

TABLE OF CONTENTS

1	General Information	1-1
1.1	About this Manual	1-1
1.2	Prerequisite Knowledge	1-1
2	Overview	2-1
3	Definitions.....	3-1
3.1	Resource	3-1
3.2	Activity.....	3-1
3.3	Tasks and Methods.....	3-2
3.4	Workflow.....	3-2
3.5	Plan.....	3-2
4	Security Notes	4-1
5	Installation	5-1
6	How to setup Scheduling.....	6-3
6.1	Developing a concept.....	6-3
6.2	Defining the instrument layout and the resources	6-3
6.2.1	Defining Resources	6-4
6.2.2	Add Resource Dialog Box.....	6-5
6.2.3	Mapping Resources.....	6-6
6.3	Defining Activities within the Step Editor	6-8
6.3.1	Creating a new method.....	6-8
6.3.2	Defining Activities	6-8
6.3.3	Activity Cancel Block	6-11
6.3.4	Activity Dialog Box	6-12
6.4	General rules	6-1
6.5	Activity Editor.....	6-2
6.5.1	Adding an activity.....	6-2
6.5.2	Editing an Activity	6-4
6.5.3	Deleting an Activity	6-4
6.5.4	Code outside of Activities	6-5
6.5.5	Changing image icon	6-5
6.5.6	Adding / Editing / Deleting Resources	6-5
6.5.7	Context menu's in activity flowchart.....	6-6
6.5.8	Running the method	6-6
6.6	Defining a Workflow using the Graphical Workflow Section	6-8
6.6.1	Creating a new Workflow.....	6-8

6.6.2	Selecting Methods for the Workflow	6-8
	Adding the selected methods to the workflow	6-9
6.6.3	Defining Task Dependencies.....	6-9
6.6.4	Task Activation Dialog Box	6-10
6.6.5	Activating multiple Tasks	6-11
6.7	Defining A Workflow Step by Step.....	6-12
6.7.1	Creating A New Workflow	6-12
6.7.2	Defining a Workflow	6-13
6.7.3	Variable Controlled Scheduling (Example).....	6-13
6.8	Run Control	6-14
6.8.1	Starting the Scheduler from Method Editor	6-14
6.8.2	Starting the Scheduler from Run Control.....	6-14
6.8.3	Task View	6-15
6.8.4	Resource View.....	6-16
6.8.5	Task List.....	6-16
6.8.6	Unattached Tasks.....	6-17
6.8.7	Optimizing a Run	6-18
6.8.8	Real Time Dynamic Adaptions (Example).....	6-19
7	Test Run.....	7-1
8	Production Runs	8-1
8.1	Starting and Running.....	8-1
8.2	Interventions at Runtime	8-2
8.2.1	Removing or Cancelling a Task.....	8-2
8.2.2	Rescheduling	8-2
8.2.3	Adding, Editing and Deleting Unattached Tasks	8-3
8.2.4	Edit/New Unattached Task Properties Dialog Box	8-4
8.2.5	Grouping and ordering of unattached tasks	8-5
8.2.6	Schedule unattached tasks.....	8-6
9	Rules for Programming.....	9-1
9.1	General rules	9-1
9.2	Rules for Global and Local Variables	9-2
9.2.1	Default Definitions of Variables.....	9-2
9.2.2	Specification of Global Variables	9-2
10	Export and Import of Workflows.....	10-1
11	Additional Functions for the scheduler.....	11-1
11.1	System Configuration.....	11-1
11.1.1	Scheduler Settings	11-1
11.1.2	Schedule view options.....	11-2
11.2	Scheduler Steps.....	11-1

11.2.1	Select Method Step	11-1
11.2.2	Activate Task Step.....	11-3
11.2.3	Cancel Task Step	11-7
11.2.4	Schedule Block Step	11-8
11.2.5	Reschedule Step	11-8
11.3	Resource distribution	11-9
11.4	Programming additional tasks at runtime (example)	11-10
11.5	Scheduler Step Modes.....	11-12
11.6	Additional Tools for the <i>VENUS Dynamic Scheduler</i>	11-12
11.6.1	Workflow Toolbar.....	11-12
11.6.2	Scheduler Toolbar	11-12
11.7	Additional Menu Functions for the <i>VENUS Dynamic Scheduler</i>	11-13
11.7.1	File Menu.....	11-13
11.7.2	Edit Menu	11-13
11.7.3	View Menu.....	11-13
11.7.4	Workflow Menu.....	11-13
12	Glossary	12-1



1 General Information

The VENUS Dynamic Scheduler is a supplemental software package for the VENUS base software. After the installation, scheduler depending functions and editors are added to the base package of VENUS. The VENUS Scheduler is a dynamic scheduler, that means that a plan is recalculated constantly. Having changes in the execution of a plan (e.g. a task that uses more time than scheduled), the scheduler will recalculate the plan and find the optimal way to process the tasks.

This chapter should be read thoroughly before programming and using the *VENUS Dynamic Scheduler*. It contains important information about the use of the *VENUS Dynamic Scheduler* and this manual.

All features described in this manual are valid for the *VENUS Dynamic Scheduler*. Refer to the corresponding manuals for programming and operating the instruments also when using the *VENUS Dynamic Scheduler*.

After working through this manual, you will be quite comfortable with programming the VENUS Dynamic Scheduler.

Warnings and *Notes* are included in this manual to emphasize important and critical instructions. They are printed in italics and begin with the word 'Attention' accompanied by the  symbol, or the word 'Note' accompanied by the  symbol, as appropriate.

This manual refers to VENUS Dynamic Scheduler release 5.1 and MICROLAB[®] STAR VENUS Software base package 4.2.

1.1 About this Manual

The *VENUS Dynamic Scheduler User Manual* is the basic manual to help programmers to program and run the *VENUS Dynamic Scheduler* correctly and safely. The corresponding manuals for programming and operating the instruments controlled by the *VENUS Dynamic Scheduler* contain all relevant basic information.

The *VENUS Dynamic Scheduler User Manual* shows you step by step how to perform typical operations using the *VENUS Dynamic Scheduler*. Examples of typical applications guide you through programming and running the schedules. When you have worked through this manual, you should be able to write and run your own workflows.

1.2 Prerequisite Knowledge

In order to be able to program the *VENUS Dynamic Scheduler* successfully and safely, the programmer must be thoroughly familiar with the Windows environment, must understand the laboratory processes that is automated, and must have taken appropriate Hamilton-authorized training on Hamilton hardware and software.

If hardware and software of other manufacturers is controlled by the *VENUS Dynamic Scheduler*, equivalent education and knowledge is required.

2 Overview

A scheduler is a tool to manage parallel processes with respect to available resources. Accordingly, the *VENUS Dynamic Scheduler* helps to organize and control the workflows (assays, plate runs etc) or methods of a laboratory equipped with Hamilton and other instruments.

The *VENUS Dynamic Scheduler* optimizes the overall workflow/method by searching for a near-optimal use of the resources and preventing multiple use of the same resource. This depends on the available resources and their capacity, the used time for the execution of an activity and the relation between the tasks to minimize execution time. The tasks and their activation criteria are defined in a workflow or in an activity editor method.

Example:

The picture below shows a plan to run multiple tasks.

The first plate starts with transport/pipette/transport **1**. After that, the shaking action **2** takes place. Finally, the instrument performs another transport step **3**.

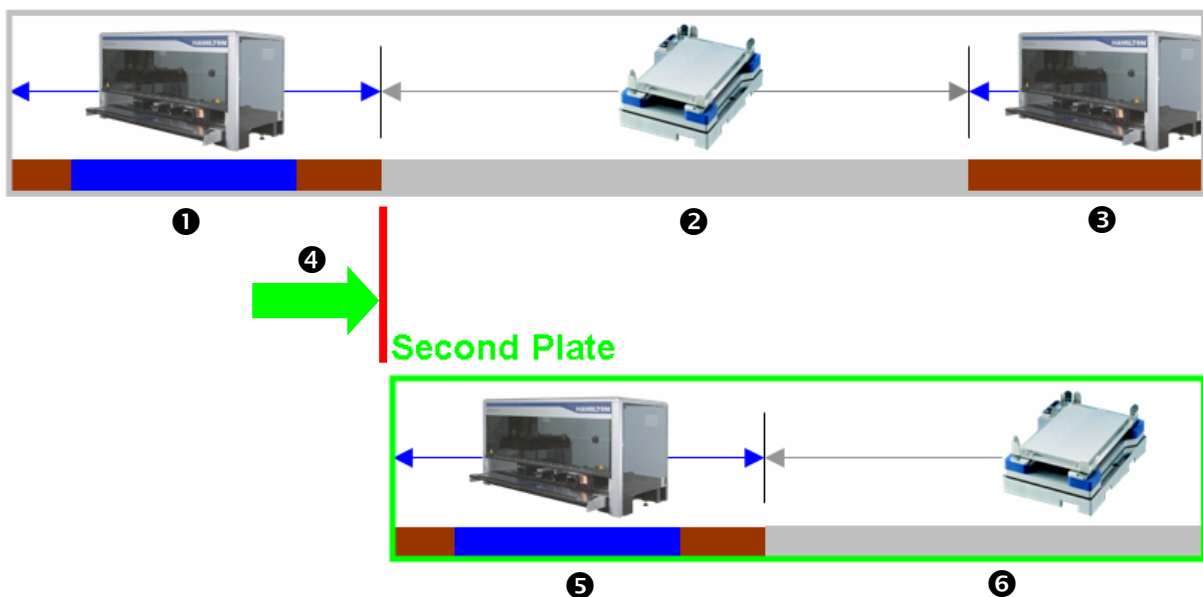
The start point for the second plate **4** is now calculated to be while the first plate is shaken. Because the 2nd plate's transport/pipette/transport task **5** is shorter than the shaking action of the first plate **2**, it will be scheduled in that gap. If there is more than one shaker available, the second plate's shaking **6** will start immediately after the second plate's transport/pipette/transport step **5**.



NOTE

As you can see, the plan is calculated by the use of resources. The Scheduler does arrange these Method blocks in a way to get the shortest possible run time for all activities.

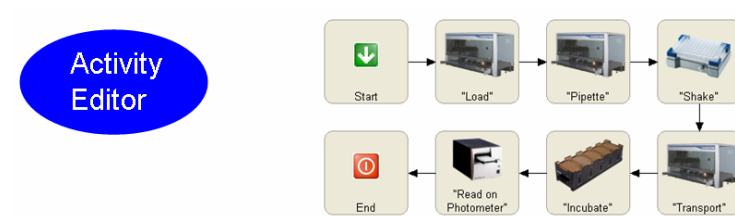
First Plate



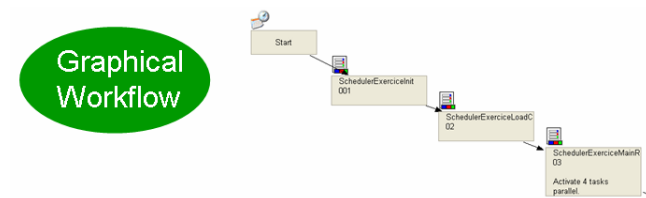
VENUS Dynamic Scheduler Manual

The scheduling is based on tasks (methods) with programmed or predefined activities. Resources (e.g., instruments) are defined in the system deck layout and are linked to activities within the methods.

Three levels of programming are available:



A simple, intuitive approach to the scheduling environment. Ideal for throughput calculation and system design.



Graphically-oriented programming with intermediate flexibility. Allows easy integration from the non-scheduling environment into the powerful scheduler world.



Step-wise programming for full flexibility. Offers the full span of VENUS options

Instruments of other manufacturers may be integrated if they are compatible with the Hamilton system. Please ask your Hamilton distributor for a list of available drivers for instruments of other manufacturers called '3rd party drivers'.

3 Definitions

3.1 Resource

A **resource** is an instrument, a part of an instrument, e.g. a pipetting arm, a transfer position, a device like a reader, or a piece of labware. Resources perform activities. Each resource has a capacity. Here's a list of examples:

resource	capacity	of
Pipetting arm	1	all pipetting channels
Carrier	5	MTP positions
Incubator	20	e.g., MTP positions
Incubator Transfer Pos.	1	MTP positions

3.2 Activity

An **activity** is the smallest planning unit for the scheduler. An activity may include a single step, a sequence of steps or a sub method call. An activity uses the same set of resources for a certain time. Activities are defined – or groups of steps are assigned to an activity – within the activity editor or the step editor.



NOTE

For throughput calculation or resource planning, an activity may contain only a comment. Within the activity editor, it is also possible to create empty activities.



Within the activity definition, all resources used in this activity must be defined.

3.3 Tasks and Methods

A **method** is a sequence of activities. Once a method is scheduled or running, it is called a **task**. If you have a method to process one plate, and you schedule five methods to process five plates, the plan will show five tasks.

Task

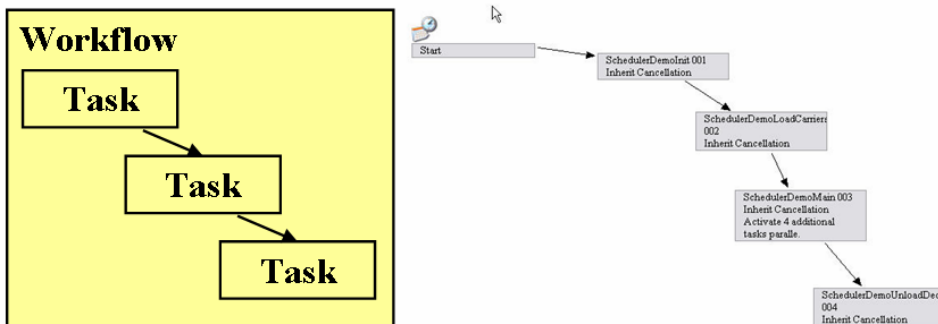
Activity

Activity

Activity

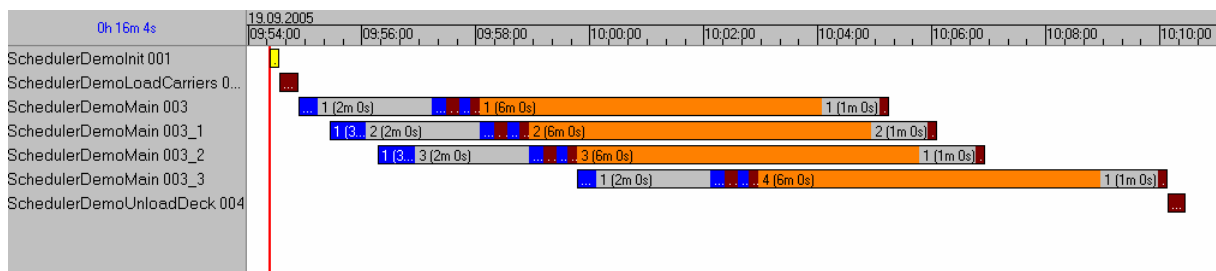
3.4 Workflow

A **workflow** is a number of chained sequences tasks (methods) and their dependencies. It defines the order of the tasks on runtime. It can be defined using the graphical workflow section and/or using the step by step editor (the advanced window) for individualized workflows.



3.5 Plan

A **plan** is a scheduled workflow. After scheduling, the plan shows the start and end times of each activity due to the availability of the resources and on the dependencies that you set.



4 Security Notes



ATTENTION

To perform a scheduling process, ALL OF THE METHOD CODE outside of activities IS EXECUTED.

This is true both when scheduling in the Method Editor and when loading a workflow into the Run Control program.

This is also true when executing a reschedule on a method that hasn't previously (during the current Run Control execution) been scheduled.

The result of the scheduling process running the code is that any INSTRUMENT COMMANDS that are not inside of an Activity block WILL BE EXECUTED. To prevent the instruments from moving in an unpredictable way during this reschedule the user should:

- *Ensure that all instruments referenced (in any way) by the method are running in simulation mode, or*
- *switch off the instrument, or disconnected it from the PC, if no simulation mode is available.*



ATTENTION

Please make sure that all Init commands for HAMILTON and third party instruments are also part of an activity. Otherwise, these init commands are executed during the calculation run → the instruments moves in an unpredictable way.



ATTENTION

Avoid name conflicts between global, task local, and local variables!

Refer to chapter Rules for global and local variables

All security and safety advice for using the involved instruments and liquids must be followed strictly.

5 Installation

Insert the installation disk in the CD or DVD drive. The Installation is started automatically. Follow the advice of the installation program.



NOTE

If the installation does not start automatically, start it using the Windows program execution function (“Start button” – “Run” and enter [CDROM]:\SchedulerInstaller.exe).



ATTENTION

- *The installation of the VENUS Dynamic Scheduler software has to be done **after** the installation of the VENUS software and the installation of all instruments.
During installation of an additional HAMILTON instrument, the VENUS Dynamic Scheduler software may be removed and must be reinstalled.
*This is also true if a repair installation is executed.**
- *De-activate anti-virus programs during installation as they may interfere with the installation program. The VENUS Dynamic Scheduler software has been checked and is free of the momentary known viruses.*



NOTE

The installation of the scheduling software does not create an own icon on the desktop. All additional features and editors are included to the base package of the VENUS software.

6 How to setup Scheduling

The way to set-up a system is described in the following steps:

- Develop a concept (see 6.1).
- Define the instrument layout(s) and the resources (see 6.2).
- Define activities within method(s) and/or the Activity Editor.
- Define the workflow graphically **or** define the workflow step by step in the Advanced window.
- Create the schedule.
- Test (and improve) the schedule.

6.1 Developing a concept

Using the user's knowledge about the final product, the following has to be planned:

1. Which processes can run in parallel? → Splitting into tasks (methods).
2. Which resources will be used? → Incubator, Carrier, etc.
3. What is the optimal arrangement of the resources? → Layout planning.
4. What is the order of steps for each task? → Method
5. What resources are used for each method step? → Assignment of resources
6. Which order of steps can be combined in one activity? → Activities
7. Are there time sensitive steps that must be processed uninterrupted? → Rules

Example:

An Instrument (resource) pipettes (activity) MTPs which then will be incubated (activity) in an incubator with 10 sites (resource), measured (activity) in a photometer (resource) and placed (activity) finally on an exit stack (resource). There is one transfer station (resource) in front of the incubator which can hold one plate to move it inside the incubator to a free site or move it from there to the outside. A MICROLAB[®]-SWAP (resource) is used to move (activity) the plates between the different instruments (resources).

6.2 Defining the instrument layout and the resources

Define the deck layout as described in the “Microlab STAR Line Programmer’s Manual”.

If multiple instruments are used (e.g. ML STAR + ML SWAP + ...), a system deck needs to be created to combine multiple deck layouts in one system.

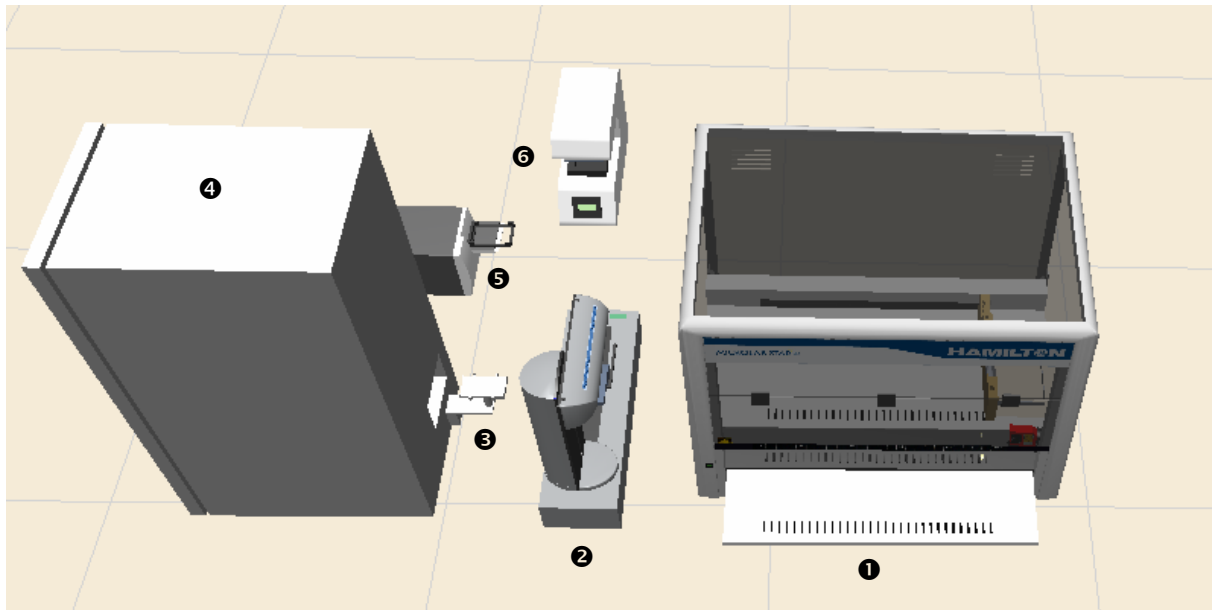
A resource variable is generated for each instrument (e.g. ML_STAR by default). It may be necessary to specify additional resources for instrument parts, e.g. a plate carrier can be used as a room temperature incubator with capacity 5.

Examples:

A wash station must be a resource of its own. Otherwise, it blocks the complete instrument during washing time.

For movable resources, all areas that they access (transfer positions) or cross must be specified as resources. **Otherwise, the instrument may crash.**

The system deck layout for the first given example might look as follows:



- | | |
|--------------------|-----------------|
| ① ML-STAR | ④ Incubator |
| ② ML-SWAP | ⑤ Photometer |
| ③ Transfer station | ⑥ Vapor station |

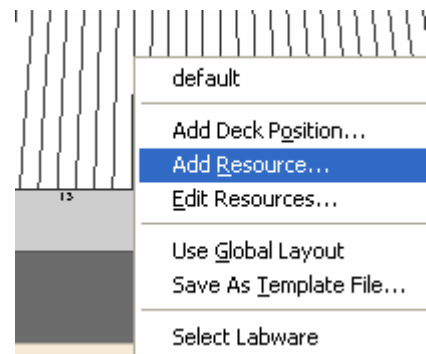
6.2.1 Defining Resources

1. Via the Method editor, open the system deck layout.
2. Right click on the deck in either the instrument view (or the system view if you have several instruments) and select “Add Resource” to define a resource

or

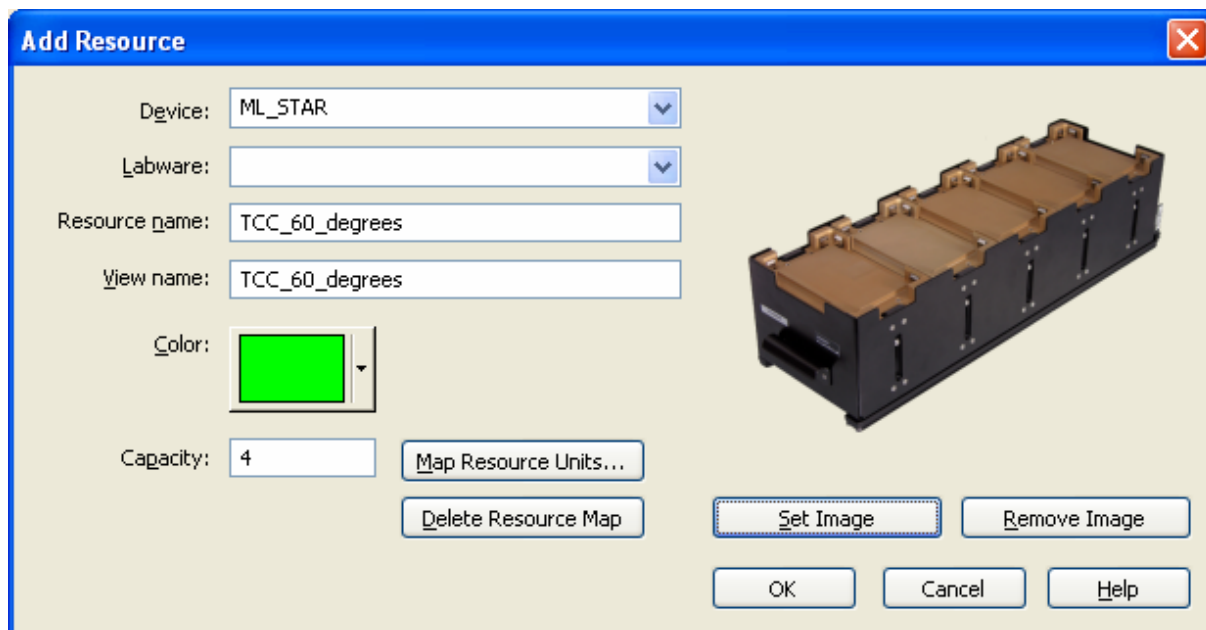
right click the labware and select “Add Resource”, to define a resource for a labware. The “Add Resource” dialog box appears.

3. Type in the parameter values, select the color and click “OK”.



Refer to sections 6.2.2 *Add resource dialog box* and 6.2.3 *Mapping Resources*.

6.2.2 Add Resource Dialog Box

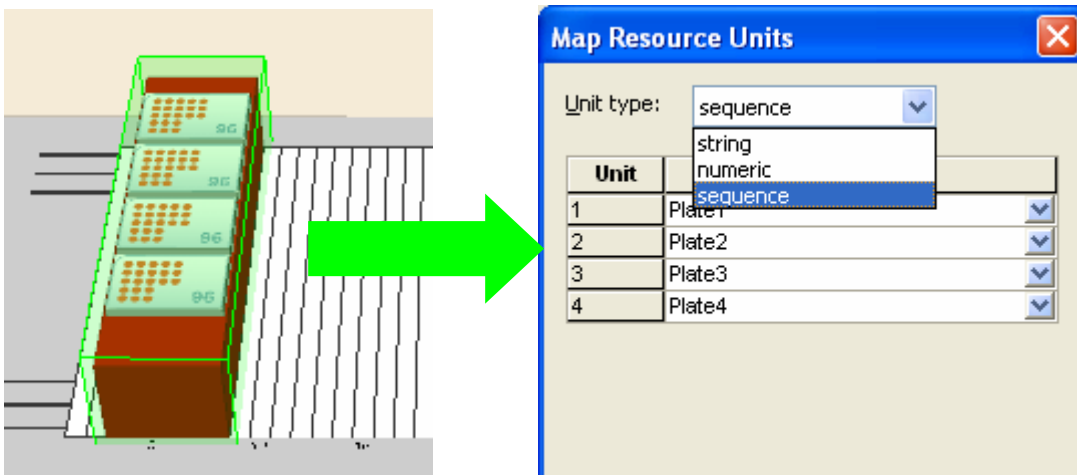


- **Device**
If a resource should be linked to a device, select it here. Used to map resource units to sequences.
- **Labware**
If a resource should be linked to a labware, select it here.
- **Resource name**
The resource name is used to reference the resource when adding activities to your method. The activity block will obtain the resource during method execution (when free) and lock the resource for use.
- **View name**
The display name is displayed in the scheduler view along with the color. Checking the correct resource assignment in the scheduler view is much easier this way.
- **Color**
The selected color will be displayed in the schedule run view during runtime. It is useful to define a different color for each resource.
- **Capacity**
The capacity specifies the total capacity of the resource (must be greater than 0). The default capacity is 1.
- **Map Resource Units...**
Refer to section 6.2.3 *Mapping Resources*
- **Delete Resource Map**
Delete resource mapping.
- **Set Image / Remove Image**
Select an image (bmp, jpeg, png etc.) for the resource to be shown in the Activity Editor or remove the image to add a different one or leave the resource without image.

6.2.3 Mapping Resources

Resource mapping is used to refer to a resource location/position by something other than the default numeric mapping of 1 to capacity.

1. In the “Add Resource” dialog box, click “Map Resource Units...”.
The “Map Resource Units” dialog box appears:



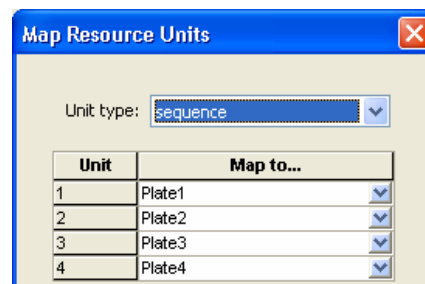
2. Select the Unit type of a string, a numeric, or sequence. The option sequence is only available if the Resource is ‘linked’ to a device.
3. Type in the mapping or select from the sequence list, and click “OK”.

This allows you to use the mapped sequence directly in the method step. This is especially helpful using transport or pipetting steps, where the resource unit variable can be inserted directly in the transportation or pipetting step.

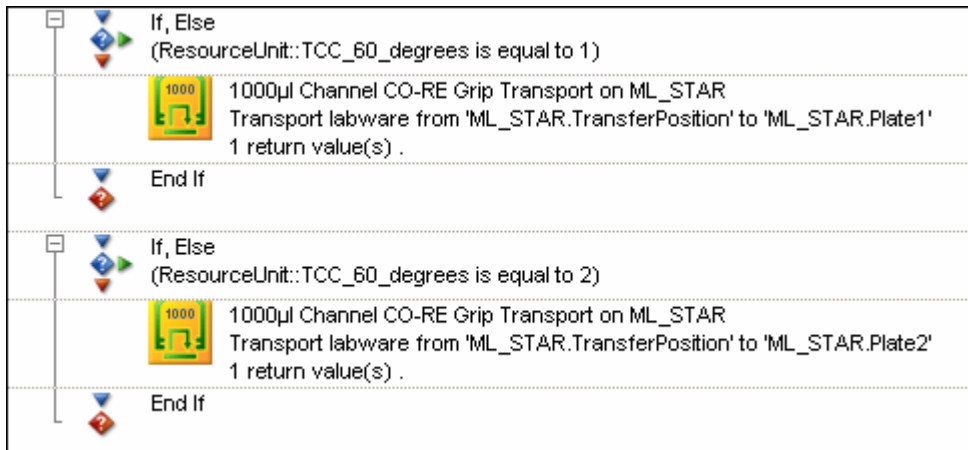
Example

If the default mapping of a resource has a defined capacity of 4, then the locations are referred to as ‘Plate1’ through ‘Plate4’. Mapping can be used to change the definition for example as “Plate1” through “Plate” sequences. Then the sequences can be accessed directly by the resource unit variable.

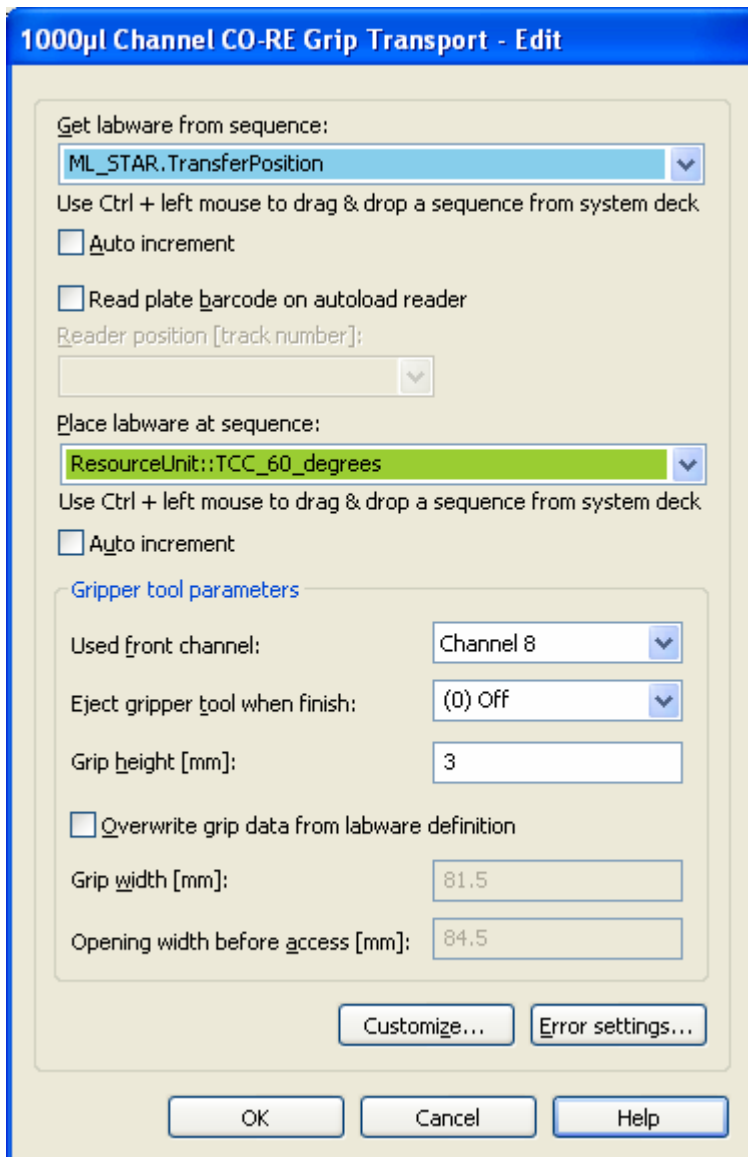
TCC position 1 = sequence ‘Plate1’
 TCC position 2 = sequence ‘Plate2’
 TCC position 3 = sequence ‘Plate3’
 TCC position 4 = sequence ‘Plate4’



Instead of using an if/else query (like “if ResourceUnit::TCC = 1 then get plate from sequence Plate1”) in the method,



you can directly use the mapped sequence in the step. To do so, just enter the resource's unit variable in the step:



6.3 Defining Activities within the Step Editor

6.3.1 Creating a new method

According to the concept, a method is necessary for each task. You can use existing methods or create new ones.

Define new methods as described in the “[Microlab STAR Line Programmer’s Manual](#)”.

Write methods that are describing the way of a single entity (like one sample, one plate, one batch etc.) through the system.

6.3.2 Defining Activities



NOTE

All methods used for scheduling have to be split into activities.

The activities contain all necessary information for the scheduler to create a plan:

- *Activity Name*
- *Activity Duration and*
- *Used Resources.*

These three values are the minimal requirements to create a plan, although you have the option to set pre- and post-activities, minimum and maximum check of the activity duration, maximum planning time etc.

The scheduler does not care about the steps inside of an activity to create a plan: The scheduler is just using the values in the activity ‘header’. Therefore, you can have empty activities.

1. In the Method editor, open the method to be scheduled.
2. Select the first set of steps that can be assigned to one activity.
3. Right-click to activate the context - menu and select “Activity...”. The “Activity” dialog box appears.
4. Specify the parameter values. Refer to section 6.3.4 *Activity Dialog Box*.
5. Add the resources needed within this activity.



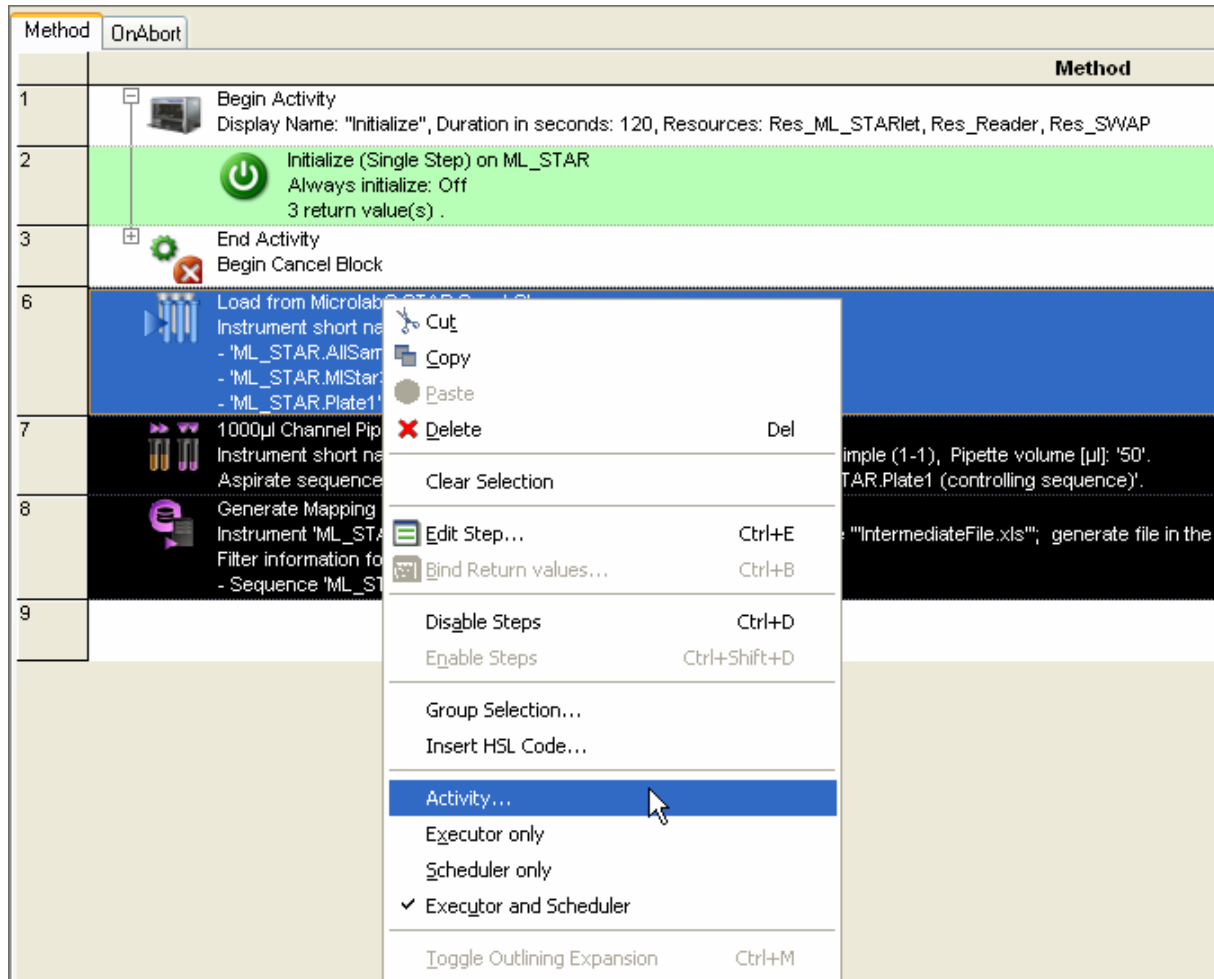
NOTE

- *Even though the scheduler is able to handle durations of 0 seconds, it is recommended to insert durations > 0. Tasks with a duration of 0 seconds are hard to see in the scheduled plan, this can be confusing to the user.*
- *Make sure that all involved resources are added. Especially transport steps need several resources: get position / place position / transporter / transfer position....*

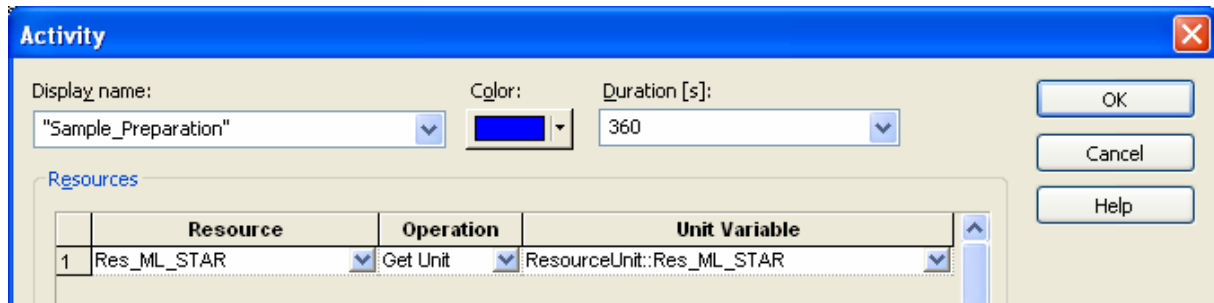
6. Click “OK”.
7. Continue with the next set of steps in the method until the method is ready for scheduling.

Example

1. In the Method editor, select one or more steps that you want to group in one activity. Right click to invoke the context menu. Select the function “Activity...” in the context menu.
















2. Type in the display name and select color and duration. Add all necessary resources used in this activity by pressing the “Add” button. Make sure that no resource is forgotten.



3. Click “OK”.

VENUS Dynamic Scheduler Manual

This is the result after click the "Plus" button   **Begin Activity** on the just created "Begin Activity" and "End Activity/Begin Cancel Block" statements.

Method	OnAbort	Method
1		Begin Activity Display Name: "Initialize", Duration in seconds: 120, Resources: Res_ML_STARlet, Res_Reader, Res_SWAP
2	1 	Initialize (Single Step) on ML_STAR Always initialize: Off 3 return value(s) .
3	 	End Activity Begin Cancel Block
6	2 	User Input Dialog Title: "Input: Number of Samples", Return Value: ", Buttons: 'Only 'OK' button', Default: 'OK', Sound: ", Timeout: 'in Input: NumOfSamples ("What is the number of samples?", Integer, 96, 24, 384)
7		Begin Activity Display Name: "Activity 1", Duration in seconds: 460, Resources: Res_ML_STARlet
8		Load from Microlab® STAR Smart Steps Instrument short name 'ML_STAR', load '3' sequence(s): - 'ML_STAR.AllSamples' - 'ML_STAR.MiStar300ulStandardVolumeTip' - 'ML_STAR.Plate1'
9		1000ul Channel Pipette - Simple (1-1) from Microlab® STAR Smart Steps Instrument short name 'ML_STAR', Standard pipette procedure: Mode: Simple (1-1), Pipette volume [µl]: '50'. Aspirate sequence: 'ML_STAR.AllSamples', Dispense sequence: 'ML_STAR.Plate1 (controlling sequence)'.
10	3 	Generate Mapping File from Data Handling Steps Instrument 'ML_STAR'; Target sequence 'ML_STAR.Plate1'; Mapping file "'IntermediateFile.xls"; generate file in the Filter information for the following 1 sequences: - Sequence 'ML_STAR.AllSamples' on instrument 'ML_STAR'
11	 	End Activity Begin Cancel Block
12		Abort
13		End Cancel Block

- 1** Specified activity
- 2** Assigned method steps
- 3** Automatically inserted "Cancel Block"



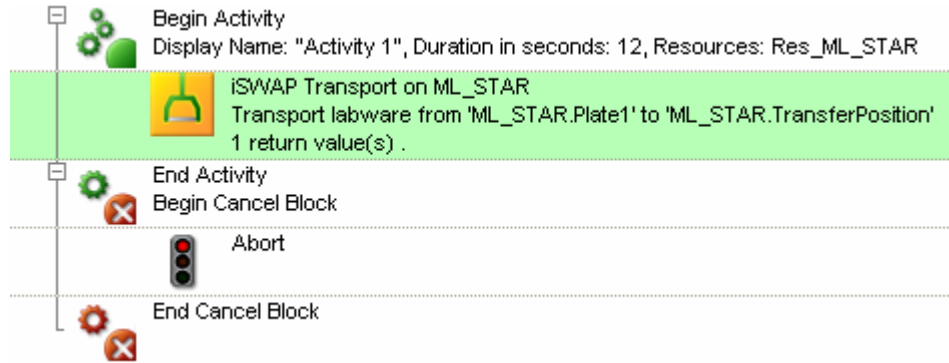
As you can see, the activity's steps are 'framed' by two 'activity step' icons: "Begin Activity" and "End Activity". This is a number of method steps which use the same set of resources. It can be displayed with or without contents.



The "End Activity" step icon is in parallel the beginning of an automatically inserted "Cancel Block" which is framed until the step icon "End Cancel Block". An Abort step is inserted by default. Refer to Section Activity Cancel Block for more details

6.3.3 Activity Cancel Block

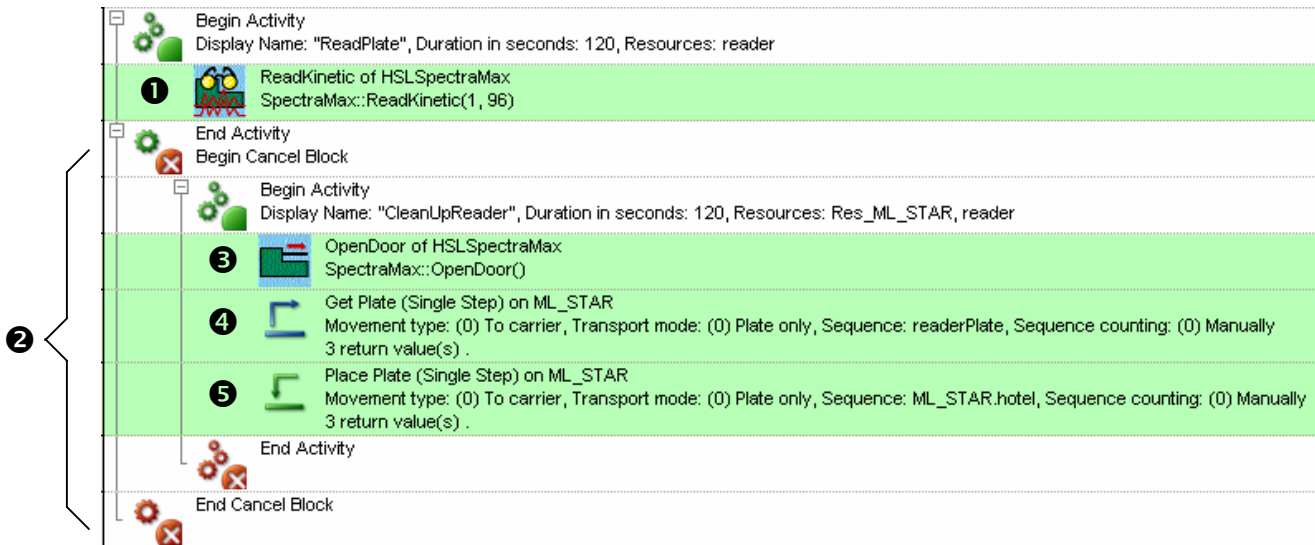
The End Activity step icon is in parallel the beginning of an automatically inserted Cancel Block which is framed until the step icon End Cancel Block.



In case of an abort of this activity, the code between Begin Cancel Block and End Cancel Block is executed. Per default, an abort step is programmed. This may not make sense if you want to have a walk away system. In this case, the Cancel Block should be programmed according to your needs.

See picture below for an example of Cancel Block programming.

The activity “Read plate” consists of a reading step for the SpectraMax photometer.



- **Case A**

Everything works fine, the Cancel Block ② is not executed, the method continues.

- **Case B**

The “ReadKinetic” step ① does not work fine and the activity is cancelled. In this case, the steps in the Cancel Block ② are executed:

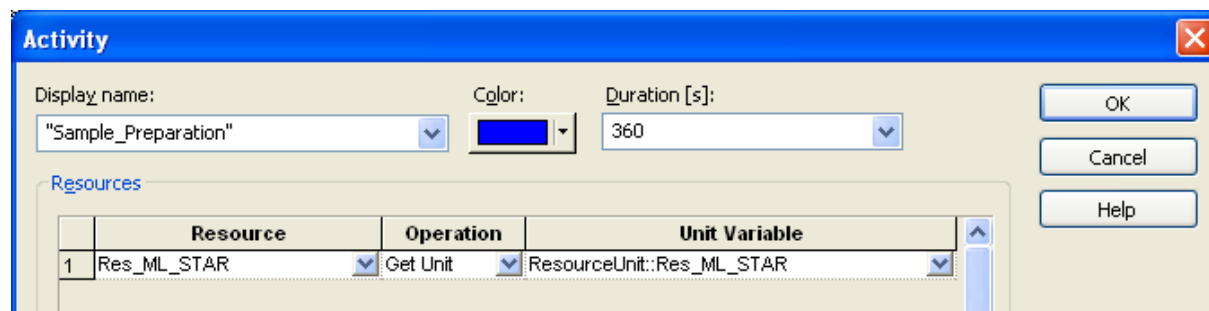
- The door of the photometer is opened ③
- The iSWAP get’s the plate from the photometer’s drawer ④ and brings it to a hotel ⑤.

This allows the method/workflow to continue, because the resource Photometer is available for further tasks. If this is not done, the resource Photometer will never be released and the method hangs.

Please note that the steps in the Cancel Block must be specified as activity too, so the used resources can be linked. As you can see, the Cancel Block can contain more resources (e.g. the iSWAP of ML_STAR) than the original activity.

The default step in the Cancel Block is an Abort.

6.3.4 Activity Dialog Box



- **Display name**

Name of the activity. The name is displayed in the scheduler view and written to the trace file for traces related to this activity. The “Display name” combo box accepts a string, which must be enclosed in quotation marks or a variable (that contains the display name in string format).

- **Color**

Color associated to the activity. The scheduler view displays the activity with the specified color. Select a color from the list box. Select same colors for same type of activities, e.g. pipettings in blue, incubations in red, transports in yellow and so on. This makes a plan more understandable.



NOTE

The colors are displayed in the scheduler view. If they are used systematically, they may serve as a visual check of the correct assignment of resources. Here is the default assignment of colors and actions

Pipette	→	blue	/	Transport	→	grey
Load	→	pink	/	Incubate	→	green
Read	→	yellow	/	Wash	→	cyan

- **Duration[s]**

Estimated time needed for executing in seconds. The “duration” combo box accepts an integer value or a variable name (that contains an integer value).

- **Resources**

Resources are the items that are needed to perform the activity. To add a resource, click the **Add** button, then select a resource from the “Resource” combo box. Activities can contain multiple resources (e.g. plate transport from transfer to shaker needs the resources ML_STAR, Transfer_position and shaker)



NOTE: HOW TO KNOW WHICH UNIT OF A RESOURCE IS USED BY AN ACTIVITY AT RUNTIME?

The VENUS Dynamic Scheduler returns the dedicated unit that is planned for this activity on runtime. The dedicated unit is assigned to the variable "ResourceUnit:<ResourceName>" and is available to the steps inside the activity. This variable can be used for programming (if-else decisions, loop variables etc) E.g.: The unit for the resource "Resource_TCC" is assigned to the variable "ResourceUnit::TCC".
A more elegant way to solve this issue is to map sequences to a resource. Refer to the section Mapping Resources for more information

- **Button "Advanced"**

Displays additional parameters at the bottom of the dialog box:

Activity

Display name: "Sample_Preparation" Color: [Blue] Duration [s]: 360

Resources

	Resource	Operation	Unit Variable
1	Res_ML_STAR	Get Unit	ResourceUnit:Res_ML_STAR

Buttons: Add, Remove, Advanced <<

Pre-Activity: [] Change...

Post-Activity: [] Change...

Monitoring

Minimum executed duration [s]: [] Unmonitored

Maximum executed duration [s]: [] Unmonitored

Maximum planned duration [s]: [] Use default

Activity ID: [{"AE294FB8-64EA-487A-A784-4655C3AD2EBF}"] Create ID

Duration

Simulate actual time Automatic adjustment

Algorithm: Arithmetic Mean

Buttons: Set Estimated Duration..., Reset Activity Duration

Buttons: OK, Cancel, Help

- **Operation: Get unit**

The activity uses one unit of the specified resource. The distribution of the units is done by the system automatically, that means the user does not care about the selection of the units.

- **Operation: Set unit**

In this case, the user must specify the unit of the resource that shall be used by this activity. This can be done with a simple assignment from the General Steps:

```
Assignment  
X=0 'ResourceUnit::Incubator' = '2'
```

The activity will now use unit 2 of this resource.

- **Unit Variable**

Having the option *Get unit* selected, the system will write the used unit into this variable. This makes it possible to make decisions depending on the value of this resource unit variable. E.g. if a plate has been transported to a shaker carrier, you can use the resource unit variable to find out where you have to pipette.

Another way to handle Resource unit variables is described in the section *6.2.3 Mapping Resources*

- **Minimum executed duration [s]:**

This field can be used to set a minimal time limit on an activity. If the activity is ended before the limit (the minimum executed duration) has exceeded, a time constraint error will be invoked.

Value: The field accepts integer values or variables of type integer. The value must be greater than or equal to 0 and less than or equal to the activity's duration.

Monitoring: If the "Unmonitored" check box is not selected, the *VENUS Dynamic Scheduler* monitors this constraint at runtime. Any violation of the constraint cancels the task. If a Cancel Block is programmed, the activities in the Cancel Block will be executed. The default value is "Unmonitored".

- **Maximum execution duration [s]**

This field can be used to set a maximal time limit on an activity. If the activity is not ended after the limit (the maximum executed duration) has ended, a time constraint error will be invoked

Value:

Integers and variable names are accepted. The value must be greater than or equal to the specified duration.

Monitoring:

If the "Unmonitored" check box is not selected, the *VENUS Dynamic Scheduler* monitors this constraint at runtime. Any violation of the constraint cancels the task and executes the Cancel Block activities (if programmed). The default value is "Unmonitored". Specifying a maximum duration allows the programmer to fully control any critical timing inside a method and ensures repeatability for every task. For example, if you wanted to add a reagent, incubate and then wash a plate, you can specify the timing for these consecutive activities to how long you wanted the reagent to be in contact with the samples in the plate.

- **Maximum planned duration [s]**

Per default, the scheduler is allowed to stretch an activity up to it's double time to improve the final plan. The Maximum planned duration allows you to limit this time for a single activity without changing the factor for all activities. Please note that this setting is only controlled at planning time.

Value:

Integers and variable names are accepted. The value must be greater than or equal to the specified duration. Make sure that you do not create senseless combinations with Minimum/Maximum executed durations.

Use default:

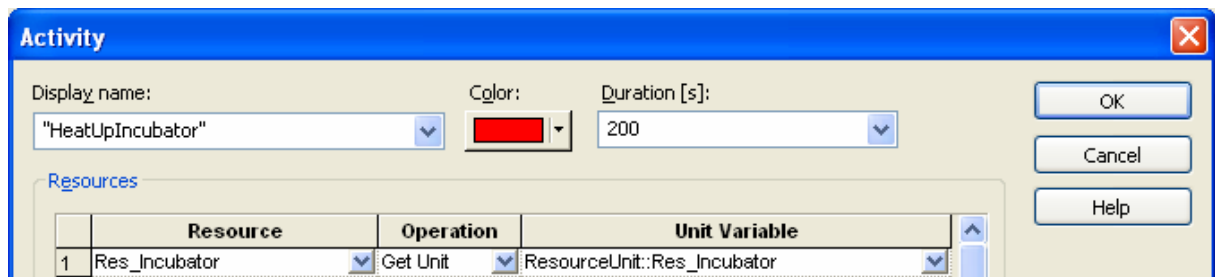
If checkbox "Use default" is not marked, the *VENUS Dynamic Scheduler* monitors this constraint at planning time. The default value is twice value of "Duration".

- **Pre-Activity / Post-Activity**

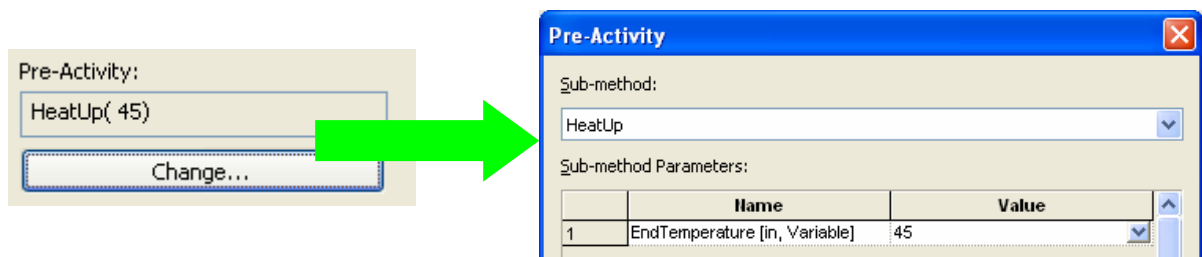
Pre- and post activities can be used to perform an action in parallel in the same task. Of course, the used resources must be free to execute both actions in parallel. An example of this function could be the heating up of an incubator while pipetting:



The pre- or post activity has to be defined as submethod. In this submethod, you have to create an activity containing resource(s) and timing(s) of the pre- or post activity.



Finally, the submethod can be inserted in the Activity dialog as a pre- or post activity:



- **Simulate actual time**

When this check mark is checked the actual time is consumed in simulation mode. Otherwise, the execution time of the steps in the activities is used.

Using times from previous run(s) for the planning of the next run

The lower part of the Advanced window is used for activity timing issues. It is possible to collect the times of each activity over several real runs. These data is used for a calculation with different algorithms and then inserted in the activity header to have a more precise planning of the next run.

- **Activity ID**

The Activity ID is the 'address' under which the times of an activity are stored. Enter a value or create a unique ID pressing the "Create ID" button. This could be used if you want to have the same planning time for a transport to *and* from a position. Use the same Activity ID in both transport activities and they will always be planned with the same time.

- **Automatic adjustment**

When this option is enabled the actual execution durations of each real run will be collected.



NOTE

No Activity times will be calculated in simulation mode. The times would not be representative.

- **Algorithm:**

Select one of the three possibilities to calculate the estimated duration for the next run out of the collected real run times: Arithmetic Mean, Minimum or Maximum.

- **Set Estimated Duration**

Clicking this tab will calculate the mean / minimum / maximum time (depending on the settings in 'Algorithm') out of the collected run data and pass this value to the "Duration" combo box of the activity.

- **Reset Activity Duration**

The collection of activity durations from run time will be cleared. This is the recommended way to clear the activity time database. If you only create a new Activity ID, the old data will remain in the database.



NOTE

The setting of times from previous runs has to be made by the programmer manually: You have to actively click the 'Set Estimated Duration' tab to get the values from previous runs calculated and passed into the 'Duration' field. This has to be done for every activity.

The reset of activity times can be made globally when clicking Reset Activity Durations in the Menu Method.

6.4 General rules



NOTE

The following rules will help you to improve the performance during runtime.

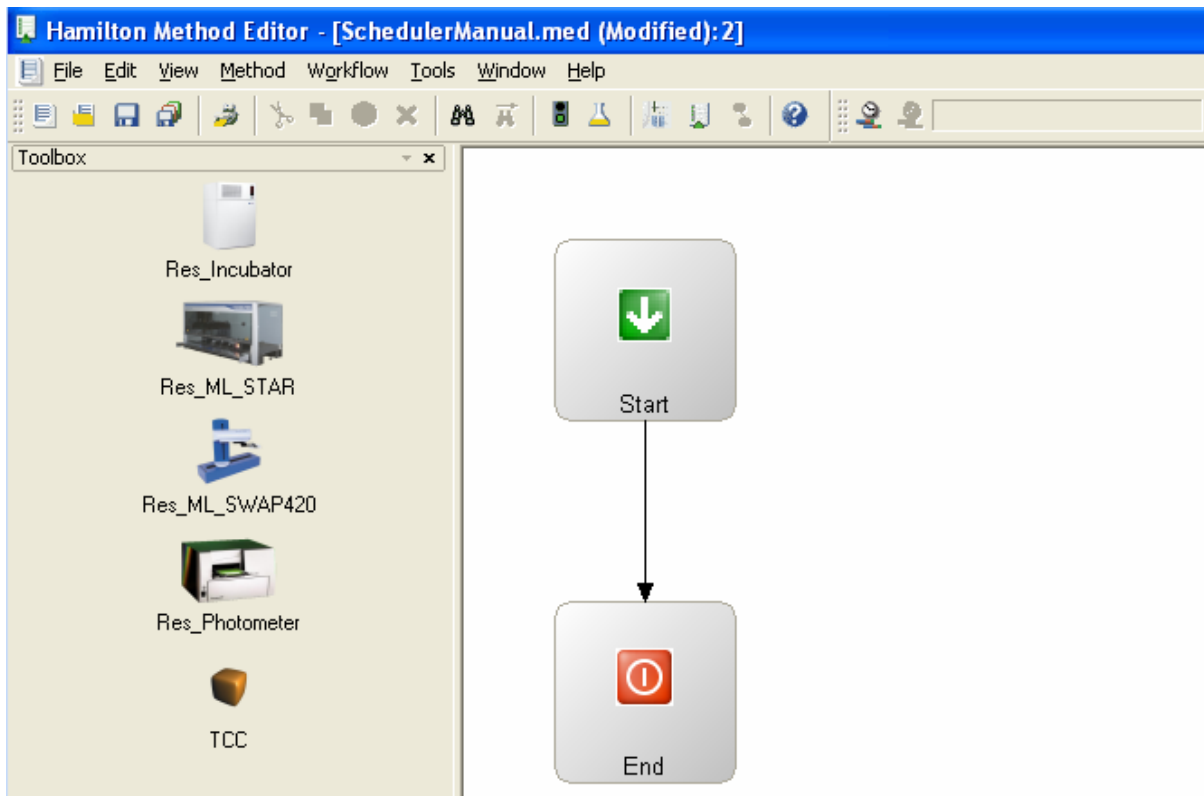
- Keep the resource specification as simple as possible.
- Start with a simple resource specification and refine it according to your workflow. Try to enable more parallel processes in this way.
- Consecutive steps of a method which use the same set of resources can be assigned to one activity.
- Create tasks (methods) which perform a complete process on one single entity like one sample, one plate, one batch, etc.
- Define as less activities as possible, i.e. merge consecutive activities requiring the identical set of resources.
- Whenever possible, specify the dependency **after the start** of the predecessor task.
- Avoid scheduling high capacity resources that never reach their maximum workload, e.g., do not define a Cytomat Incubator with 100 positions if the Cytomat has a built-in location control. In this case, it is sufficient to define the transfer position as a resource.

6.5 Activity Editor

The **Hamilton Activity Editor** is an easy, graphical oriented approach to the scheduling environment.



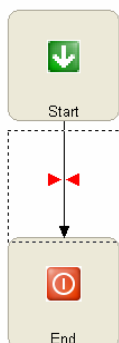
The Activity Editor can be activated using the toolbar icon shown above or via menu "View – Activity Editor". For an empty method the following screen might appear:



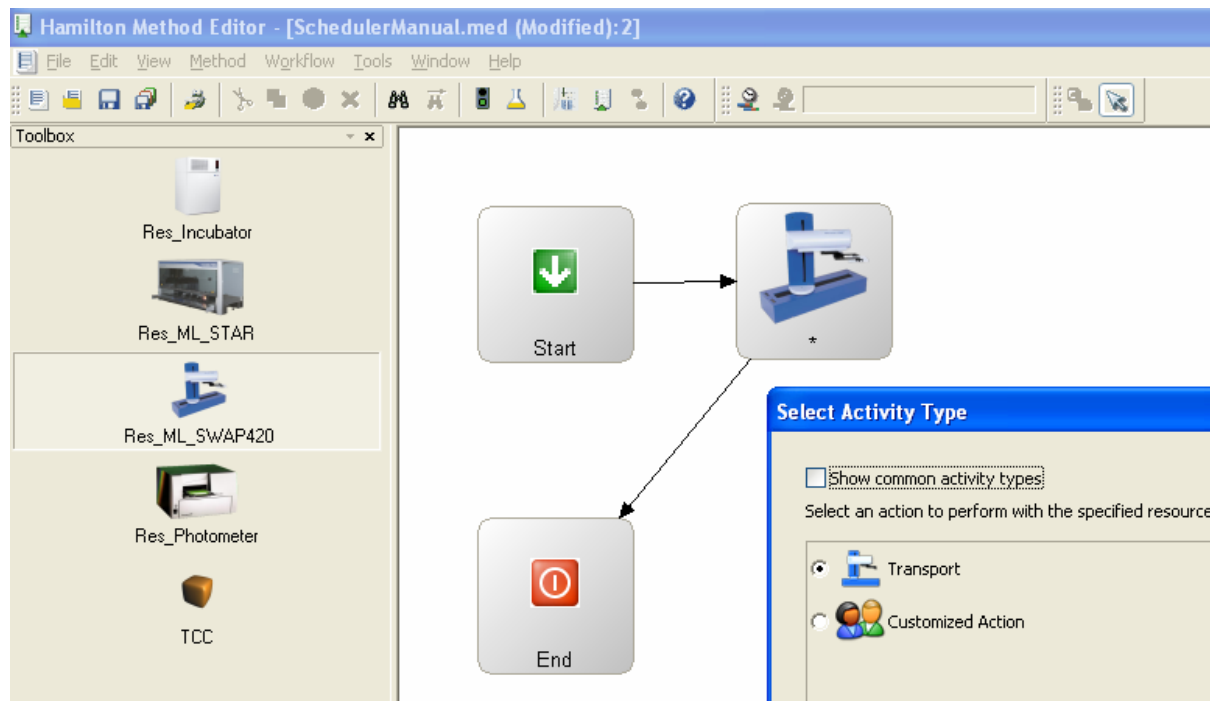
In the toolbox on the left all resources from your system deck layout are listed. If no image has been assigned to a resource, the resource will be visualized with a yellow dice (see the last icon in the toolbox). The window on the right shows an empty activity flow chart – only start and end symbols are shown.

6.5.1 Adding an activity

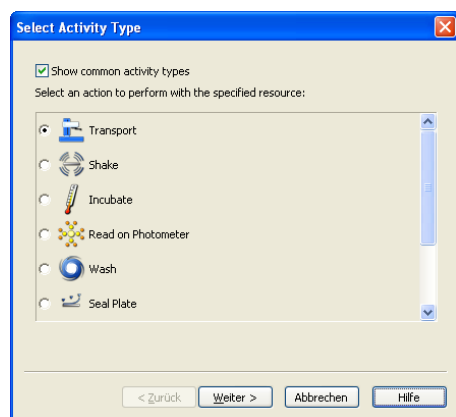
Activities are depending on resources. Therefore you have to select a resource from the toolbox and drag it into the activity flow chart window. The arrows are linked automatically in the order of adding blocks.



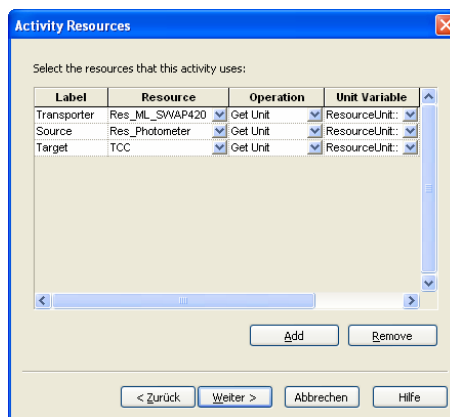
Move the dashed square to a place where you'd like the new activity icon to place and drop it there. The resource symbol is placed and the dialog box "Select Activity Type" appears. The result looks as follows:



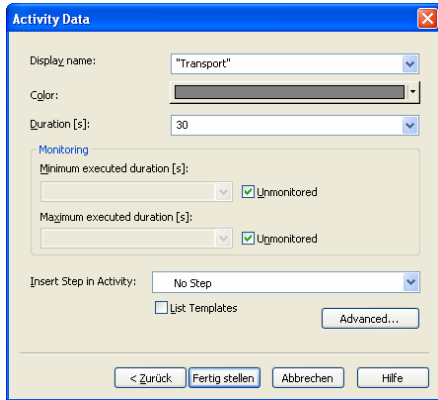
Depending on the device (see [Add Resource Dialog Box](#)) several predefined actions are available. Checking the 'Show common activity types' box, some more predefined activity types are shown. Select one from the given action types or use the "Customized Action" for a not predefined action. The information to be given in the "Next" and "Advanced..." steps are similar to the [Activity Dialog Box](#).



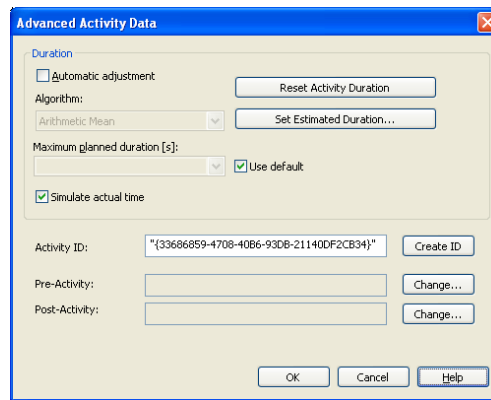
Select activity type



Insert used resources



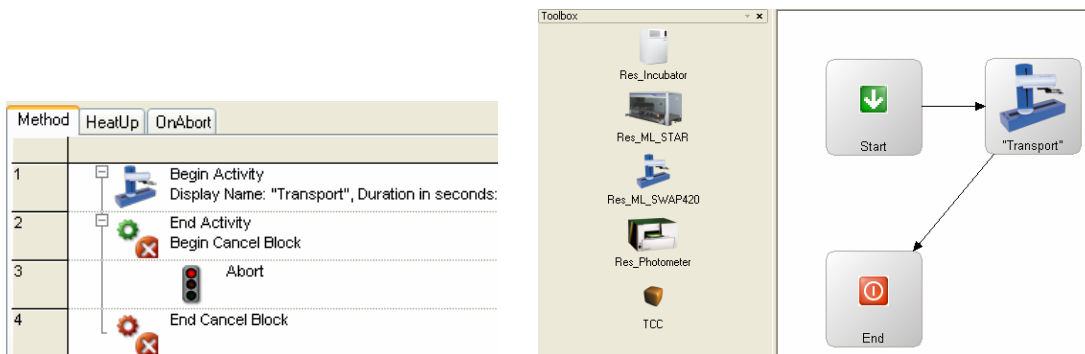
Define name / color / duration



Set advanced options

The "Insert Step in Activity" in the "Activity Data" window allows you to create an activity with a smart step inside: This makes it possible to create running methods out of the activity editor. In the method editor, you can change the activities made in the activity editor. Having "No Step" selected, an empty activity will be created. Empty activities are created to perform a fast and easy throughput or resource calculation.

The result looks as follows in the Activity Editor and in the Method Step Editor.

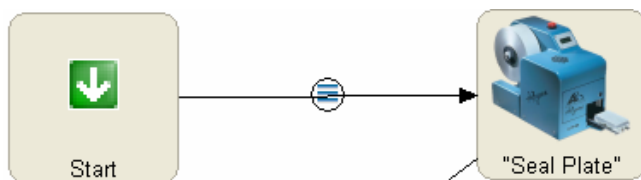


6.5.2 Editing an Activity

An activity can be edited within the Activity editor by double-clicking its icon. No changes are possible in the first dialog box since the selection of the Activity type can not be changed. For the following one's again see chapter [Activity Dialog Box](#).

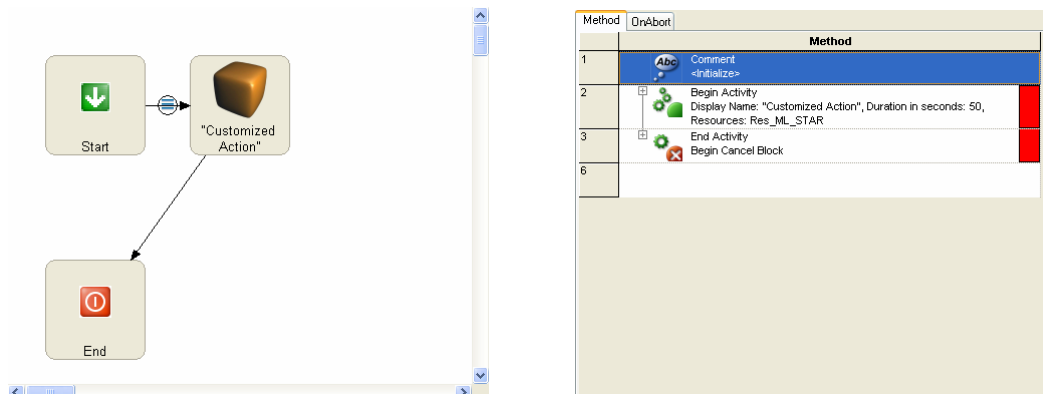
6.5.3 Deleting an Activity

An activity can be deleted within the activity editor by selecting it and pressing the "Del" key. The code within the activity will not be deleted but disabled. On the resulting arrow you can observe a circle with three bars which refers to the [Code outside of Activities](#): see the corresponding chapter.



6.5.4 Code outside of Activities

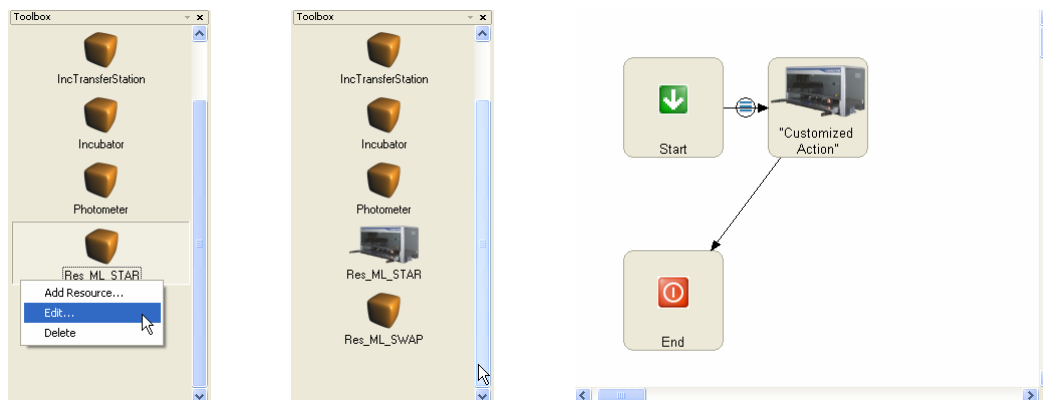
Code, even comments as well as disabled steps, outside of an activity is marked in the Activity Editor with a circle (with three bars) on the arrow between two activities. By double-clicking on this symbol, the view directly changes to the method step editor and on the first step of this code.



6.5.5 Changing image icon

When right-clicking on a resource icon within the toolbox select "Edit..." and in the following dialog box press the "Set Image" button. Select a picture of your choice. Closing and opening the activity editor again will take effect in the chart view.

Another way to change an image is to go to the system deck layout and change the image by selecting "Edit Resources" from the context menu.



6.5.6 Adding / Editing / Deleting Resources

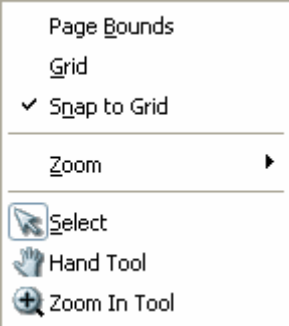
Right-clicking within the toolbox frame allows you to add resources. The known dialog box "Add resource" with empty fields appears. See chapter [Add Resource Dialog Box](#) for detailed information.

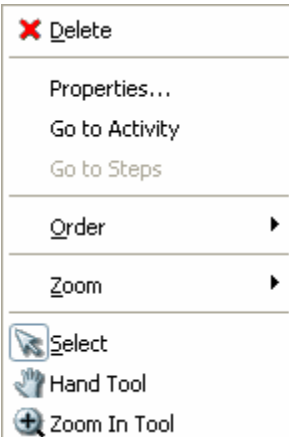
Right-clicking on a resource allows you also to delete a resource.

6.5.7 Context menu's in activity flowchart

The context menus below are available in the activity flowchart depending where the right mouse button was clicked.

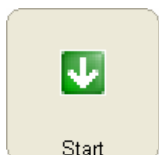
Right click on a free window area, on an activity, on code symbol on an arrow, or on an arrow to get the following context menu commands available.

	Page Bounds	show page bounds
	Grid	show grid
	✓ Snap to Grid	snap activity icon to grid
	Zoom	Select zoom size
	Select	activates the select tool
	Hand Tool	activates the hand tool to move the view
	Zoom In Tool	activates the zoom in tool to enlarge the view

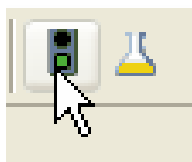
	Delete	deletes the selected activity
	Properties...	edit properties of an activity
	Go to Activity	switch to the selected activity
	Go to Steps	switch to the selected code step
	Order	Set's display options (bring to front /back)
	Zoom	Select zoom size
	Select	activates the select tool
	Hand Tool	activates the hand tool to move the view
	Zoom In Tool	activates the zoom in tool to enlarge the view

6.5.8 Running the method

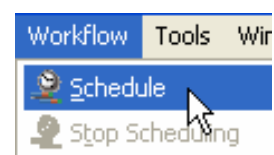
Your method is now ready for scheduling. To start the scheduling process, there are different possibilities.



Doubleclick 'Start'

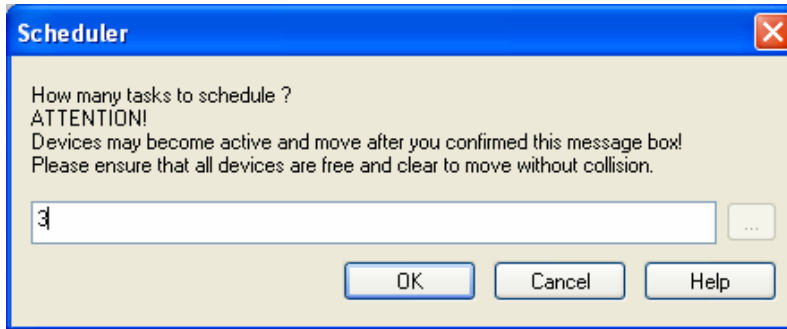


Click the traffic light



Choose Workflow → Schedule

The system now asks for the number of tasks that shall be scheduled:



After a number is entered, click ok and the software will calculate a plan. Make sure the Scheduler view is activated in the Run control window:



6.6 Defining a Workflow using the Graphical Workflow Section

A workflow is an arrangement of several tasks for scheduling. This section describes defining a workflow using the graphical workflow section. The result is a workflow. Normally, all processes run automatically and do not need any user interventions during runtime because these are rather unplannable (user is having lunch, system waits for input...).

6.6.1 Creating a new Workflow

1. In the “Method” editor, click the menu “File” – “New” – “Workflow”. The “Save as” dialog box appears.
2. Type in a file name and click “Save”. The graphical workflow section with the start task appears.

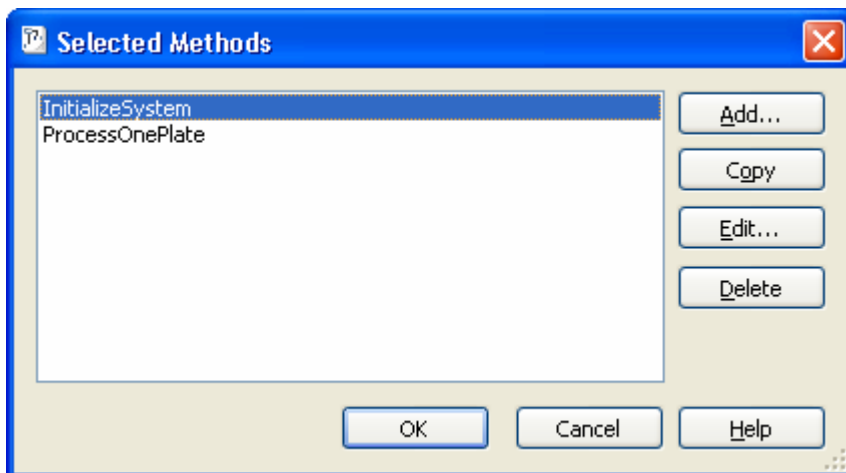
6.6.2 Selecting Methods for the Workflow



NOTE

To use a method in a workflow, it must be selected and linked to the workflow. It is possible to use an existing method in multiple workflows (e.g. InitializeSystem)

1. On the “Workflow” menu, click “Selected Methods”. The “Selected Methods” dialog box with already selected methods appears.



2. Click “Add”. The “Selected Method” dialog box appears.
3. Type in the name of the method or search for it using the “Browse” button.
4. Click “OK”. The “Selected Method” dialog box is closed and the selected method is added to the list.
5. Add all methods that are used for the workflow (repeat steps 2 to 4).
6. After you have selected all methods, click “OK”. The “Selected Methods” dialog box is closed and the selected methods are available in the toolbox.

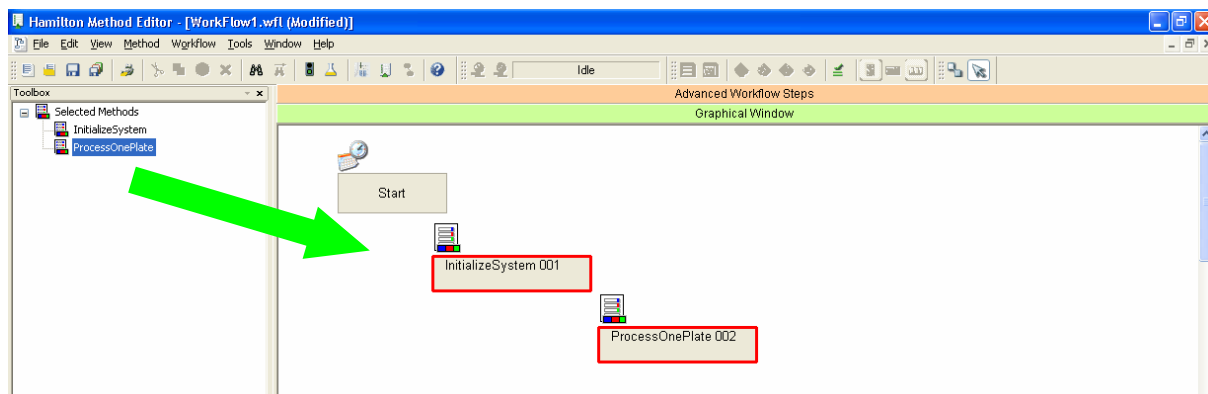


NOTES


It makes sense to split-up methods to prevent unnecessary repetition of activities. E.g., if the init or a loading procedure is not in its own method, the init would be processed multiple times as well. Using a specific init task, results in only the plate processing executed multiple times.

Adding the selected methods to the workflow

Place the methods in the “Graphical Window” box using “drag and drop”. Arrange the methods according to the logical flow.

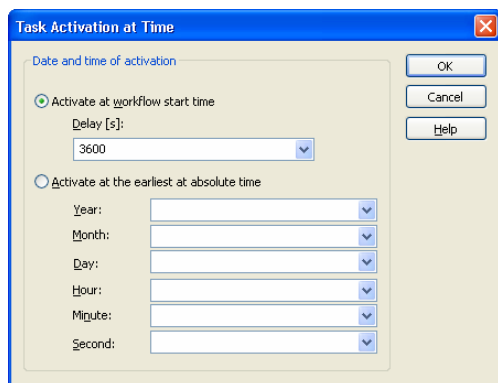


6.6.3 Defining Task Dependencies

1.  Click the “Add Activation Link for Task” icon.
2. Click the START icon. Then, click the next task. The “Task Activation at Time” dialog box appears. In this box, you can select if the start shall be delayed for a specific time OR the start shall happen at a specific date and time.

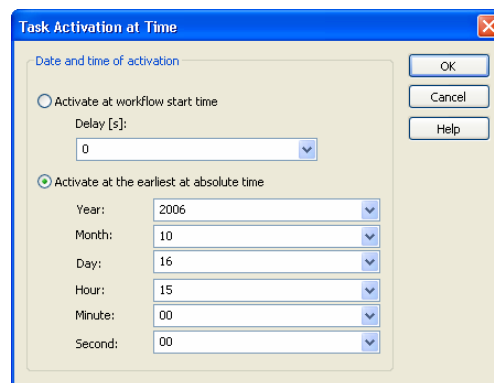
For an immediate start, select *Activate at workflow start time* and set a delay of 0 seconds.

Examples:



*Start at workflow start with delay
e.g. to heat up an incubator*

OR

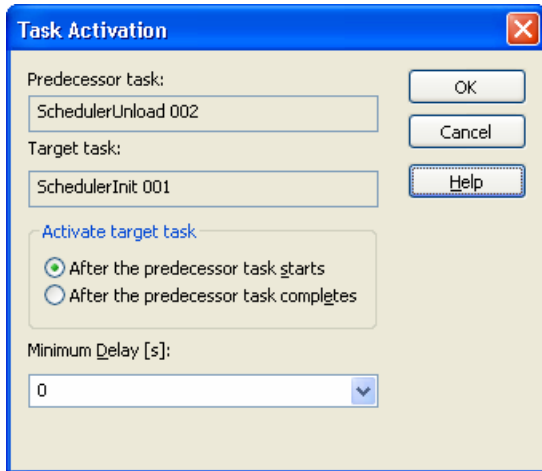


Start at specific date and time

3. Proceed in the same way to link all tasks. Unconnected tasks are shown with a red frame.

6.6.4 Task Activation Dialog Box

The “Task Activation” dialog box shows the dependencies between two tasks. By doing a double left click on a link between tasks, you get the dialog for that link.



- **Predecessor task**

This field is read only since the Predecessor task is already defined.

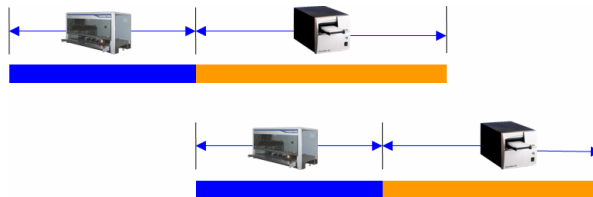
- **Target task**

This field is read only since the Target task is already defined.

- **Activate target task**

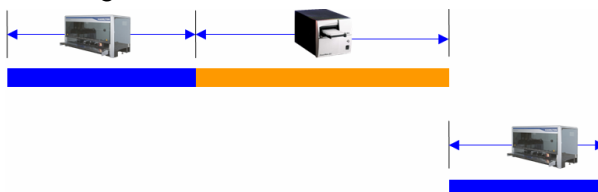
After the predecessor task starts

The target task is activated after **activation** of the predecessor task. The two tasks will be executed in parallel. The predecessor *has higher priority* to resolve a possible resource allocation conflict between the tasks.



After the predecessor task completes

The target task is activated after **termination** of the predecessor task.



- **Minimum Delay(s)**

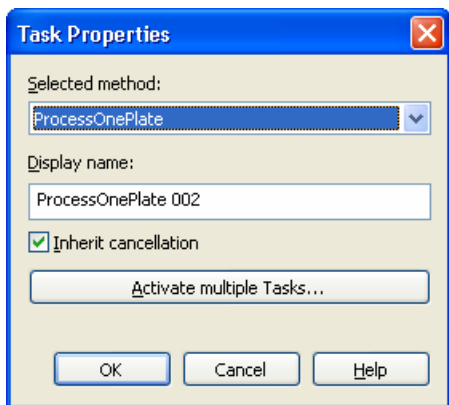
Specifies a delay in seconds between two tasks.



6.6.5 Activating multiple Tasks

1. For multiple activation of a task, e.g. using the assay for several plates, double click on the desired task. The "Task properties" dialog box appears.

It is also possible to activate multiple task during runtime, using the advanced workflow editor and the 'Activate task' step from the Scheduler steps.



- **Selected Method**

The Select method is shown.

- **Display name**

The Display name is shown.

- **Inherit cancellation**

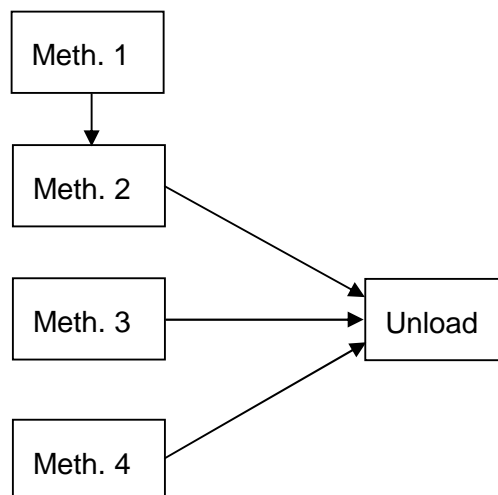
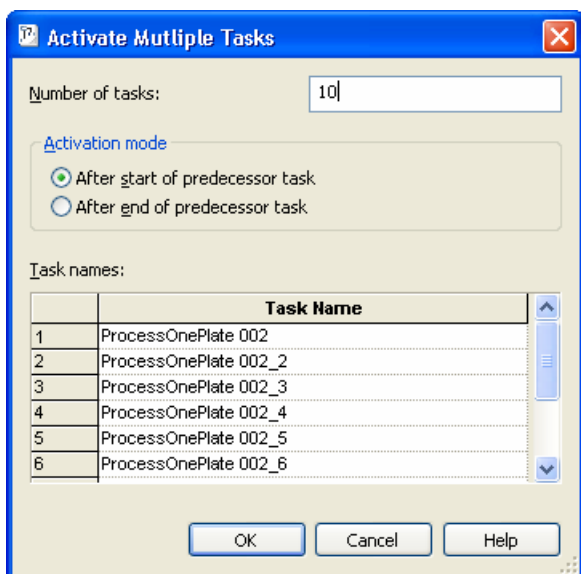
Task will be cancelled if predecessor task is cancelled.

2. Click on "Activate Multiple Tasks..." and enter the number of repetitions. The selected task will be executed multiple times.



NOTES

Each activation link defines a constraint. If there is more than one activation link to the target task, the task is activated when all constraints are fulfilled. In the example below: the Unload will be executed when Method 2, 3 and 4 are finished.

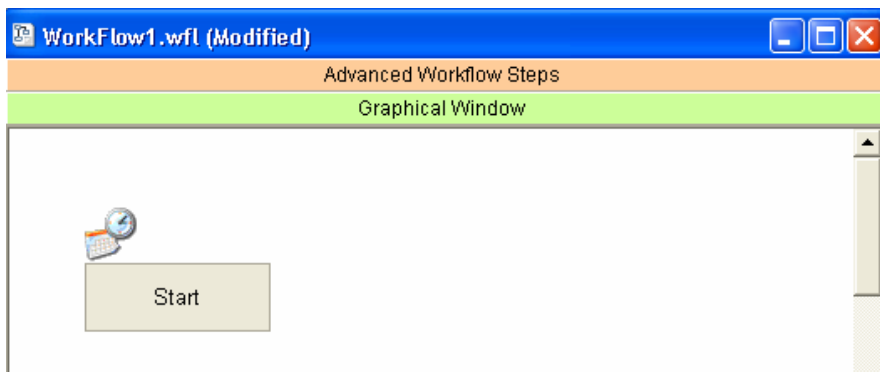


6.7 Defining A Workflow Step by Step

A workflow is an arrangement of several tasks for scheduling. Suppose you want to create a workflow dynamically, e.g. with a variable number of plates with one task per plate. This section describes programming using the *Advanced Workflow Steps* section to do so. The result is an individualized workflow. User interventions (e.g. a User Input step) can be integrated and are then possible at runtime.

6.7.1 Creating A New Workflow

1. In the "Method" editor, click the menu "File" – "New" – "Workflow". The "Save as" dialog box appears.
2. Type in a file name and click "Save". The graphical workflow section with the start task appears.
3. Click on the bar "Advanced Workflow Steps".

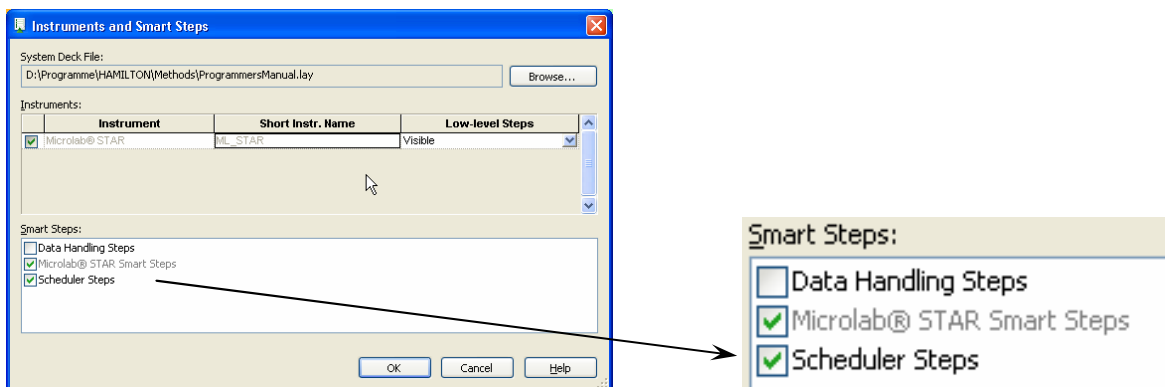


Scheduler steps (together with general steps) are used to individualize workflows. User interventions can be integrated and are possible at runtime.

- A workflow can be created dynamically, e.g., dependent on user inputs or work lists.
- Tasks can be added in loops.
- Tasks can be executed depending on conditions.
- Tasks can be executed depending on variables.

If the scheduler smart steps are not available in the toolbox, add them as follows:

1. Go to menu "Method" – "Instruments and Smart Steps"



2. Add the "Scheduler Steps" by marking the check box
3. Click "Ok"

6.7.2 Defining a Workflow

1. In the Method editor (workflow active), change to the “Advanced Workflow Steps” section.
2. Select all methods using “Select Method” steps from toolbox "Scheduler Steps". Refer to section [Select Method Step](#) for details.
3. Define the workflow step by step.
 Activate the tasks and define the task dependencies using “Activate Task” steps from toolbox "Scheduler Steps". Refer to section [Activate Task Step](#) for details.



NOTES

Using the scheduler steps and all the other steps in the toolbox, it is possible to specify workflows corresponding to your individual requirements. The programming step by step procedure is similar to other method programming

6.7.3 Variable Controlled Scheduling (Example)

In this workflow, the incubation temperature is selectable.

1. Change to the “Advanced Workflow Steps” section.
2. Add a “User Input” step and specify the parameters as follows:

The screenshot shows the 'User Input - Edit' dialog box with the following configuration:

- Dialog title:** "User Input: What is the incubation temperature?"
- Buttons:** Only 'OK' button
- Default button:** OK
- Return value:** (empty)
- Sound:** (empty)
- Timeout [s]:** Infinite
- Input request(s):**

	Variable	Prompt	Type	Default	[Minimum]	[Maximum]
1	IncTemp	"What is the incubation temperat	Integer	37	25	60

For details of dialogs refer to chapter Additional Functions for the Scheduler.

6.8 Run Control

A workflow can be scheduled to identify all tasks and activities in a workflow and organize them without entering the Run Control program (HxRun). This can be done on the menu "Workflow – Schedule" or via the toolbox "Schedule workflow". A warning is displayed (to disable this warning, refer to section 12.4.4):



During this scheduling run, any steps not within an Activity Block may be executed! (❶)

While analyzing the workflow the scheduling can be stopped selecting "Stop scheduling" in the toolbox (❷) or in the workflow menu.



When a workflow is loaded into the Run Control program (HxRun), a scheduling run is immediately done to identify all the methods in the workflow, and organize them correctly. After this scheduling run, the loaded workflow (and the methods in the workflow) may be executed any number of times, without the need to go through this scheduling run again.

6.8.1 Starting the Scheduler from Method Editor

1. In the "Scheduler" toolbar, click the "Schedule workflow" button. The workflow is scheduled and both the task and resource view appear (plans).
2. Check the course of the tasks with the activities for resource conflicts to see if their resource definitions are correct.



NOTE

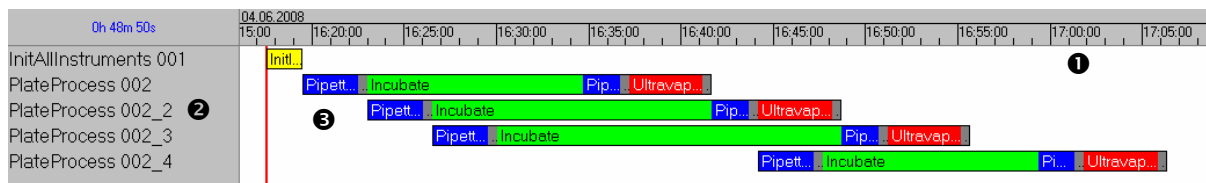
Giving each resource an own color will simplify the understanding of a plan.

6.8.2 Starting the Scheduler from Run Control

After invoking the Run Control by double clicking the 'Microlab STAR Run' Icon on the desktop, a file (Workflow or Method) can be started by selecting the File. To do so, select *File* and Open the Workflow / method that should be executed.

6.8.3 Task View

The task view on the upper area of the Run control gives you an overview at what time each task ③ is planned to be started. The time ① is counted up until the workflow is started. The box on the left side ② provides information about the name of the task.



- **Time axis ①**

Depending on the Schedule View Options, the Time bar shows the actual or an absolute time. As long as the workflow is not started, the time bar changes.

- **Task list (methods) ②**

The task list on the left side shows the task names. The tasks are indexed for a better overview.

- **Chart with the tasks ③**

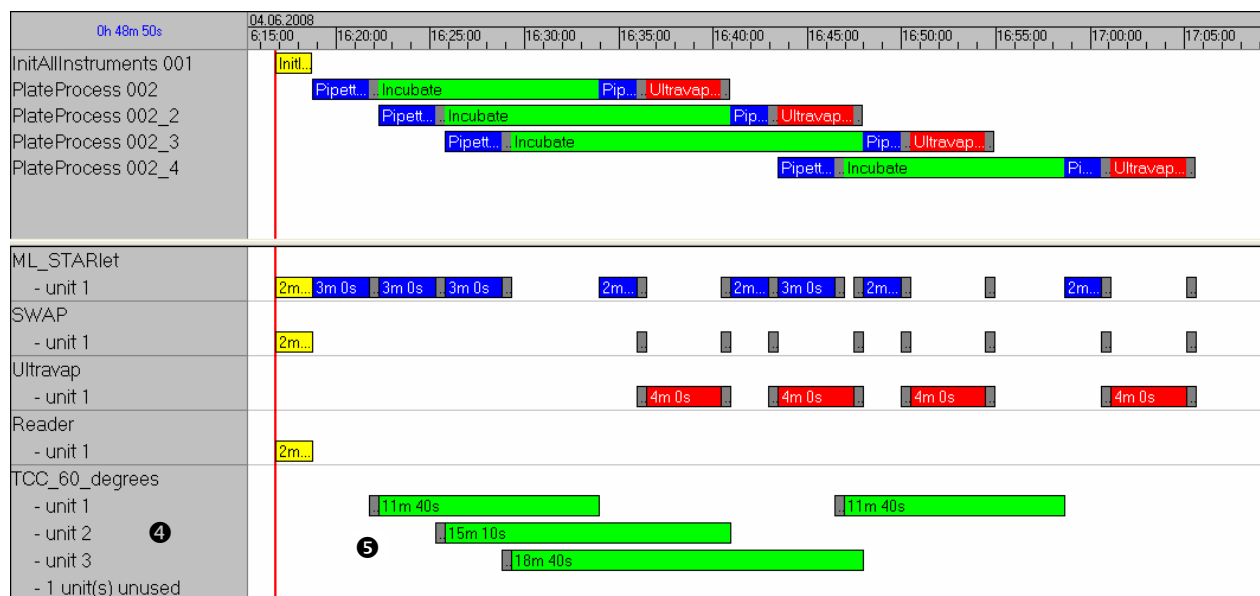
The tasks are shown in the plan. A task consists of one or several activities which are displayed in the order to be proceeded. The real times are updated during the run.

- **Indicator for the current position of the run and the corresponding time**

Before the run is started, the red time indicator remains on the left side and the time axis ① shifts from right to left. (Assumption: absolute time/date format, see Schedule View Options).

After the start of the run, the red time indicator moves to the right while the time axis ① is fixed (see above).

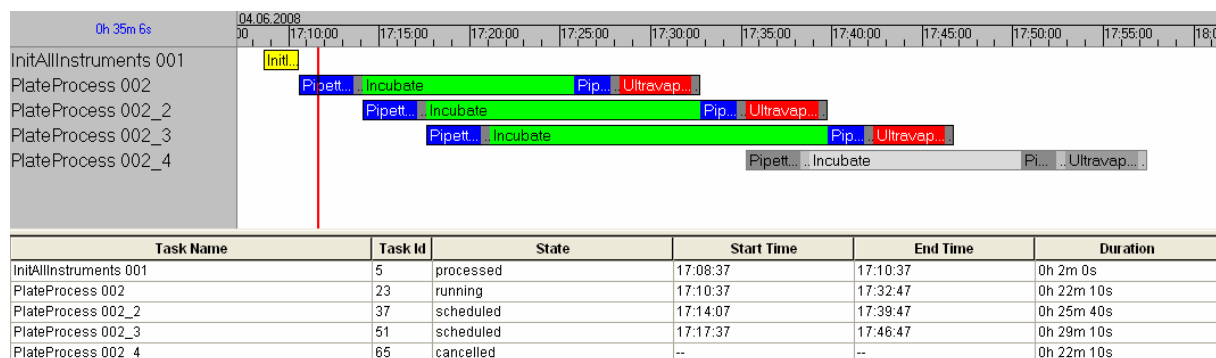
6.8.4 Resource View



- Resource list ④**
 The resource list shows the use of resources during the run. The number of unused units of a resource is also displayed (e.g. TCC_60_degrees → 1 unit is unused).
- Chart with the resources ⑤**
 Resources are displayed in the time axis with the specified (assumed) running time (duration) and the corresponding starting and ending times associated a particular activity use of that resource. The real running times are updated during the actual run.

6.8.5 Task List

Instead of the resource view, it is possible to display a list with all tasks of a workflow.

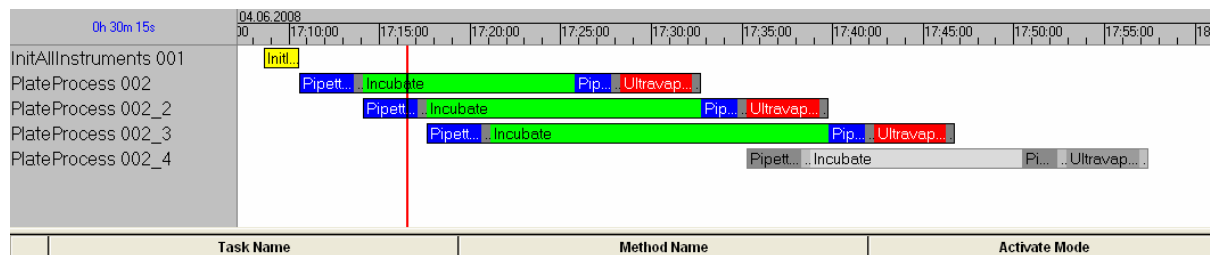


The task list provides a view in tabular form with the following information:

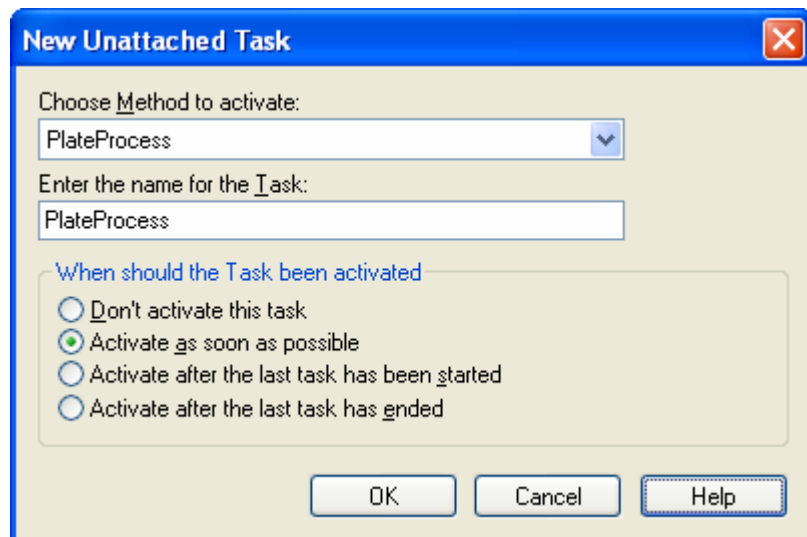
- Task name
- Task Id
- State (scheduled / running / processed / cancelled)
- Start time
- End time
- Duration

6.8.6 Unattached Tasks

Instead of the resource or task list view, it is possible to switch to the 'unattached tasks' where several actions are possible:



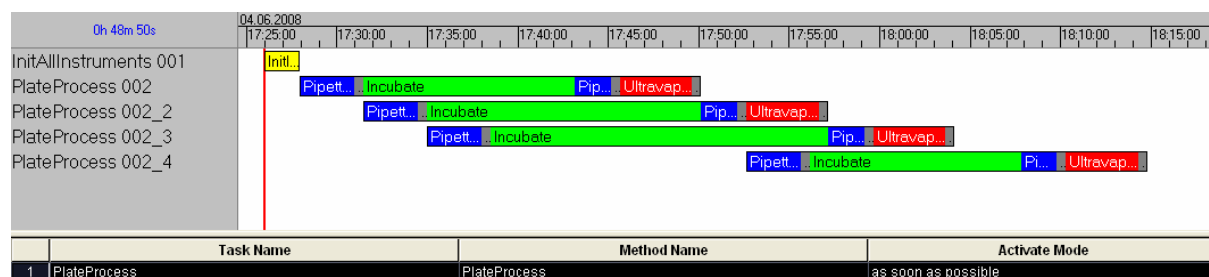
After a right mouse click in the Unattached task window, it is possible to add a new task to the run:



After selecting a method to be activated, an activation mode can be selected:

- Don't activate this task → no action
- Activate as soon as possible → The method is inserted at the earliest position
- Activate after the last task has been started → Existing tasks are executed first
- Activate after the last task has ended → e.g. for Unload task, waits to all tasks end

The method will now be visible in the Unattached Task window. It is also possible to add more than one task.



A right mouse click starts the context menu with the options:

New (which executes the action described above)

Schedule Unattached Tasks (which integrates the Unattached task(s) in the plan) and

Delete All Unattached Tasks (which will remove the selected task from the list)

6.8.7 Optimizing a Run



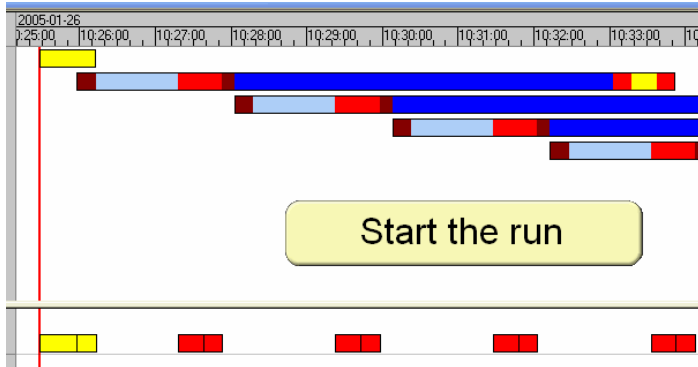
NOTE

Optimizing of the run can be done by

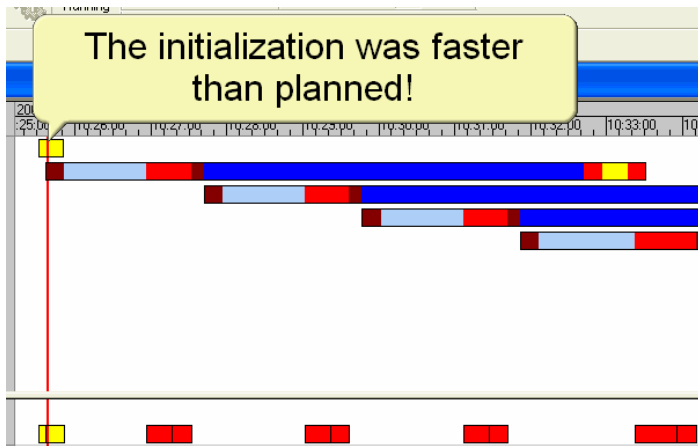
- *adding or changing hardware resources*
- *changing activity times e.g. incubation, shaking (but still within the limits of your protocol/specification)*
- *defining the scheduler resources in a smaller, more precise manner than an instrument and/or*
- *by defining the activities differently (e.g. the iSwap loads four plates at once using a stacker instead of individually loading every plate).*

6.8.8 Real Time Dynamic Adaptions (Example)

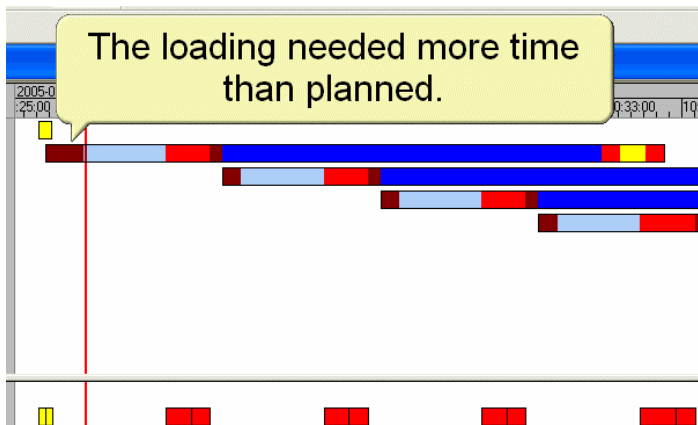
The scheduler software dynamically changes the plan depending on the real time uses of the resources.



At starting time, the calculated schedule with the values from the activities is displayed.



If an Activity is completed earlier than planned, the start times of the following tasks may be *shifted to the left*.



If an Activity needs *more* time than planned, the following tasks may be *shifted to the right* (start later).

7 Test Run



ATTENTION

Again: to perform a scheduling process, ALL OF THE METHODE CODE outside of Activity Blocks IS EXECUTED. This is true both when scheduling in the Method Editor and when loading a workflow into the Run Control program. This is also true when executing a reschedule on a method that hasn't previously (during the current Run Control execution) been scheduled. The result of the scheduling process running the code is that any INSTRUMENT COMMANDS that are not inside of an Activity Block WILL BE EXECUTED. To prevent the instruments from moving in an unpredictable way during this reschedule the user should:

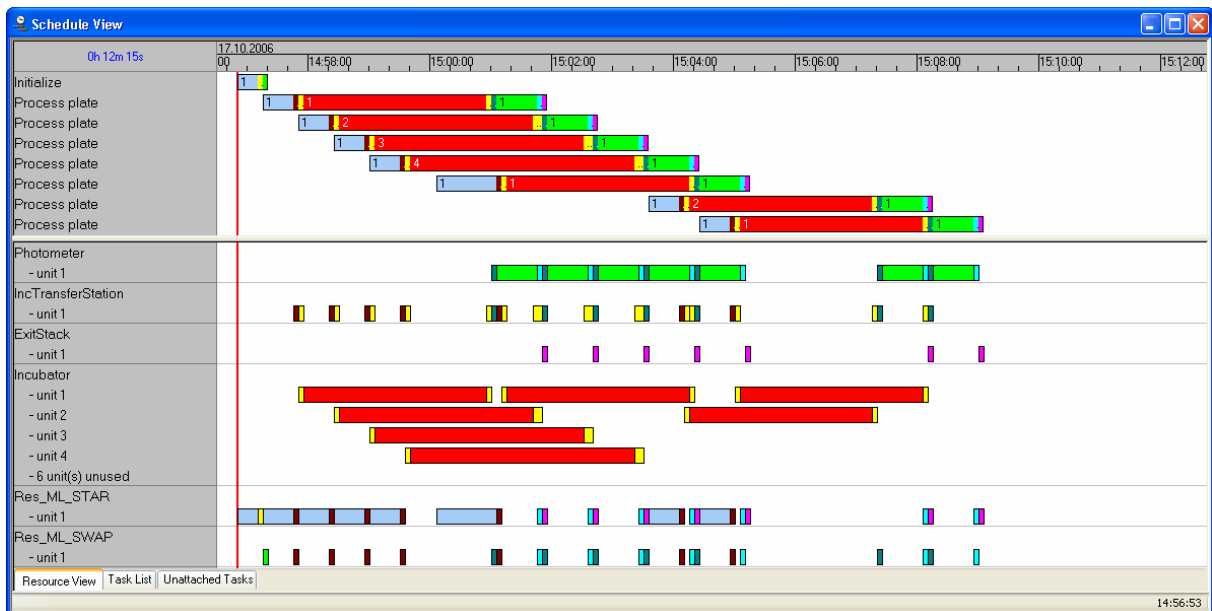
After the workflow definition has completed and has successfully been run in the simulation mode, it should be **tested** with the actual hardware.

1. Prepare labware and liquids.
2. If necessary, reduce loop counters to a small number of loops.
3. Start the Hamilton Run Control.
4. Switch off the simulation mode.
5. Check connections to all instruments and make sure they are switched on.
6. Open the workflow.
7. Run the workflow.

8 Production Runs

8.1 Starting and Running

1. Prepare labware and liquids.
2. Start the Hamilton Run Control.
3. Check that the simulation mode is switched off.
4. Check connections to all instruments and make sure they are switched on.
5. Open the workflow. The workflow is scheduled automatically.
6. Click the start button or press “F5”. If the “Schedule View” is selected, you can follow the running (red line).



Refer to section [Task View](#) for a description.

8.2 Interventions at Runtime

8.2.1 Removing or Cancelling a Task

In the “Schedule View”, you are able to...

- Remove tasks using the “Remove” function of the context menu (right-click the task). The task is moved to the “Unattached Tasks” tab. The start parameter is set to “Do not activate”.
- Cancel tasks using the “Cancel” function. The task is still visible in the view.

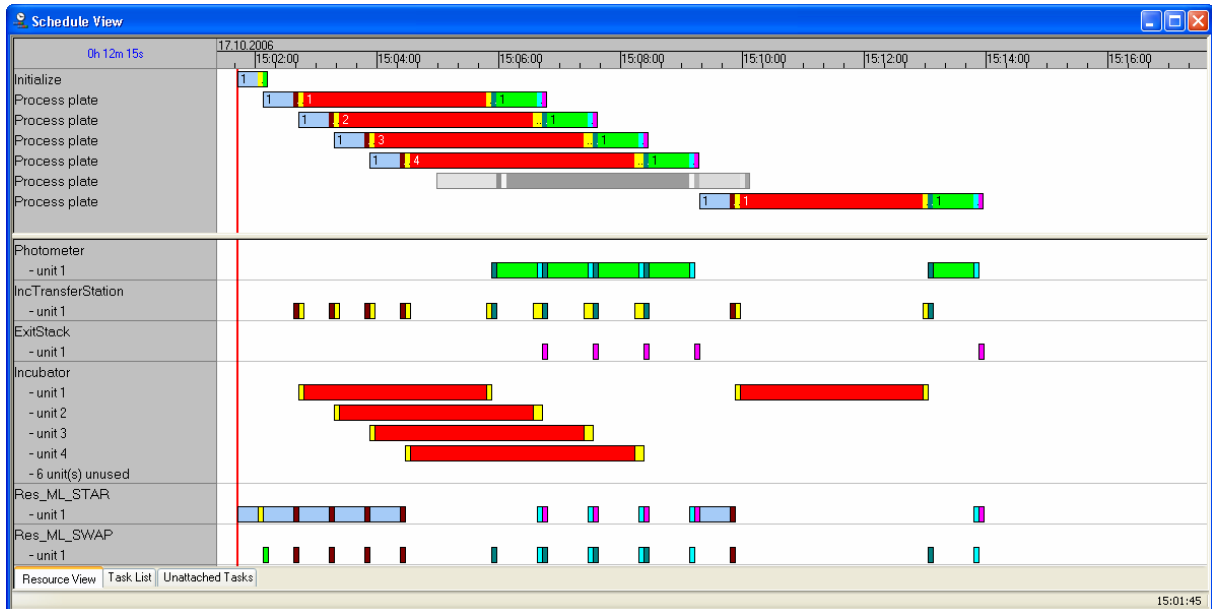


NOTE

The rescheduling function in the context menu of the “Schedule View” is useful, if tasks have been canceled or removed. This is only true when an Error Handler does not have it’s default settings (Abort & Inherit cancellation)

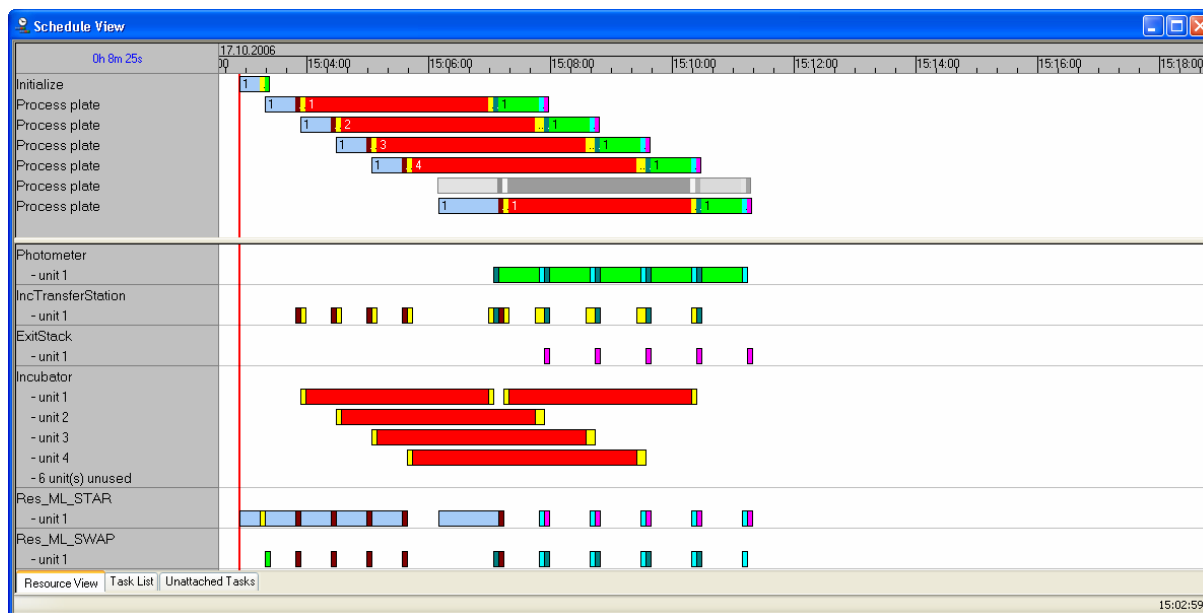
8.2.2 Rescheduling

A plan with canceled and or removed tasks may lead to gaps in the task and resource plan. The plan above with one task canceled and one removed might lead to the following schedule view:



In this case, a rescheduling will help to rearrange the tasks in a better way. Move the cursor to the task window and perform a right mouse click.

Then, select “Reschedule...” to optimize the plan:



ATTENTION

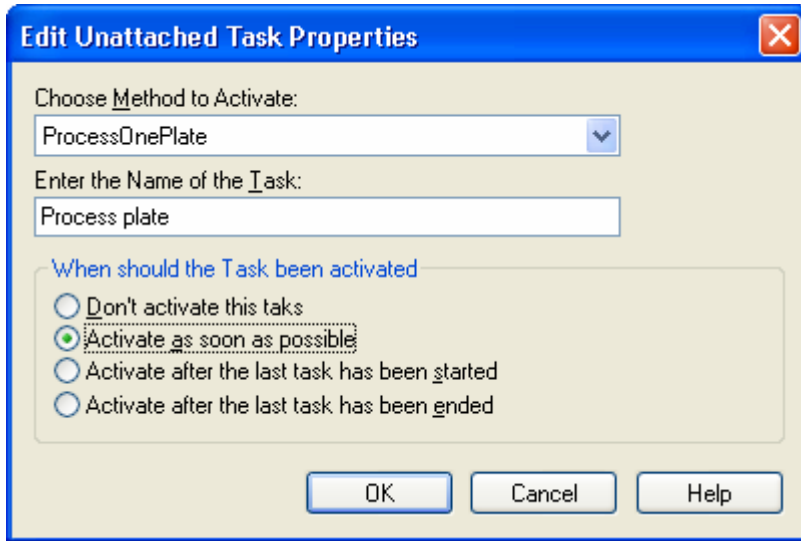
Rescheduling will cause a scheduling identify of any method that is loaded and has not previously been identified for scheduling. This method identification may cause statements not within Activity Blocks to be executed. If these statements are device commands, unexpected movement of the hardware may result.

8.2.3 Adding, Editing and Deleting Unattached Tasks

During a run, it is possible to insert additional tasks to a running or scheduled Method/Workflow. This option is accessible by clicking in the Unattached Tasks window of the Run Control. Task that are added, edited or removed in this area are called ‘Unattached tasks’.

1. In the “Schedule View”, select the “Unattached Tasks” tab. The list of unattached tasks appears.
 - You can add tasks to the list using the “New” function of the context menu (right-click the task) or by double-clicking into the free space below the list. In the “Task Properties”, you can set the name and the start parameter of the task. Set the start parameter to “Do not activate”, if the task should not be scheduled.
 - You can remove all tasks from the list using the “Delete all unattached tasks...” function of the context menu.
 - You can remove a single task from the list using the “Delete” function of the context menu.
 - You can change (edit) the task parameters using the “Edit Properties” function of the context menu or double-clicking the appropriate task.
2. In the context menu, select “Schedule Unattached Task...”. All tasks are activated according to the settings in the “Task Properties”.

8.2.4 Edit/New Unattached Task Properties Dialog Box



- **Choose Method to Activate**

Defines the method which should be activated. Only methods that are registered in the workflow are selectable. (Using the “Selected Methods” dialog box or a “Select Method” step.)

- **Enter the Name of the Task**

Defines the task name used in the “Schedule View” after the unattached task is added to the scheduled plan.

- **When should the Task be activated**

Defines the mode used for activation of the task:

Don't activate this task

This task will not be added to the schedule plan.

Activate as soon as possible

Activates the task immediately, corresponding to the used resources.

Activate after start of last task

Activates the task after the start time of the last scheduled task within the current scheduled plan.

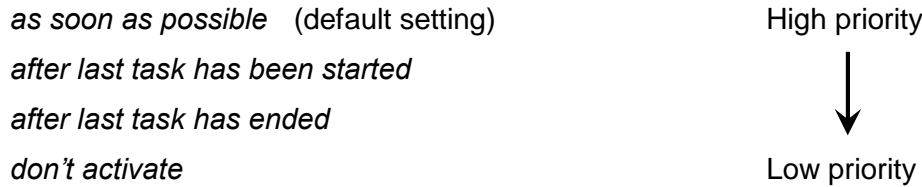
Activate after end of last task

Activates the task after the end of the last scheduled task within the current scheduled plan.

8.2.5 Grouping and ordering of unattached tasks

Unattached tasks will be grouped to their activation modes (see column *Activate Mode*)

The priority at the activation of tasks is like this:



Inside each group, the displayed order will be the order in which the tasks are activated. For example, the task *'Process one plate 4'* will be activated before the task *'Process one plate 5'* will be activated.

The order within any group can be changed by clicking the group number and moving it up- or downwards within the group:

	Task Name	Method Name	Activate Mode
1	Process plate 4	ProcessOnePlate	as soon as possible
2	Process plate 5	ProcessOnePlate	as soon as possible
3	Process plate 6	ProcessOnePlate	as soon as possible
4	Process plate 7	ProcessOnePlate	after last task has been started
5	Process plate 8	ProcessOnePlate	after last task has been started
6	Process plate 9	ProcessOnePlate	after last task has been started
7	Process plate 10	ProcessOnePlate	after last task has been ended
8	Process plate 11	ProcessOnePlate	after last task has been ended
9	Process plate 12	ProcessOnePlate	after last task has been ended
10	Process plate 1	ProcessOnePlate	don't activate
11	Process plate 2	ProcessOnePlate	don't activate
12	Process plate 3	ProcessOnePlate	don't activate



Whenever a task will be changed from one group to the other it will be moved to the end of the corresponding group:

	Task Name	Method Name	Activate Mode
1	Process plate 5	ProcessOnePlate	as soon as possible
2	Process plate 4	ProcessOnePlate	as soon as possible
3	Process plate 6	ProcessOnePlate	as soon as possible
4	Process plate 7	ProcessOnePlate	after last task has been started
5	Process plate 8	ProcessOnePlate	after last task has been started
6	Process plate 9	ProcessOnePlate	after last task has been started
7	Process plate 10	ProcessOnePlate	after last task has been ended
8	Process plate 11	ProcessOnePlate	after last task has been ended
9	Process plate 12	ProcessOnePlate	after last task has been ended
10	Process plate 1	ProcessOnePlate	don't activate
11	Process plate 2	ProcessOnePlate	don't activate
12	Process plate 3	ProcessOnePlate	don't activate



8.2.6 Schedule unattached tasks

As long as a task is not scheduled into the plan, it will remain in the 'unattached task list'. To schedule such a task, perform a right mouse click in the "Unattached Tasks" window and select 'Schedule unattached tasks'.

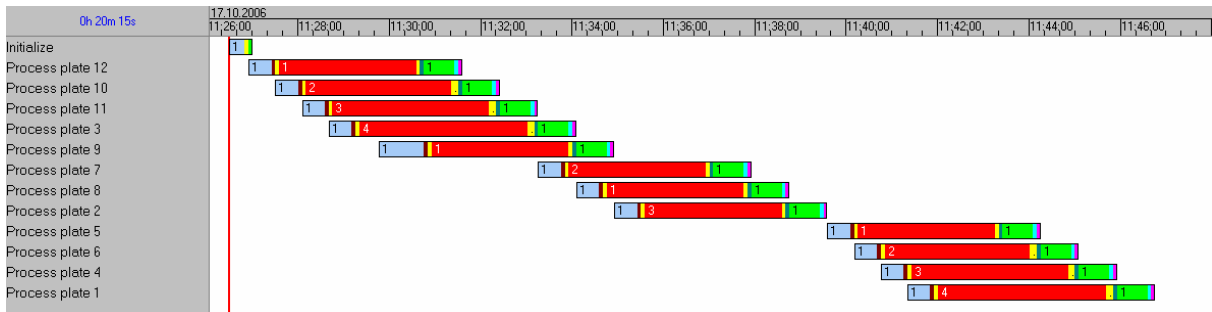
	Task Name	Method Name	Activate Mode
1	Process plate 10	ProcessOnePlate	as soon as possible
2	Process plate 12	ProcessOnePlate	as soon as possible
3	Process plate 11	ProcessOnePlate	as soon as possible
4	Process plate 3	ProcessOnePlate	as soon as possible
5	Process plate 9	ProcessOnePlate	after last task has been started
6	Process plate 7	ProcessOnePlate	after last task has been started
7	Process plate 8	ProcessOnePlate	after last task has been started
8	Process plate 2	ProcessOnePlate	after last task has been started
9	Process plate 5	ProcessOnePlate	after last task has been ended
10	Process plate 6	ProcessOnePlate	after last task has been ended
11	Process plate 4	ProcessOnePlate	after last task has been ended
12	Process plate 1	ProcessOnePlate	after last task has been ended



Process one plate 12 – 10 – 11 – 3" were selected to be **activated as soon as possible** and a reschedule unattached tasks was performed.

Then the following four tasks "9 – 7 – 8 – 2" were selected to be activated **after last task** – task 3 is meant – **was started** and a "reschedule unattached tasks" was performed.

The last four tasks "5 – 6 – 4 – 1" were then selected to be activated **after the last task** – now that is task 2 – **was ended**.



9 Rules for Programming

9.1 General rules



NOTE

The following rules will help you to improve the performance during runtime.

Create tasks (methods) which perform a complete process on one single entity like one sample, one plate, one batch, etc.

Keep the resource specification as simple as possible.

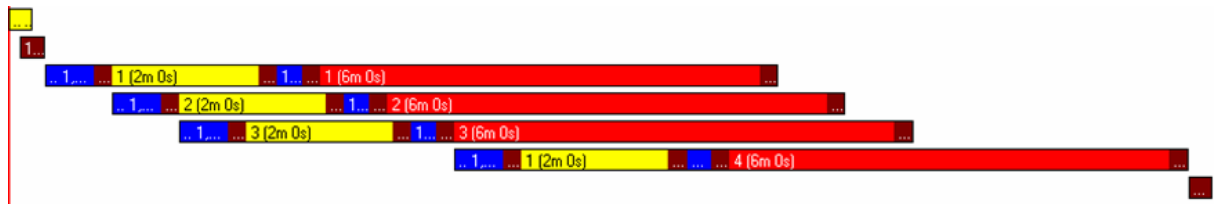
Start with a simple resource specification and refine it according to your workflow. Try to enable more parallel processes in this way.

Define as less activities as possible, i.e. merge consecutive activities requiring the identical set of resources.

Whenever possible, specify the dependency **after the start** of the predecessor task.

Always check the scheduling results after a modification.

Try to get a ‘stair’ of tasks as regular as possible by increasing the delay between succeeding tasks and/or by increasing the duration of some specific activities (e.g shaking and incubation times).



Avoid scheduling high capacity resources that never reach their maximum workload, e.g., do not define a Cytomat Incubator with 100 positions if the Cytomat has a built-in location control. In this case, it is sufficient to define the transfer position as a resource.

Consecutive steps of a method which use the same set of resources can be assigned to one activity.

Use your best guess or measured run durations in activities while programming. After some real runs you can use the automatic adjustment to fill in the actual times.



ATTENTION

- Only steps that need resource must be assigned to an activity.
- All steps outside of an activity are executed during scheduling.

9.2 Rules for Global and Local Variables

9.2.1 Default Definitions of Variables

Variables defined in a sub method are local variables and have no effect outside the current sub method.

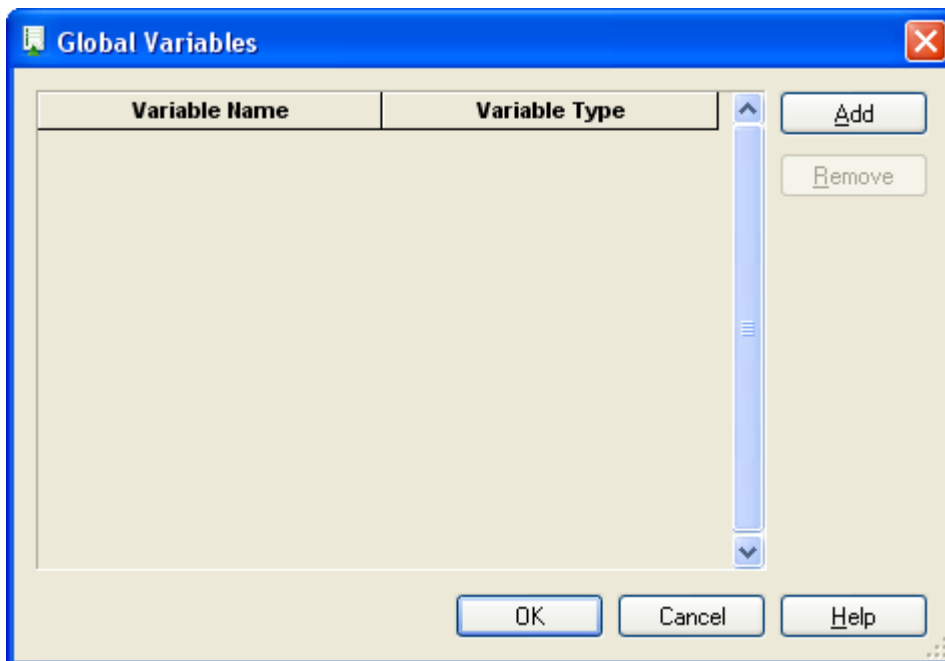
Variables defined in a method are task local variables and have no effect outside the current method and the related sub methods.

Global variables are valid for all tasks of a workflow using the same method.

9.2.2 Specification of Global Variables

It may be useful to specify global variables (or constants) for values that must be valid for all tasks of a workflow using the same method. To add a global variable:

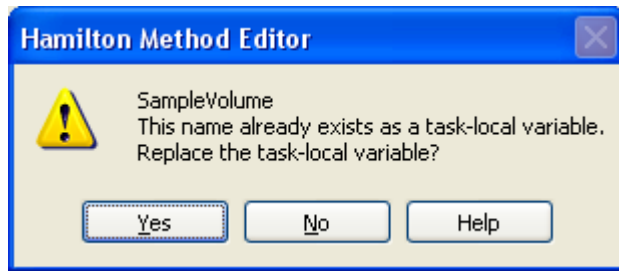
- In the “Method” menu, click “Global Variables”. The “Global Variables” dialog box appears:



- Click “Add”.
- Select the “Variable Type” and specify the “Variable Name”.

Variable Name	Variable Type
SampleVolume	Variable
	Variable
	Sequence
	File/ Communication Port
	Timer
	Event
	Object
	Array of Variables

- Click “OK” when all global variables have been specified. If you have specified a variable name used for local variable of a selected Method, a warning message box appears.



ATTENTION

Avoid name conflicts between global, task local and local variables.



NOTE

It is recommended to use internal naming conventions for the different types and use of variables, e.g.:

- **GV_<variable name>**
for global variables used for all tasks of the same method.
- **TV_<variable name>**
for task local variables used in the whole method.
- **LV_<variable name>**
for local variables used in a sub method.
- **MLV_<variable name>**
for local variables used in a method but not in the related sub methods.

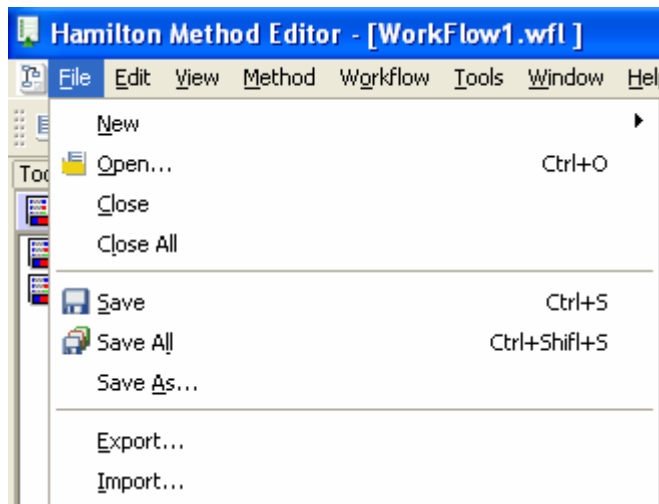
It is not possible to define variables which are valid in a method but not in the related sub methods. You have to define task local variables with different names

10 Export and Import of Workflows

The export function stores the complete workflow with all its components to a package file. Using the import function, the exported workflows can be transferred to another PC, e.g., from the development to the production environment.

The export and import of workflows is very similar to the export and import of methods. Exporting/Importing a workflow will bring all the used methods with the corresponding decklayout, libraries, labware etc into the target environment.

Use the import/export function of the Method editor while having the workflow open to perform the desired action. For more information about import/export, refer to the Microlab STAR Programmers Manual.

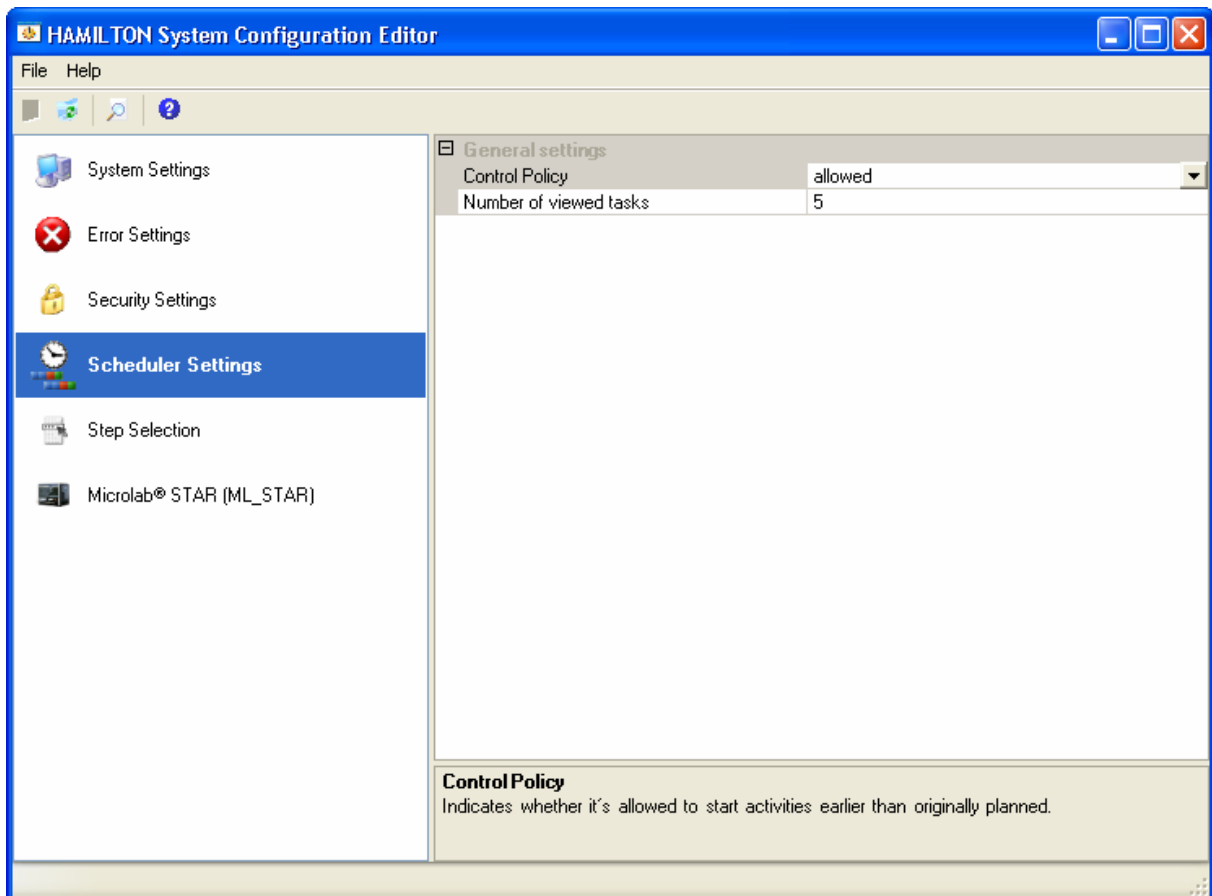


11 Additional Functions for the scheduler

11.1 System Configuration

1. Start the Method editor.
2. In the “Tools” menu, select “System Configuration Editor”. The “HAMILTON System Configuration Editor” window appears.
3. Select “Scheduler settings”.
4. Specify the parameter values or accept the defaults. See Scheduler Settings below.
5. Close the “HAMILTON System Configuration Editor” window. The settings are saved upon request.

11.1.1 Scheduler Settings



- **Control Policy**

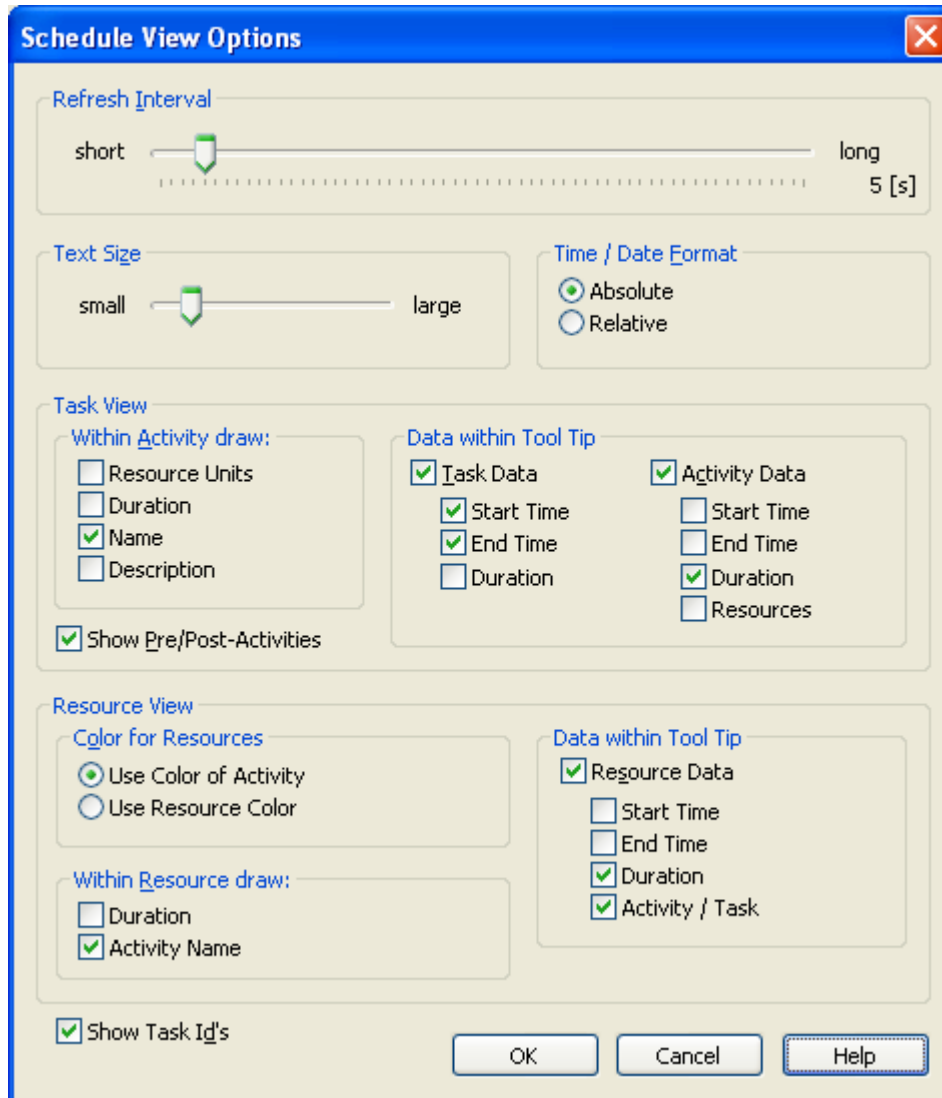
If the value “allowed” is set, the subsequent activity starts immediately after the completion of the predecessor activity to synchronize with other processes. Otherwise, the planned start time is the earliest possible start time.

- **Number of viewed tasks**

The number of maximum processed tasks to preserve in the "Scheduler View" during the next reschedule.

11.1.2 Schedule view options

The run Editor offers several settings for the view at scheduling time. These setting can be changed in the Schedule view → Options menu:

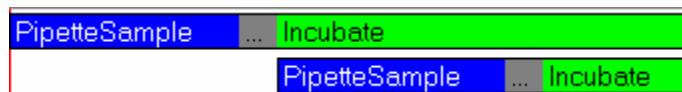


- **Refresh Interval**
This changes the delay between an update of the schedule view. Shorter delays consume more computer resources.
- **Text Size**
This changes the size of all text in the Run window
- **Time / Date Format**
Allows to display either the actual time (Absolute) or starting at 0:00:00 (Relative)

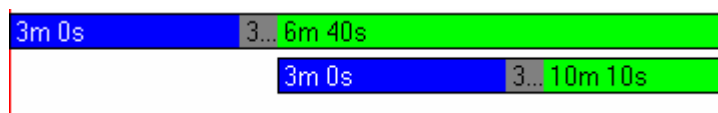
- **Task View – Within Activity draw**

These four settings change the appearance of the Activity bar

Example with setting *Name* activated:



Example with setting *Duration* activated:

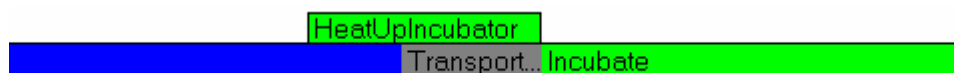


- **Task View – Data within Tool Tip**

The Tool Tip Data will be displayed when the mouse pointer is placed on a Task or activity.

- **Task View – Show Pre/Post-Activities**

Mark/unmark this checkbox will control the visibility of pre- and Postactivities. Example with checkbox marked:



- **Resource view – Color for Resources**

Allows to set the resource color specified in the resource itself or to use the color specified in the activity.

- **Resource View – Data within Tool Tip**

The Tool Tip Data will be displayed when the mouse pointer is placed on a Task or activity. These settings are valid for the resource view.

- **Resource View – Within Resource draw**

In here, the appearance of the resource draw can be controlled. Either Duration or Activity Name can be displayed.

Example with setting *Duration*:



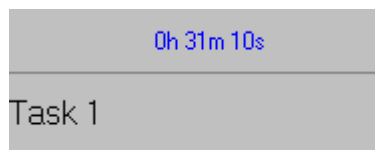
- **Show Task Id's**

Allows to display/hide the task IDs.

Example with activated checkbox:



Example with deactivated checkbox:



11.2 Scheduler Steps

The steps for the Scheduler are available in two different sections: The Scheduler Smart steps and the steps of the Scheduler Library 'HLSchedLib.hsl'.

To have the Scheduler Library available, include the HLSchedLib.hsl in *Method* → *Libraries...*

If the Scheduler Smart Steps are not available in the toolbox, add them as follows:

1. Go to menu "Method" – "Instruments and Smart Steps"
2. Add the "Scheduler Steps" by marking the check box
3. Click "Ok"

11.2.1 Select Method Step

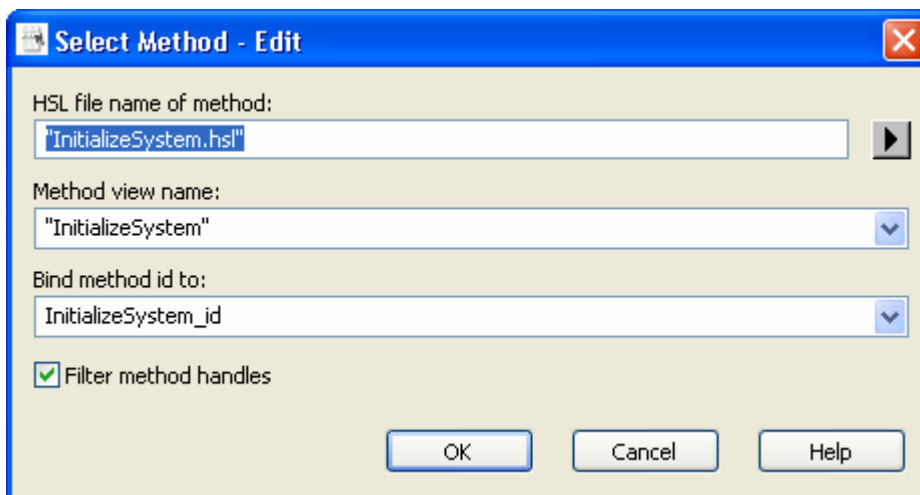


Selects a method that can be activated by the *VENUS Dynamic Scheduler*.




NOTE

A method can be activated with the "Activate Task" step.



- **HSL file name of method**

Defines the file name of the method to be selected. The selected file must have the extension *.hsl*. The value must be a string constant. Use the  button to browse for your target method file.



NOTE

*The path to the file can be relative or absolute.
If it is specified relatively, the file is searched relative to:*

- *the current directory*
 - *the default method directory*
 - *the default library directory*
 - *the directories that are listed in the PATH environment variable.*
- **Method view name**

Defines the view name of the method to be selected. This name appears in the scheduler views. The value can be a variable or a string constant.
 - **Bind method id to**

A unique identifier of the selected method will be returned via this parameter.
 - **Filter method handles**

If this check box is selected, the given list for the parameter “Bind method id to” contains only existing method handles. Otherwise, all variables are shown in this list.
 - **Method parameters**

If a method is selected that uses formal parameters, the parameter grid is shown to define the values to pass (see picture below, where an incubation time of 10 minutes is passed to this method).

Select Method - New

HSL file name of method:
"MainMethod.hsl" ▶

Method view name:
"MainMethod" ▼

Bind method id to:
MainMethod_id ▼

Filter method handles

Method parameters:

Name	Value
IncubationTime [in]	600 ▼

OK Cancel Help

Step errors

If the step fails, a runtime error occurs. Use of the general step “Error Handling by the User” allows handling runtime errors and are helpful for continuing method execution.

11.2.2 Activate Task Step



Schedules a previously selected method to become active at a certain time with certain precedence relations.



NOTE

A method can be selected with the "Select Method" step.

Activate Task - Edit
✕

Method id / task id:

Filter method/task handles

Task view name:

Bind task id to:

Filter task handles

Inherit cancel from predecessor tasks:

Activation criterions

	Relation	Id of <OtherTask>	Period [s]
1	Delayed to <OtherTask>	<After startup>	0

Filter task handles

Activate at

Date

Year:

Month:

Day:

Time

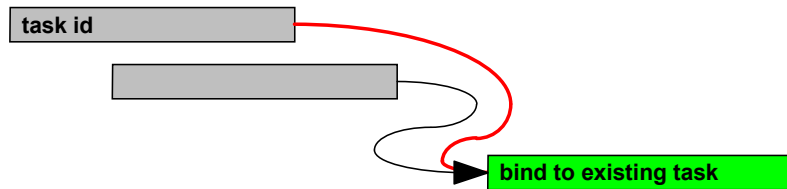
Hour:

Minute:

Second:

- **Method id / task id**

Defines either the id of a registered method to be activated, OR the id of an already activated method (task). The second case is especially useful to introduce an additional precedence relation between two already activated methods (tasks).



The value can be a selection given in the list.

- **Filter method/task handles**

If this check box is selected, the given list for the parameter “Method id/task id” contains only existing method and task handles. Otherwise, all variables are shown in this list.

- **Task view name**

Defines the view name of the task to be activated. This name appears in the views. The value can be a variable or a string constant.

- **Bind task id to**

A unique identifier of the activated method (task) will be returned via this parameter. The value can be a variable.

- **Filter task handles**

If this check box is selected, the given list for the parameter “Bind task id to” contains only existing task handles. Otherwise, all variables are shown in this list.

- **Inherit cancel from predecessor tasks**

Indicates whether the new task inherits the cancel from the predecessor task. If the value of this parameter is not zero, the new task will be cancelled if the predecessor task is cancelled. The value can be a variable or a selection out of the given list.

- **Activation criterions**

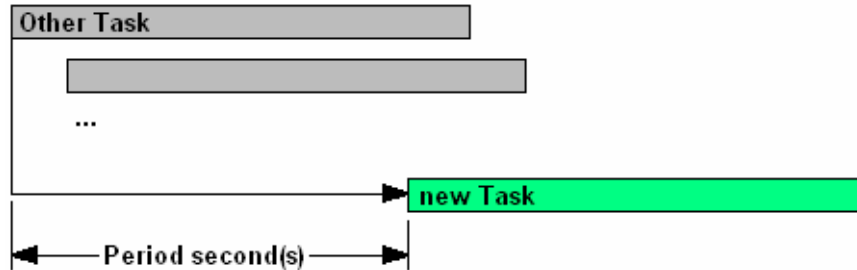
This table is used to define an arbitrary number of precedence relations for the new task.

- **Relation**

Three different types of criteria can be defined:

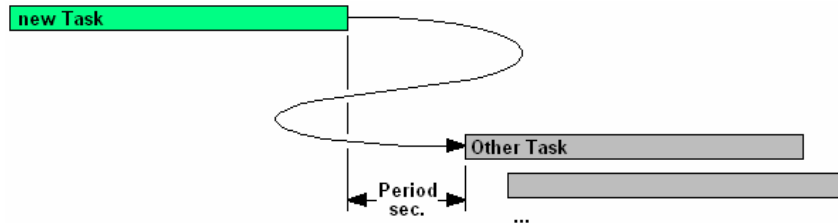
Delayed to <Other Task>

The new task will be activated at the earliest opportunity, with a delay of “Period second(s)” after the other Task has been activated. This relation can also be defined to the start time of scheduling by choosing the <After startup> option for “Other Task”.



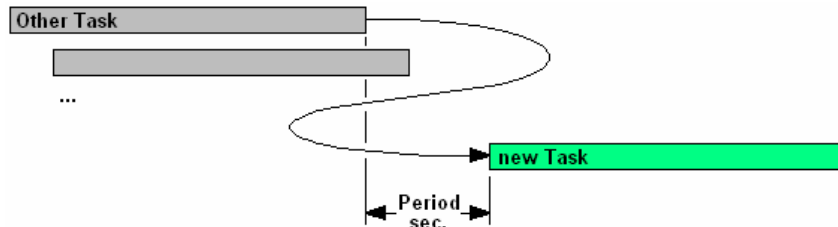
Before <Other Task>

The other task will be activated at the earliest opportunity, with a delay of “Period second(s)” after the new task has finished.



After <Other Task>

The new task will be activated at the earliest opportunity, with a delay of “Period second(s)” after the “other task” has finished.



- **Id of <Other Task>**

The Id of an already activated method (task) to which the relation is to be made. For the "Delay Task" relation, this can also be <After startup>. The value can be a variable or a selection out of the given list.

- **Period [s]**

Earliest start time of the new task relative to another method (task) in seconds.

The value can be an integer or a variable.

- **Filter task handles**

If this check box is selected, the given lists in the grid for the parameter Id of <Other Task> contain only existing task handles. Otherwise, all variables are shown in this list.

- **Activate at**

Used to define the earliest start of the method by specifying an absolute activation time.

Year, Month, Day

Defines the absolute earliest start date of the method to be activated. The date can be selected by using the "Calendar control". Each field can be an integer or a variable.

Hour, Minute, Second

Defines the absolute earliest start time of the method to be activated. The time can be selected by using the "Time set function". Each field can be an integer or a variable.

Step errors

If the step fails, a runtime error occurs. Using of the General Step "Error Handling by the User" enables to handle such runtime errors and to continue method execution.

11.2.3 Cancel Task Step

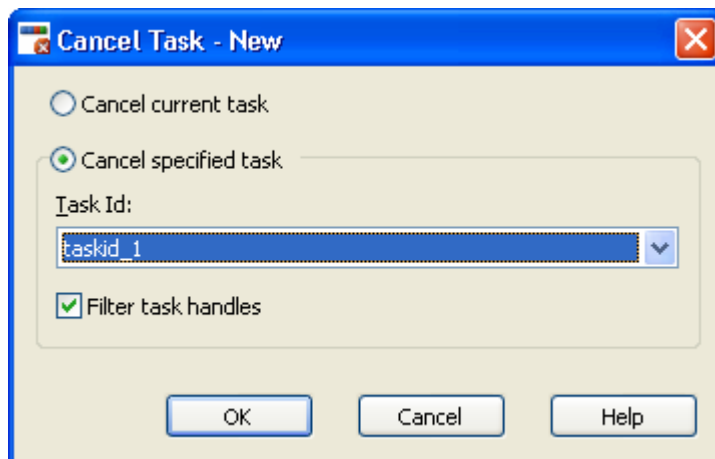


Cancels the processing of the given task. All planned resources of a cancelled task are released.



NOTES

- *The task will be cancelled before it executes the next statement. That means, outstanding blocking calls (e.g. instrument low-level steps) cannot be cancelled.*
- *A cancelled task cannot be resumed.*
- *If a cancel handler has been specified for the activity block that is currently being executed, program control of a cancelled task will be transferred directly to the Activity Block cancel handler. This is especially useful for cleanup operations.*
- *If the optional cancel handler is omitted, the program control of a cancelled task will be transferred to an activated error handler by raising an error.*



- **Cancel current task**
Cancels the currently processing task.
- **Cancel specified task**
Cancels the task with the specified id.
- **Task Id**
This field defines the id of the task to be cancelled. The value can be a selection out of the given list.
- **Filter task handles**
If this check box is selected, the given list for the parameter “Task id” contains only existing task handles. Otherwise, all variables are shown in this list.

Step errors

If the step fails, a runtime error occurs. Using of the General Step “Error Handling by the User” enables to handle such runtime errors and to continue method execution.

11.2.4 Schedule Block Step



Activates the *VENUS Dynamic Scheduler* at runtime and schedules the statements of this block in the current context.



NOTE

This is especially useful to activate the VENUS Dynamic Scheduler in an automated environment during runtime at any time to reschedule newly requested workflows.

11.2.5 Reschedule Step



Activates the *VENUS Dynamic Scheduler* at runtime and reschedules all currently schedulable tasks in the workflow based on the current resource availability.

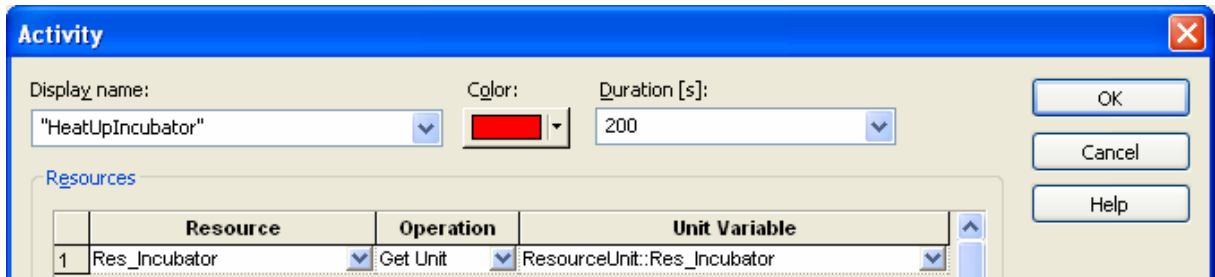


NOTE

This is especially useful to activate the VENUS Dynamic Scheduler in an automated environment at runtime to reschedule a running workflow after resource breakdown in recovery procedures.

11.3 Resource distribution

There are several options to control the distribution of resources. The most common is to make use of the Operation field in the Activity dialog:

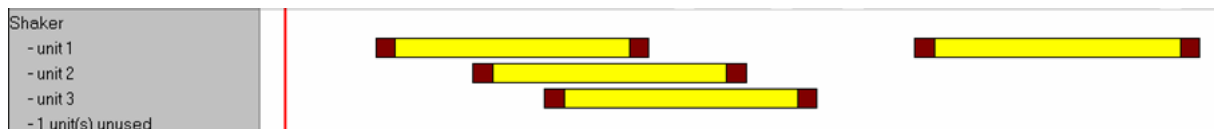


Using Get Unit will automatically calculate the empty unit positions on a resource. Depending on the resource distribution settings (available in the Scheduler steps), the resource units will be used in different ways:

Distribution	Effect
0	The Scheduler chooses the first unit that fits best. In this case, some units may be unused.
1	From the currently used unit, the Scheduler chooses the next unit that fits best.
2	The Scheduler chooses a free unit in random mode.

To use the Scheduler for hardware calculations, the Distribution mode 0 shall be selected (it is default).

E.g. having a shaker with 4 modules on it, the plan will show that only 3 modules are used (because shaker unit 1 is free when plate 4 arrives). Therefore, the customer does not have to buy 4 shakers.



11.4 Programming additional tasks at runtime (example)

Tasks can be scheduled while the previously scheduled tasks are running.

1. Add a “Schedule Block” step ❶ to the method and the included “Activate Task” step ❷.
2. Mark the schedule block and the corresponding method selection ❸.

Click “Executor only” on the context menu ❹ (see [Scheduler Step Modes](#)).

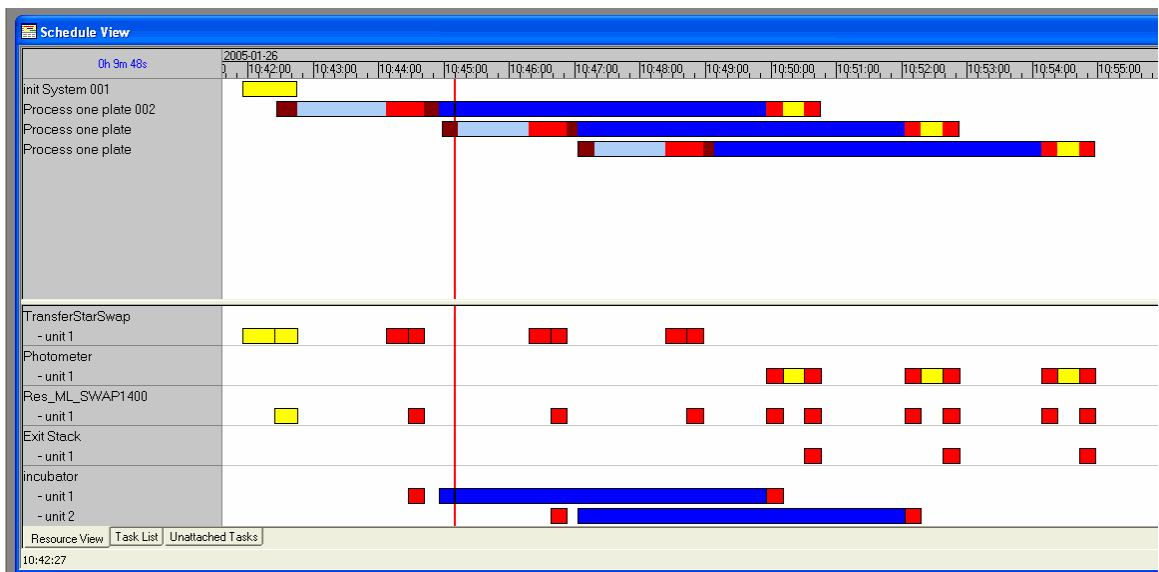
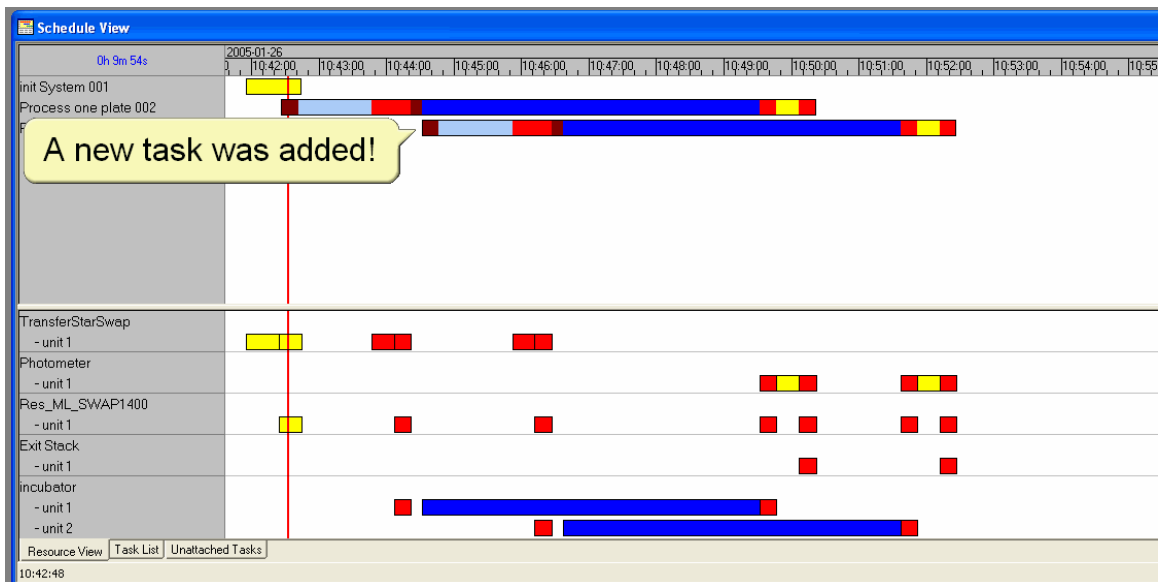
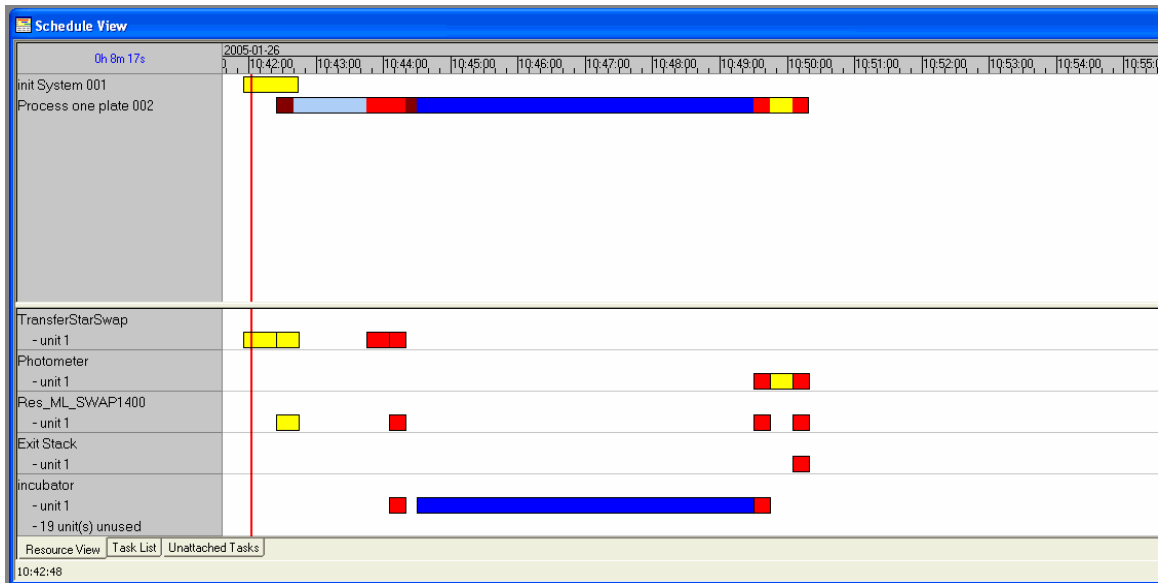
The screenshot displays the VENUS Dynamic Scheduler interface. At the top, a menu bar includes 'Method', 'MovePlateToIncubator', 'PipetteOnStar', 'Sleep', 'TransferFromIncubatorToPhotometer', 'TransferFromPhotometerToExit', and 'OnAbort'. Below this is a table of method steps:

Method	MovePlateToIncubator	PipetteOnStar	Sleep	TransferFromIncubatorToPhotometer	TransferFromPhotometerToExit	OnAbort
1	Schedule Block from Scheduler Steps Begin of Schedule Block					
2	Select Method from Scheduler Steps Method file name: "ProcessOnePlateInfini.hsl" Method view name: "Process one plate" Method id: 'ProcessOnePlateInfini_id'					
3	Activate Task from Scheduler Steps Method id / task id: 'ProcessOnePlateInfini_id' Task view name: "Process one plate" Bind task id to: 'taskid_1' Activate: - '0' second(s) after scheduler startup					
4	Schedule Block from Scheduler Steps End of Schedule Block					
5	Assignment X=0 'speedFactor' = '3'					
6	Begin Activity Display Name: "Load carriers", Duration in seconds: 15, Reso					
7	Advanced Load Settings from Microlab® STAR Smart Instrument short name 'ML_STAR', load settings for '3' - 'ML_STAR.Plate'; Error recovery for barcode errors - 'ML_STAR.Samples'; Error recovery for barcode errors - 'ML_STAR.Reagent'; Error recovery for barcode errors					
8	Load from Microlab® STAR Smart Steps Instrument short name 'ML_STAR', load '2' sequence - 'ML_STAR.Plate' - 'ML_STAR.Samples'					
9	Sleep of ProcessOnePlate Sleep(12)					
10	End Activity					
11	Begin Activity Display Name: "Pipette reagent and samples", Duration in sec					
14	End Activity Begin Cancel Block					

A context menu is open over step 3, showing the following options:

- Cut
- Copy
- Paste
- Delete (marked with ❹)
- Del
- Clear Selection
- Edit Step... (Ctrl+E)
- Bind Return values... (Ctrl+B)
- Disable Steps (Ctrl+D)
- Enable Steps (Ctrl+Shift+D)
- Group Selection...
- Insert HSL Code...
- Activity...
- Executor only
- Scheduler only
- Executor and Scheduler
- Toggle Outlining Expansion (Ctrl+M)

Result: Tasks are added during runtime



11.5 Scheduler Step Modes

Steps outside of activities can be selected to be executed at schedule time only, at execution time only, or at both. To select the desired mode:

1. Select the steps to be set to a specific mode
2. Click the right mouse button
3. Select the desired mode "Scheduler only", "Executor only", or "Executor and Scheduler" (default)

The background color will change depending on the selected mode: white (default) is for scheduler and executor, light yellow for scheduler only, and light green for executor only. Activities have the same color as executor only because they will be executed at runtime only.

An example for a 'Scheduler only' step could be a variable incubation time that has to be entered by the user at scheduling time. This time will then be passed to the executor and has not to be entered at runtime again.

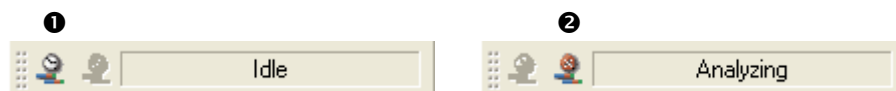
11.6 Additional Tools for the VENUS Dynamic Scheduler

11.6.1 Workflow Toolbar



- | | |
|------------------------|--|
| ❶ Activation Link | Used to connect the tasks for defining the workflow. Refer to section Defining Task Dependencies . |
| ❷ Select or move items | Standard Windows selection tool. |
| ❸ Tools for alignment | Aligns the selected workflow items. |
| ❹ Tools for spacing | Equally spaces currently selected items in the workflow. |
| ❺ Tools for movement | Moves the selected items step by step. |
| ❻ Tools for ordering | Orders elements (front, back) |

11.6.2 Scheduler Toolbar



- | | |
|---------------------|---|
| ❶ Schedule workflow | Starts the calculation of the workflow. |
| ❷ Stop scheduling | Stops the scheduler run. |

11.7 Additional Menu Functions for the VENUS Dynamic Scheduler

11.7.1 File Menu

Export / Import These functions have been expanded. Workflows may be imported and exported.
Refer to Export and Import of Workflows.

11.7.2 Edit Menu

Add Activation Link ***This function is only activated when using the Graphical Workflow Window.***
Used to insert a relationship between two methods.
Refer to section [Defining Task Dependencies](#).

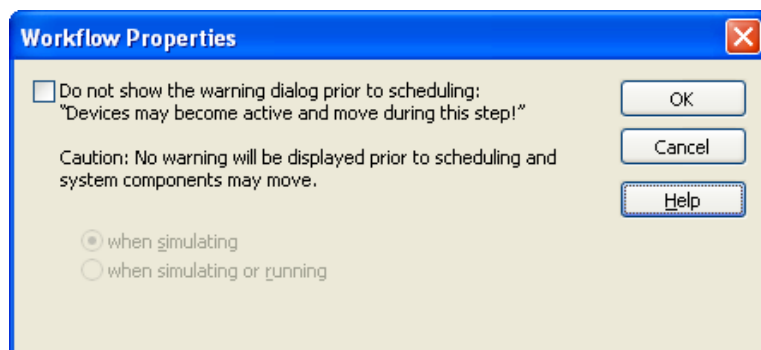
11.7.3 View Menu

The following functions are only activated together with the Graphical Workflow Window. They can also be selected by the context menu (right click in the graphical workflow definition area).

Grid Toggles the visibility of the grid On/Off.
Snap to Grid Toggles the snap function of the grid On/Off.
Page Bounds Toggles the visibility of page bounds

11.7.4 Workflow Menu

Schedule Starts the scheduling of the currently displayed workflow.
Stop Scheduling Stops the currently active scheduler run.
Selected Methods Opens the “Selected Methods” dialog box.
Workflow Properties Opens the “Workflow Properties” dialog box.



Reset Activity Durations The collection of activity durations will be cleared.

12 Glossary

Activity

An activity is the smallest unit for the scheduler. An activity includes a single step, a sequence of steps or a submethod. Activities use the same set of resources for a certain time. The set of method steps of an activity is executed uninterrupted at runtime as one block. Activities are defined within the activity editor or the method editor.

Aspirate

To draw up liquid into a pipetting device.

Barcode Reader

Device for reading sample/plate Barcodes. Part of the Autoload Option.

Carrier

Unit for loading plates, tubes and tips on the MICROLAB[®] STAR Line deck. Can be used manually or if possible by the Autoload option.

Container

A container defines a tube, vessel or a single well of a plate.

System Deck

A collection of Labware placed upon a Deck.

Dispense

To distribute quantities of liquid from a pipetting device.

HSL

Hamilton Standard Language

Instrument

Hardware of e.g. the MICROLAB[®] STAR Line (mechanics, electronics, and firmware)

Instrument steps

The commands made available by the firmware for controlling the instruments.

Labware

Refers to movable items to be placed on instruments 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)

Positive pipetting of liquid which may be achieved either by pressure, or capacitive signal detection and transfer.

Loading, unloading

The process by which plate, tube and tip carriers are brought on and off the MICROLAB® STAR Line deck. This can happen automatically by means of the Autoload Option, or manually.

Method

A method is a sequence of activities. Once a method is scheduled or running, it is called a task. If you have a method to process one plate, and you schedule five methods to process five plates, the plan will show five tasks.

MICROLAB® STAR VENUS Software

Software to run the MICROLAB® STAR Line.

Pause

Interruption of processing. The current processing steps are completed.

Pipetting

Transfer of liquids, usually a defined volume, from one container to another.

Pipetting arm

Assembly equipped with the pipetting tool and/or plate handler, as well as the common X-drive.

Pipetting channel

Hardware including the function of picking up a tip, aspirating, dispensing, tip eject, liquid level detection and the Y/Z- movements.

Pipetting module

Firmware (-processor-program) which controls a pipetting channel, in which category are included the Y and Z pipetting movement, and the LLD.

Plan

A plan is a scheduled workflow. After scheduling, the plan shows the start and end times of each activity. These are arranged on the availability of the resources and on the dependencies that you set.

Rack

Grouping of containers, as DWP, MTP, etc.

Random access

Means that every channel can access any spot anywhere on the work area.

Resource

A resource is an instrument, a part of an instrument, e.g. a pipetting arm, a transfer position, a device like a reader, or a piece of labware. Resources perform activities. Each resource has a capacity.

Run

Execution of the processing steps defined in the relevant 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® STAR Line according to the processing plan. The run can be interrupted to load more elements. Then processing goes on according to a newly calculated processing plan, with the run being started again. Loading is not a part of the run.

Run abort

Cancelled run by the user or by the MICROLAB® STAR.

Run Visualization

Visualization of the current run, reporting the status of the MICROLAB® STAR Line.

Sample

Refers to a liquid in an unambiguously identified container which is to be processed.

Stacker

Storage unit for racks

Tip

Disposable tip for pipetting

Trace

Record of the status during processing.

User

User of the software. Access rights for different type of user can be defined, as operators, laboratory manager, etc.

Work area

The area on the MICROLAB[®] STAR Line to which access is provided during processing. Elements to be pipetted or handled can be placed in this area.

Workflow

A workflow is a number of chained tasks (methods) and their dependencies. It defines the order of the tasks on runtime. It can be defined using the graphical workflow section and/or using the step by step editor (the advanced window) for individualized workflows. This is helpful if you want to mix different methods within the same plan.

Worklist

Information sent from outside the system, as to what method(s) is (are) to be executed on the MICROLAB[®] STAR, and with what liquid.