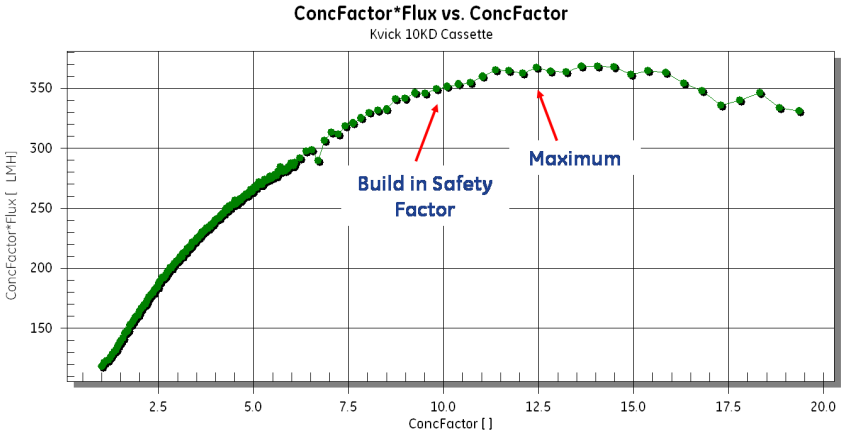
Example plot - Normalized Water Flux

The illustration below shows a ***Normalized Water Flux*** plot for tracking filter performance and cleaning procedures.



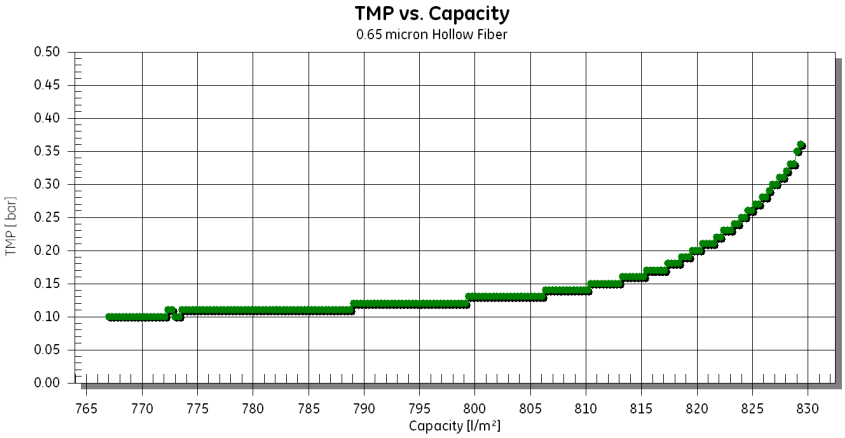
Example plot - Diafiltration Time Optimization

The illustration below shows a ***Diafiltration Time Optimization*** plot for 20× result.



Example plot - Capacity Plots

The illustration below shows a plot of ***TMP*** vs. ***Capacity***. ***Capacity Plots*** allow the user to plot any process parameter versus the accumulating permeate volume normalized to the surface area.



8 Reference information

Literature

For further information on UniFlux systems and UNICORN, refer to the following:

- UniFlux system, Operating Instructions (28961749)
- UNICORN 5.2, User Reference Manual for Bioprocess (28401054)
- UNICORN 5.1, Evaluation for Cross Flow Filtration / User Reference Manual (28401099)
- Operating Handbook - Hollow fiber cartridges for membrane separations (18116530)
- CIP cassette for UniFlux 30, 120, 400 (28964760)

9 Support

On-line support for filter applications

Support for filter applications is available at *cytiva.com*.

On-line support from filtration devices and systems

For support for filtration devices and systems, visit *cytiva.com/service* , and click Bioprocess filtrations.

Remote support

Remote support options are available through Cytiva's Bio InSite application. BioInSite links your instruments with Cytiva experts over a secure digital connection. This provides a possibility to get help when you need it.

For further information contact your Cytiva representative. Additional information can also be found online. Visit *cytiva.com/service* .

Appendix A
Sample handling

Introduction

Normal flow filters (NFF) can be used in conjunction with crossflow applications using UniFlux.

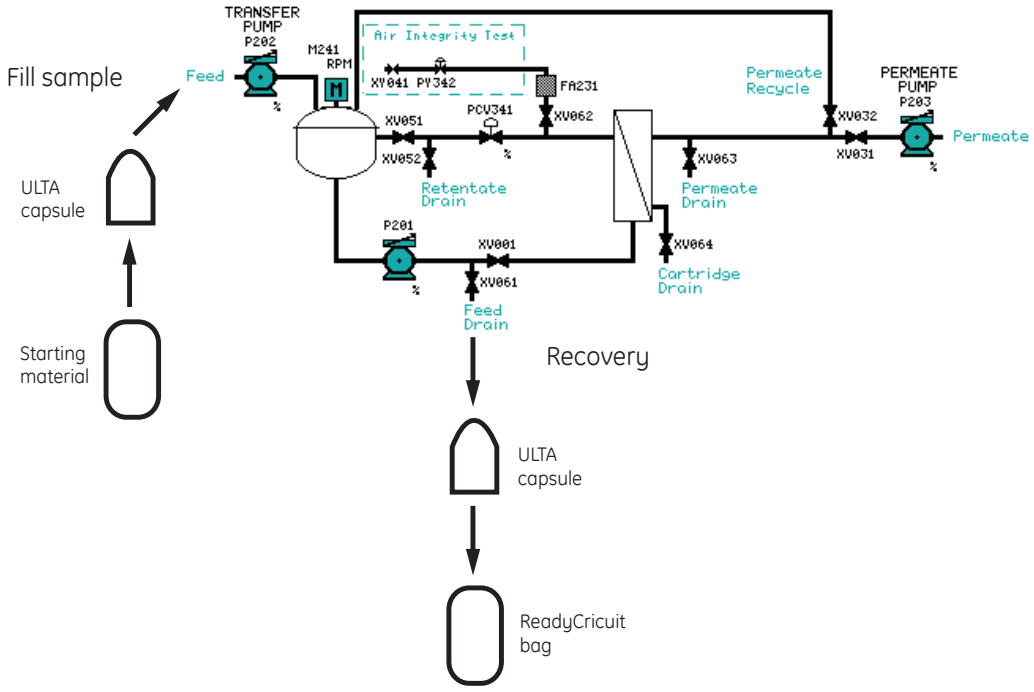
Recommended capsule for sample preparation

Consider and ULTA™ Prime CG capsule from Cytiva for sample preparation prior to and after processing. This 0.6/0.2 µm PES filter can be inserted between the sample tank and the transfer pump, see illustration below. The capsule will filter out any particles that may have settled during sample hold.

Recommended capsules during recovery

Consider an ULTA Prime CG, ULTA Pure HC, or ULTA Pure SG capsule during recovery. ULTA Pure HC and ULTA Pure SG are 0.2 µm sterilizing grade filters. The capsule can be inserted between the feed drain and recovery container or ReadyCircuit bag (part of the ReadyToProcess™ portfolio), see illustration below.

Illustration



Page intentionally left blank

cytiva

cytiva.com/filtration

For local office contact information, visit cytiva.com/contact

28982285 AB V:6 06/2021