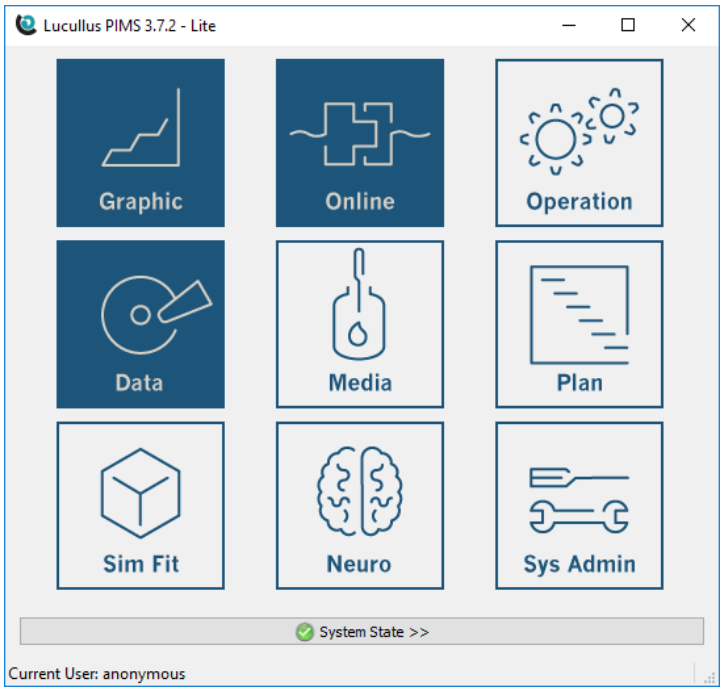
Lucullus PIMS Lite
Windows Version



OPERATOR MANUAL

Software Version 3.7.4
Document Version 1.0

applikon® BIOTECHNOLOGY

TABLE OF CONTENTS

1 GENERAL

1.1 INTRODUCTION

The Lucullus software suite is a PIMS (Process Information Management system) developed by SecureCell in association with Applikon Biotechnology. It is a rich toolbox designed to support planning, preparation, execution, and evaluation of experiments in bioprocess development and production environments.

In its core functionality, the **full** version of Lucullus manages *control* and *data acquisition* in a heterogeneous installation from a single central database. In its core functionality, Lucullus manages *control* and *data acquisition* in a heterogeneous installation from a single central database. A set of **Tools** have been built around this core functionality:

- The Graphic Tool
- The Online Tool
- The Operation Tool
- The Data Tool
- The Media Tool
- The Planning Tool
- The SysAdmin tool

Users with the most privileged license have access to all these tools. In addition to licensing privileges, SecureCell has developed a **simplified** version of Lucullus (Lucullus Lite) consisting of the following components:

- The Graphic Tool
- The Online Tool
- The Data Tool
- A simplified Operation Tool (running as a separate *Configuration Editor*)

The installation and configuration of Lucullus Windows Lite is described in the separate installation manual.

Most of the screenshots will be based on a Windows 10 system. The administrative procedures under Windows 10 differ in small details from those under Windows 7. If you have a Windows 7 system, this manual should still be applicable since most of the procedures are very similar.

For a proper understanding of the application, it helps to consult the Basic Lucullus Concepts in the Appendix (BASIC CONCEPTS).

Note
A common error during startup yields the following message:

application startup failed: unable to connect to database

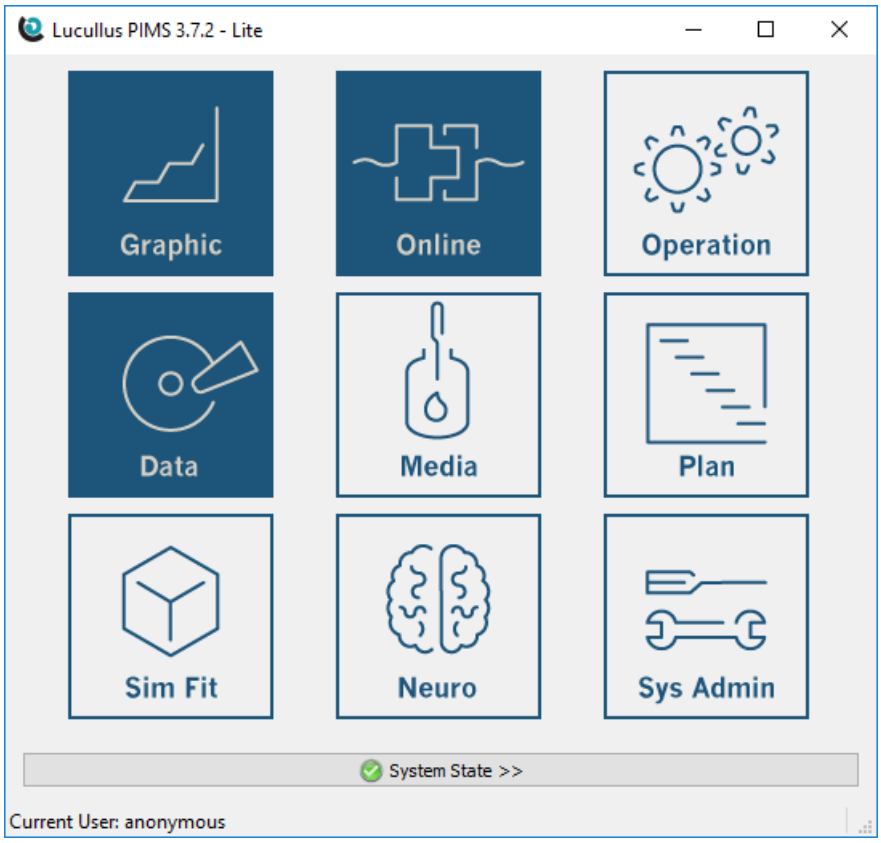
The reason for this is fairly simple: the user is double-clicking on the LPIMS icon too quickly after switching on the system. The user should give the system some time (15-30 seconds) to establish a connection with the database.

2 STARTING AN EXPERIMENT

As explained in the installation manual, one runs the **Configuration Editor** after installation in order to set up a number of reactors and controllers. Any number of reactors and accompanying controllers, up to a total of 8, can be set up. To quickly become familiar with the system, this chapter starts an experiment with a single reactor, the one called **R01**. Maybe more than 1 reactor (and accompanying controller) have been set up, but to make things easy, the first experiment uses just 1 reactor.

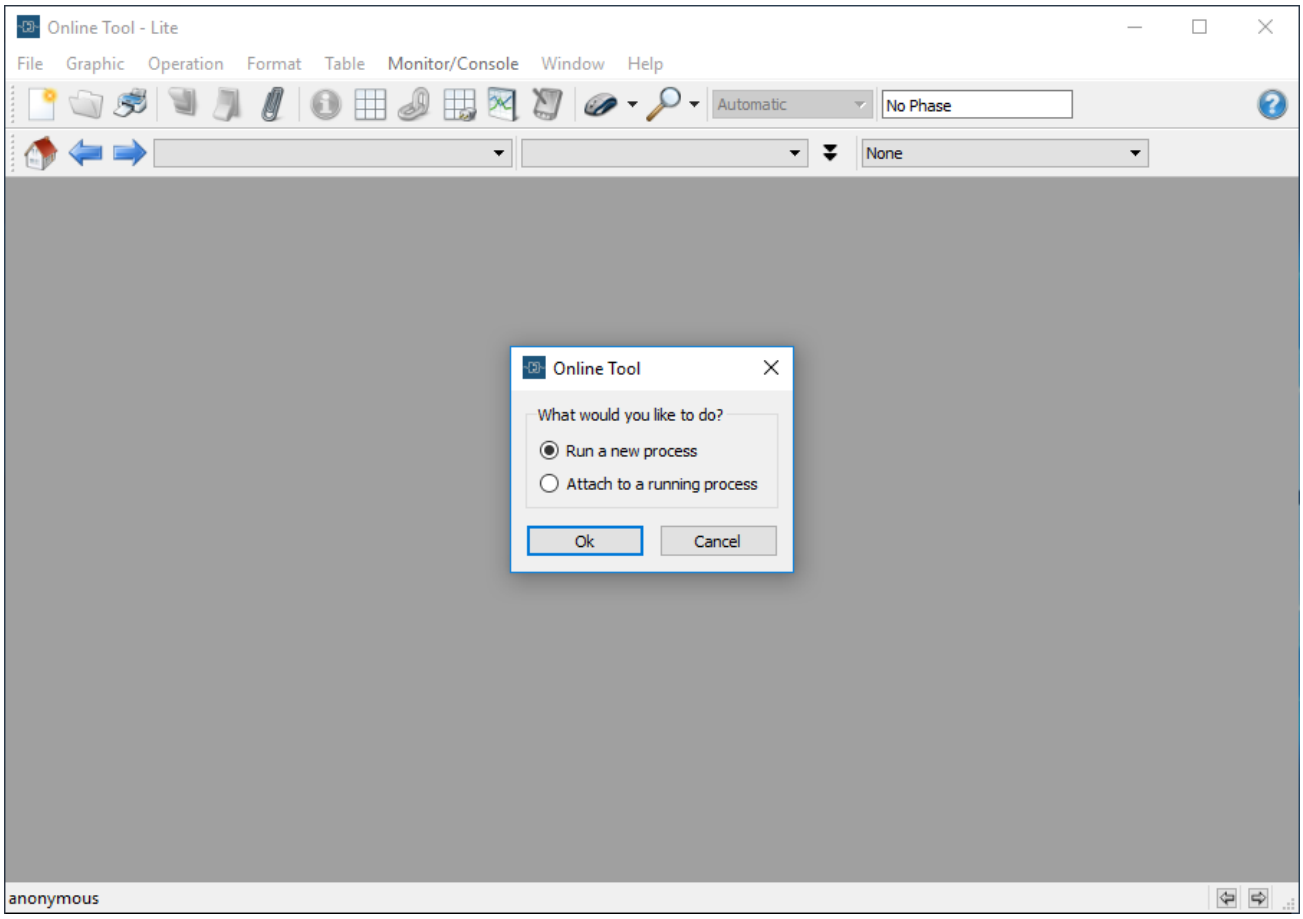
The setup therefore consists of: reactor R01 with a medium, an Applikon controller (*my*-Control, *in*-Control, or *ez*-Control) and a temperature and pH probe.

Start Lucullus via the Lucullus PIMS icon on the desktop, or otherwise via the Windows **Start** button:



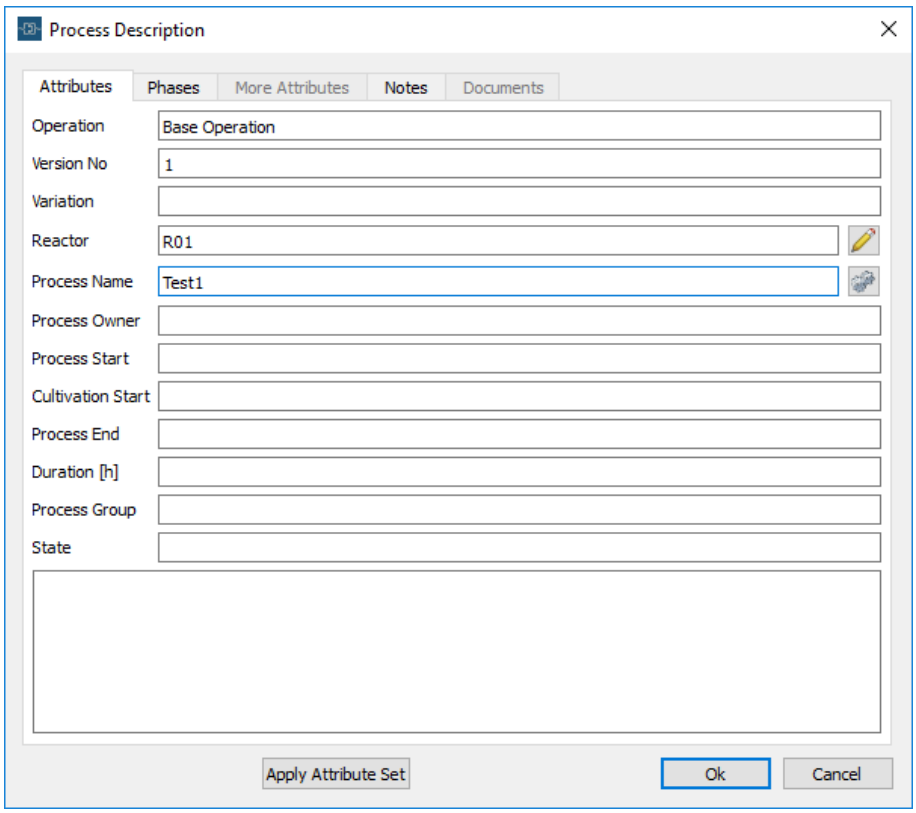
Lucullus for Windows Lite provides 3 tools for viewing processes: The **GRAPHIC**, **ONLINE,** and **DATA** tools. The **Online Tool** is used to start a data viewing & logging session. The **Graphic** and **Data Tools** allow one to view the incoming data graphically or in tabular form.

Click on the **Online** tool to start a data measurement (and viewing) process. This launches the (initially blank) ONLINE Tool:



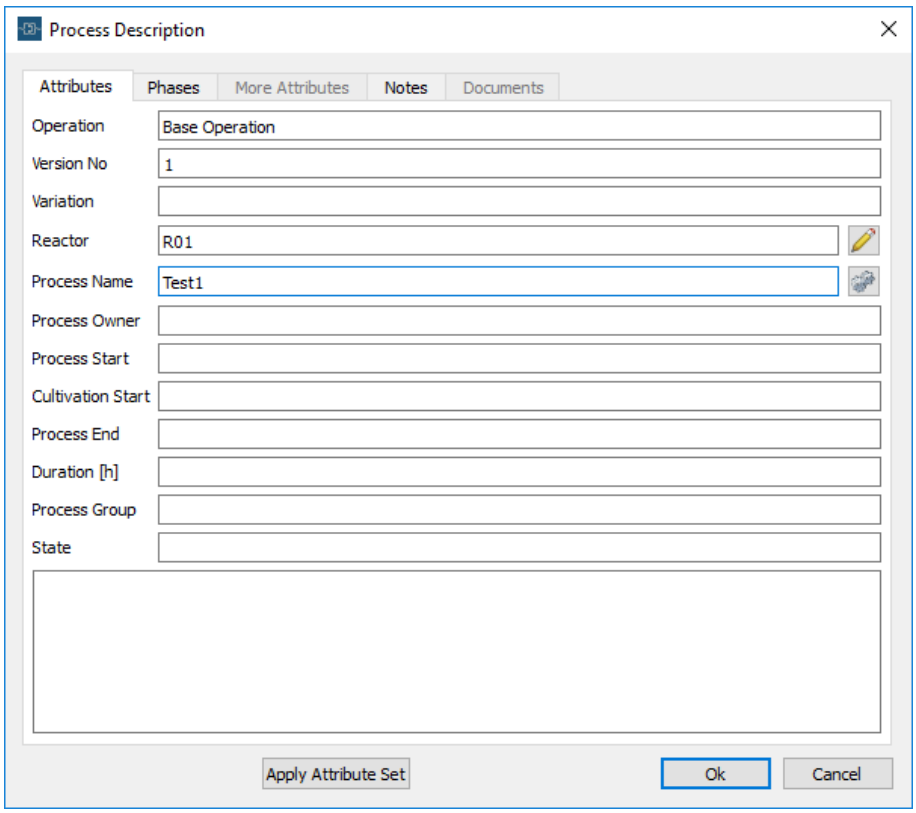
Select **Run a new process** and click on **OK**.

The following screen is shown:

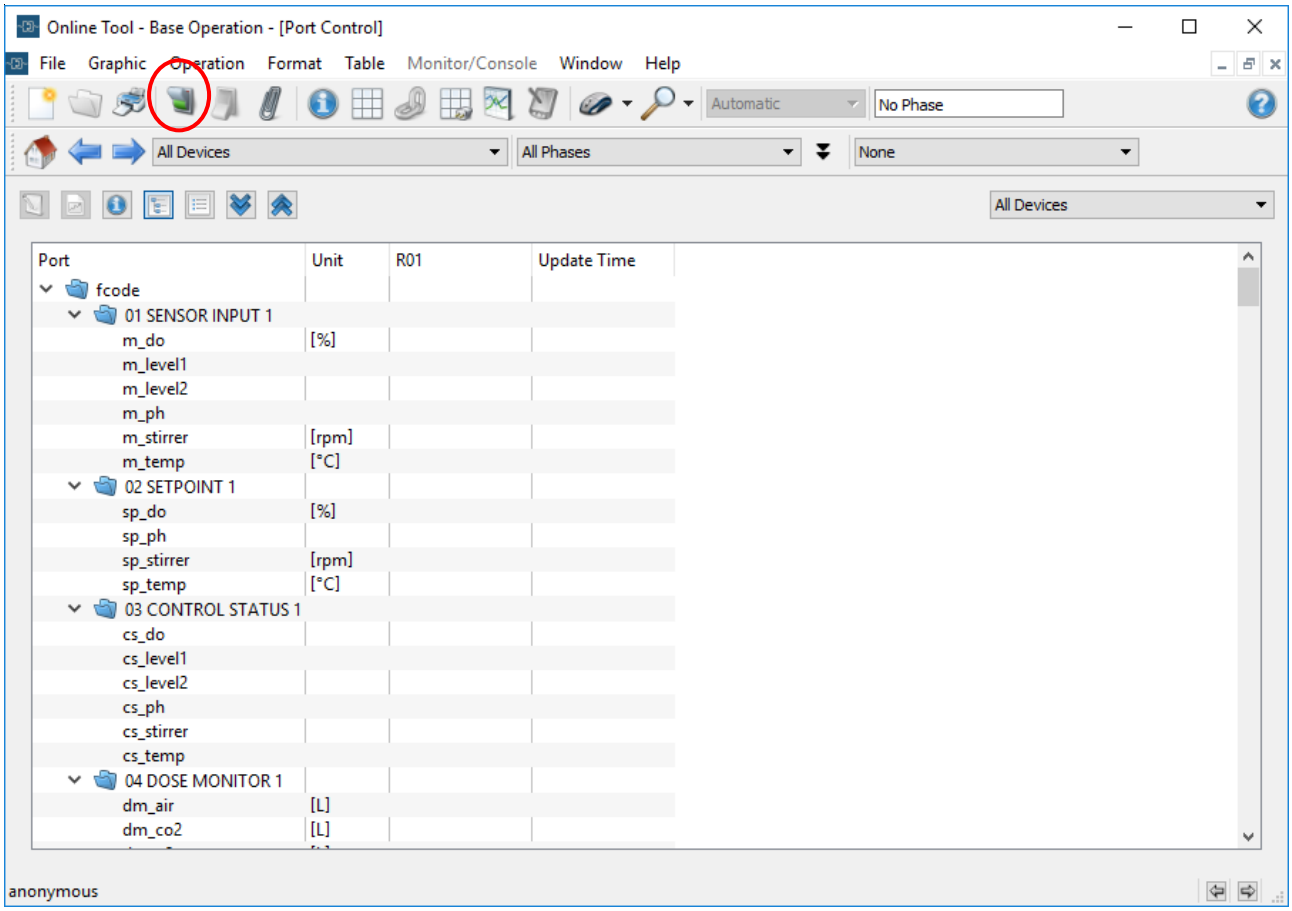


Fill in the **Process Name** field to identify your process.

Click on **OK**.



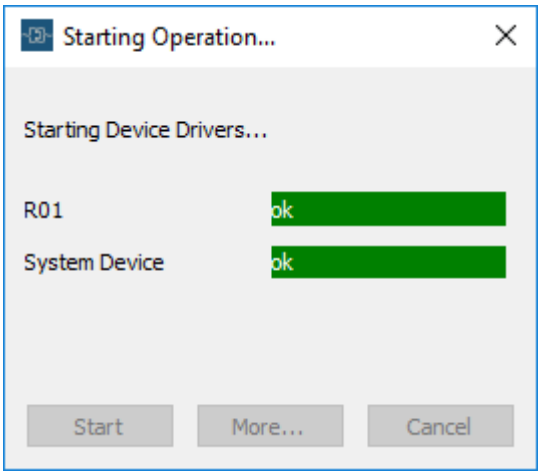
This automatically displays the **port table** for the selected reactor:



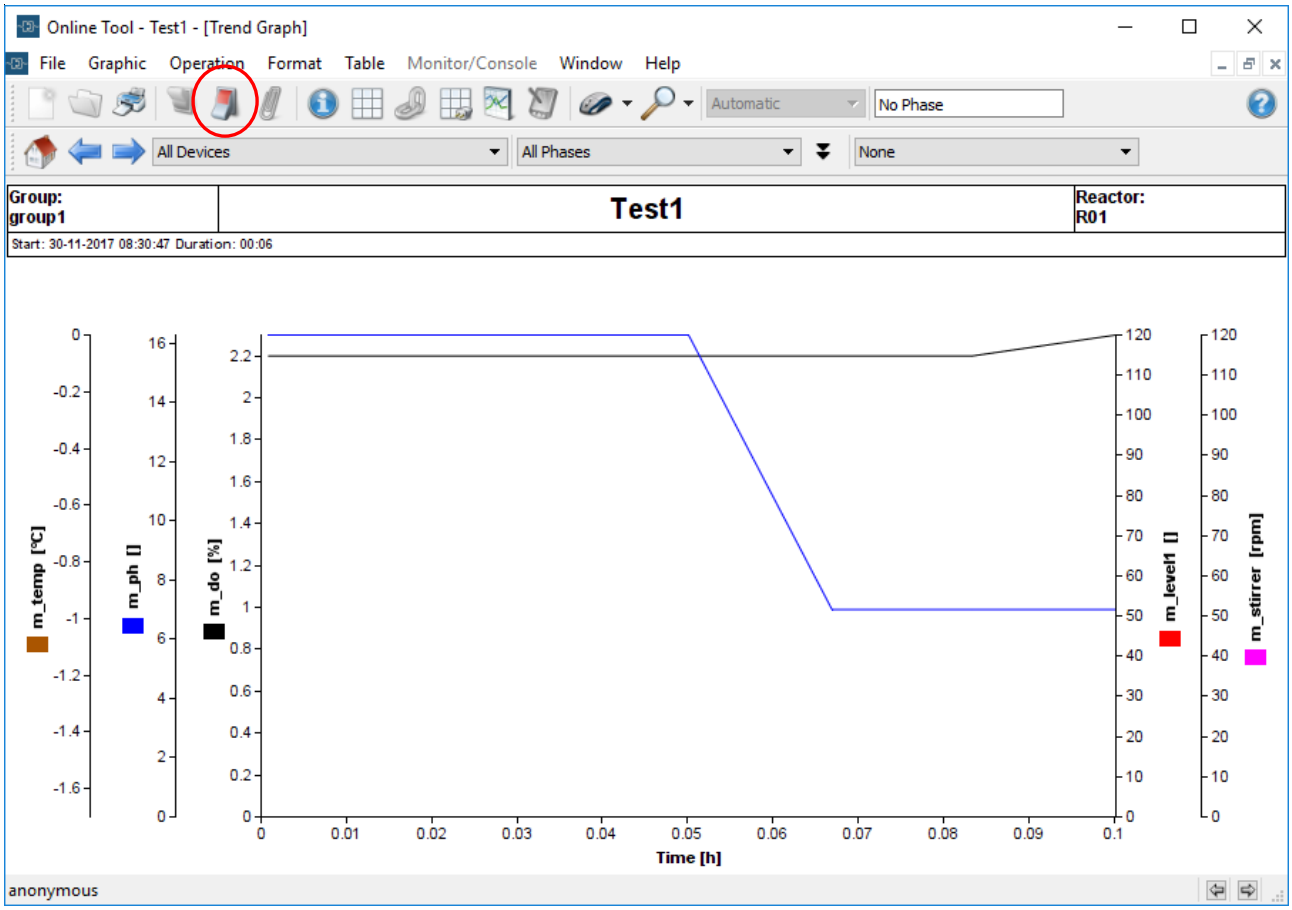
To start collecting and viewing data, click on the process **Start** button (shown circled).

After pressing on Start, the reports the progress of process startup using status bars as shown at right.

The green bar indicates that communication between the device driver and the bioreactor has been successfully established.



After the device drivers have started, the **Trend Graph** is automatically invoked:



It will display the trends for a default set of ports. Initially the default is to record data upon changes only, so it is possible that this display will remain blank for a long while. You can restart the Configuration Editor and set the port of interest (for instance, **m_ph**) to *active logging* every so many seconds (instead of logging on *changes only*).

With the STOP button (circled) you can stop the process. The data is permanently saved, under the same name as the name of the process.

3 THE MAIN FUNCTIONS

This section will present the main Lucullus Windows Lite functions that an operator can execute in addition to starting an experiment. The main additional functions are:

1. Attach to an existing process
2. Load historical data
3. Make calculations on the data
4. Smooth the data
5. Do a regression analysis on the data
6. Make graph overlays
7. Export data
8. Save graphs

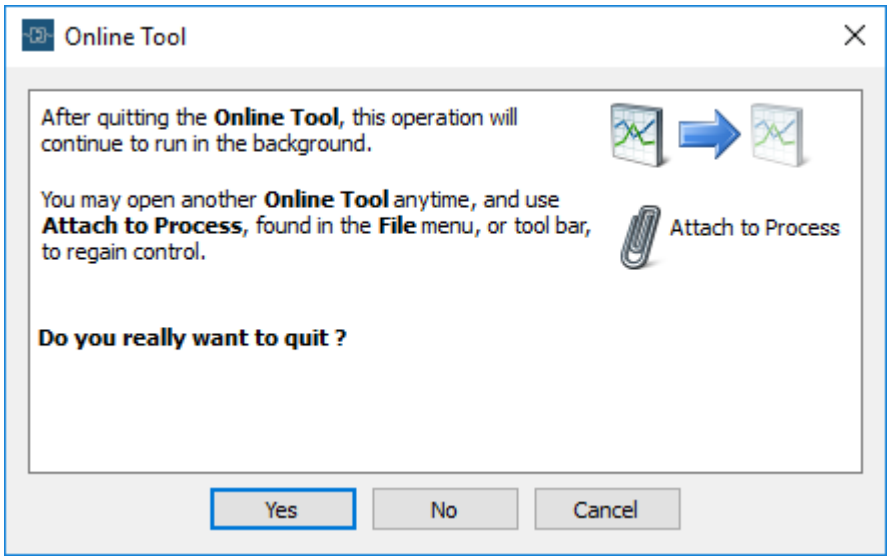
A basic aspect of working in Lucullus involves dealing with *attributes* and there are various kinds of attributes in Lucullus. Generally there are **physical attributes** such as those relating to layout (color, line, line width of the axes of a graph) or those relating to behavior (state of logging, logging mode, logging interval, and dead band of a port). Then there are **informational attributes** such as process attributes (process name, duration, start time, organism, etc.) which can be used to organize processes, and to filter or search for them when selecting from the process list.

When having to select stored data sets, Lucullus will automatically display the relevant screens within which the user can select the attributes to search on. The addition of attributes is an advanced capability available in the Full version of Lucullus.

3.1 ATTACH TO AN EXISTING PROCESS

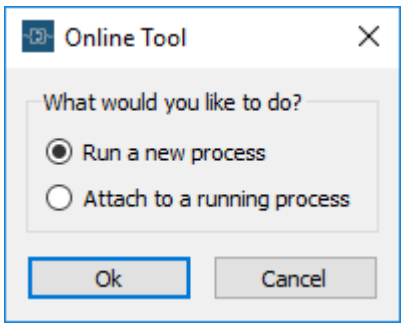
In this situation, you have started a process by following the steps described in the chapter RUNNING A SIMPLE EXPERIMENT. After following all the steps, you will have a running process collecting the desired data.

Now suppose that instead of clicking on the STOP button, you simply close Lucullus. You will get the following screen:

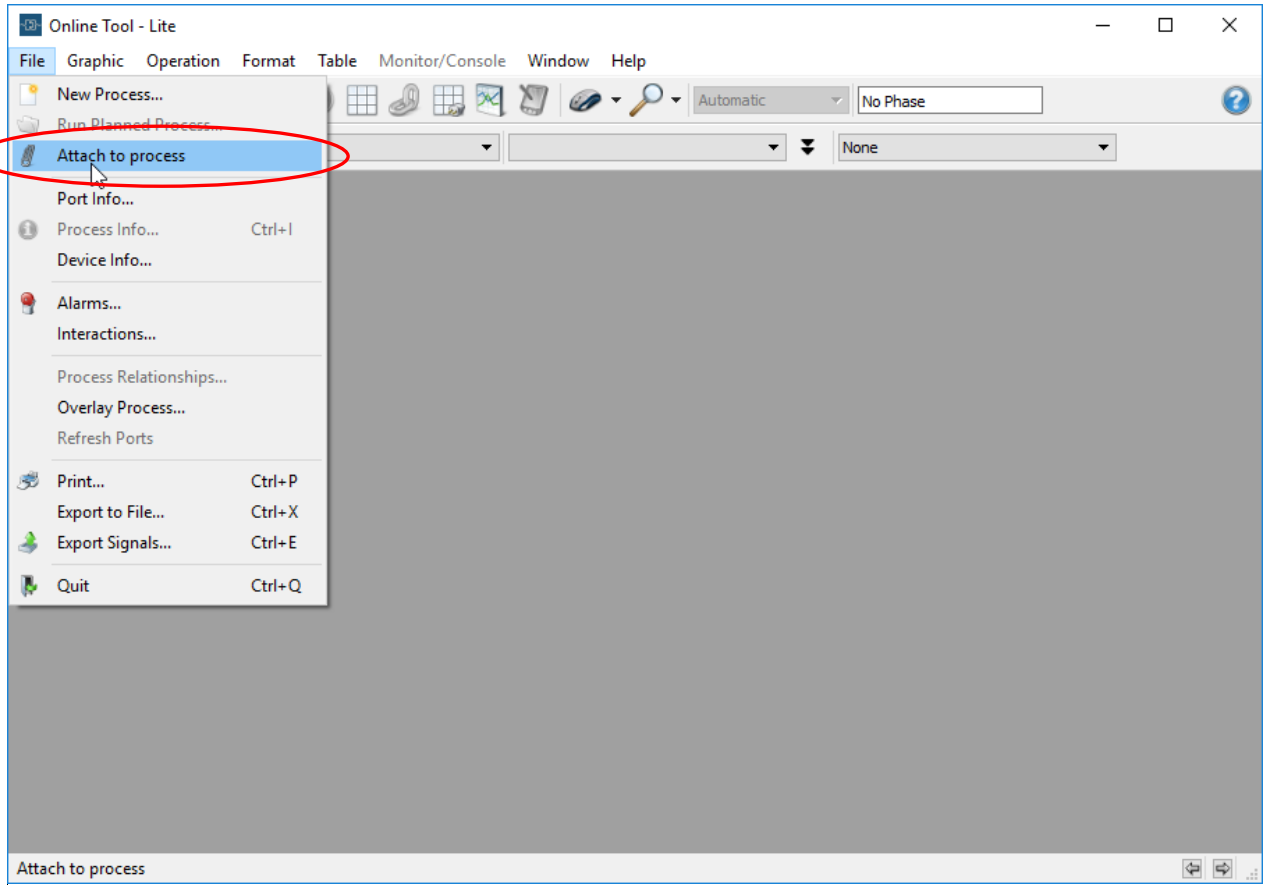


Click on **Yes** and the Online Tool closes but the process will continue to run, as it should. This is the way to prevent unauthorized access to Lucullus, for instance when you leave the lab for the evening, while the experiment continues to run.

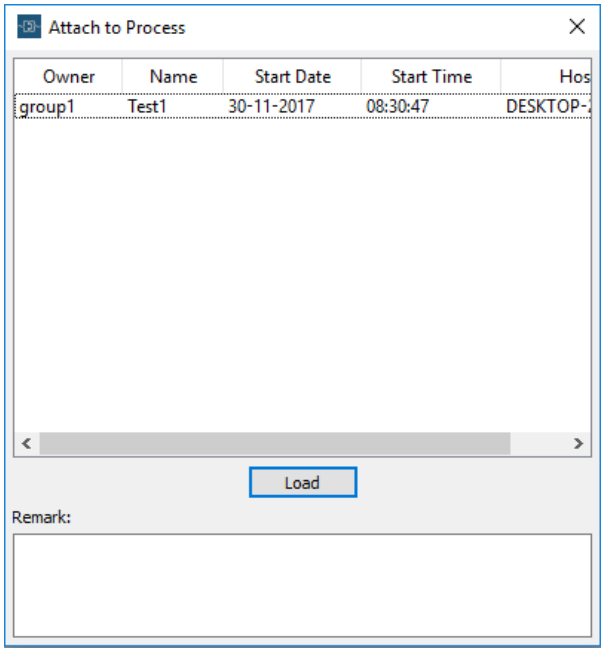
If you start Lucullus Windows Lite again, the process will still be running but will no longer be visible in the PIMS interface. You need to *reattach* the process, which can be done in two ways. The first way is to select *Attach to a running process* in the popup that is automatically displayed when you start the Online Tool again:



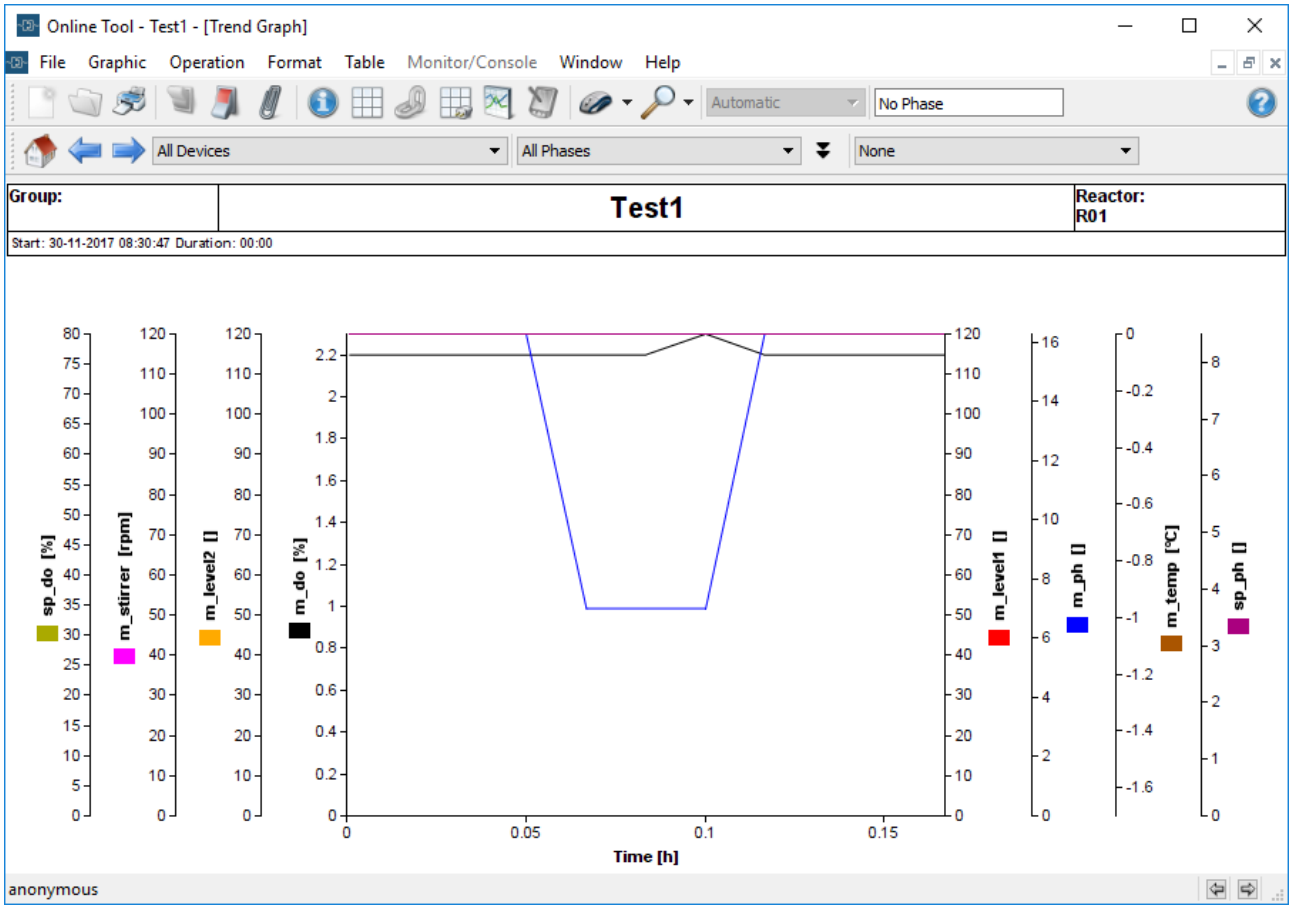
But if you here close the popup via 'X' or clicked **Cancel**, you can still reinvoke it via the **File** menu item:



Click on **Attach to process** as shown above (circled) and you will now get a screen showing all running background processes:



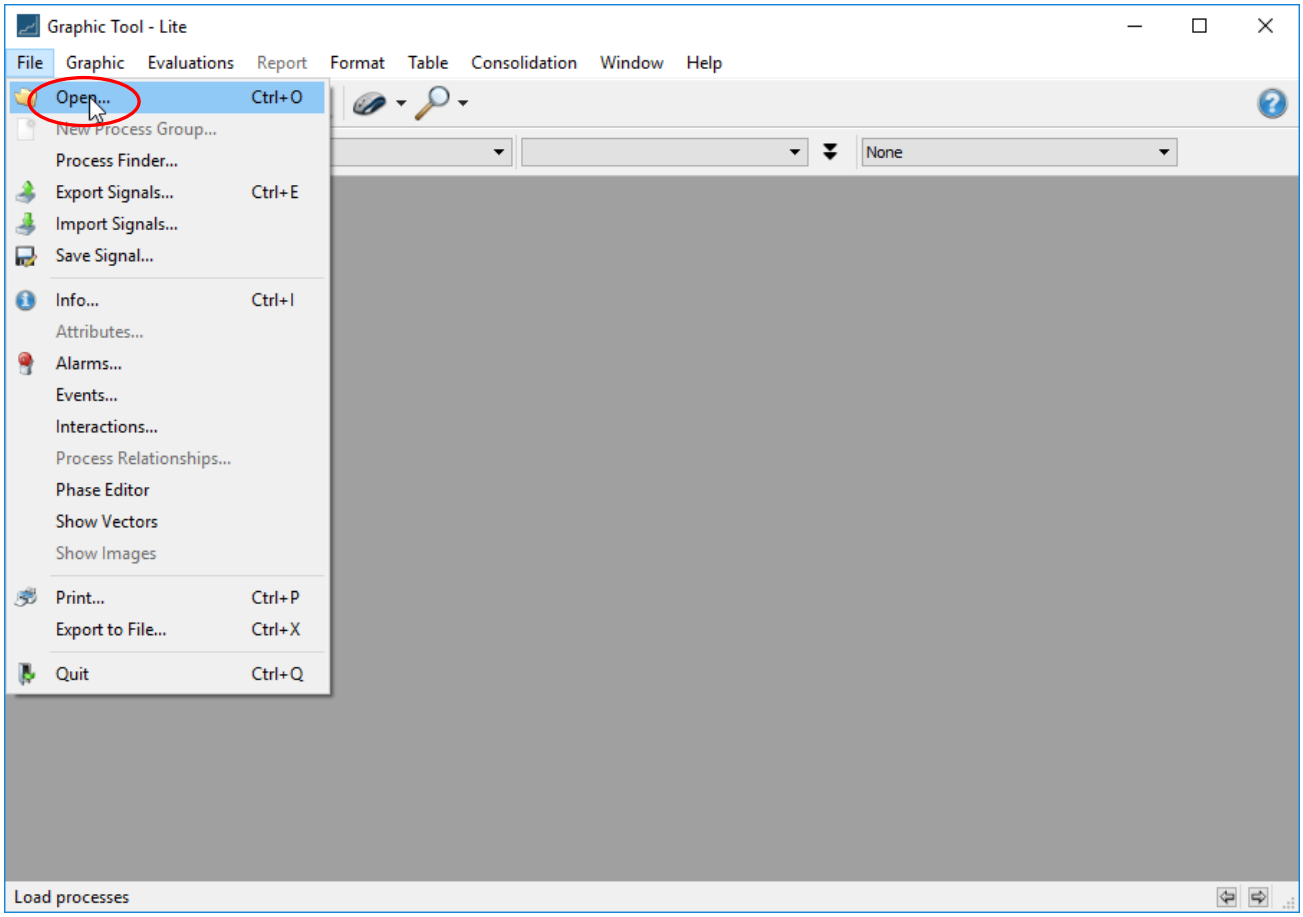
In this case, there is only one process ‘Test1’. We can select it and click on **Load**. This loads the data gathered so far and displays it, initially using the default set of axes (which you can then change):



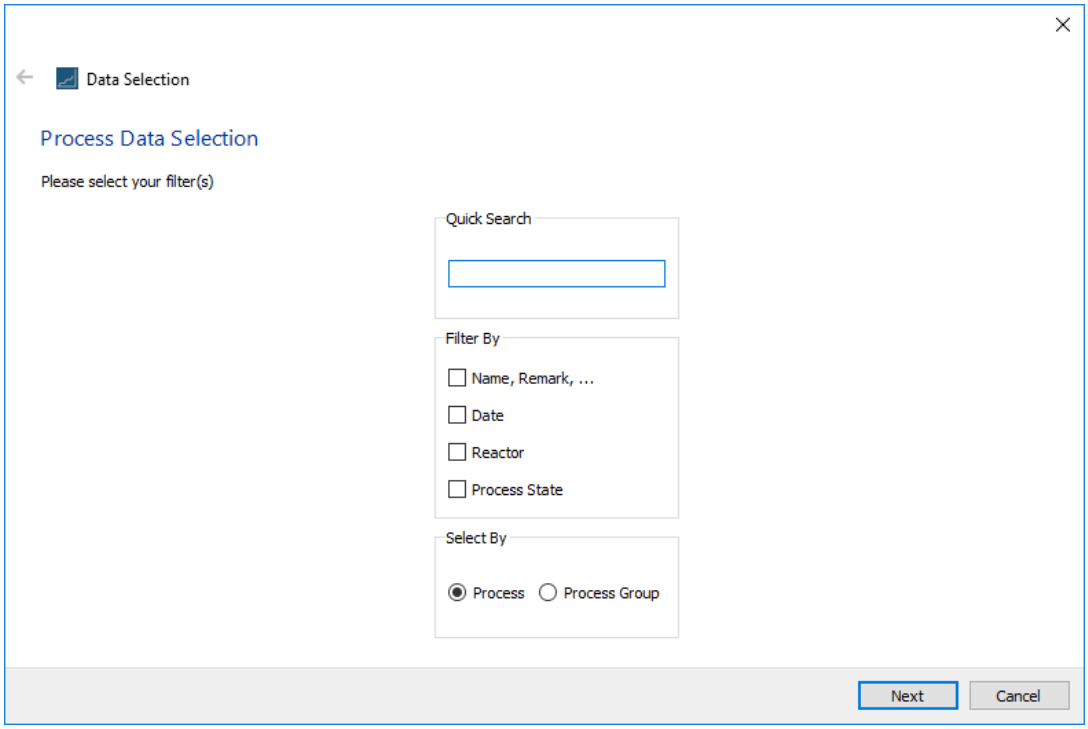
Since this is a running process, the display will continually update with new data.

3.2 LOADING HISTORICAL DATA

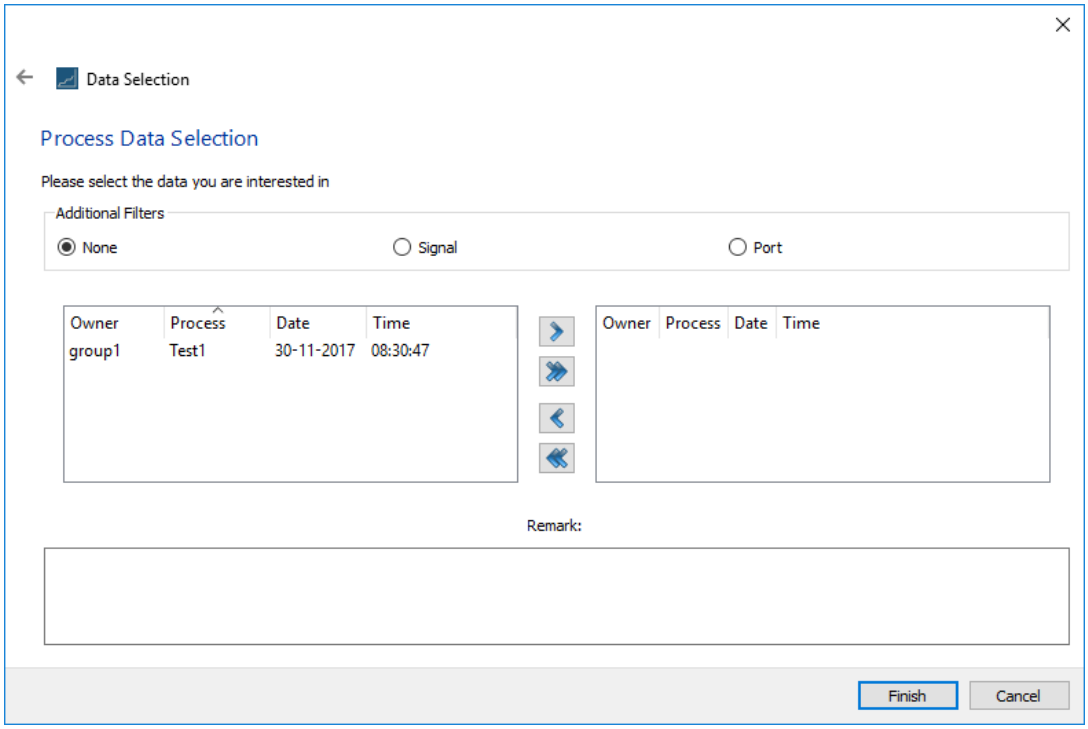
A valuable feature is the ability to recall the full data from previous experiments. You will follow this procedure very frequently. Start up the **Graphic Tool,** then navigate to **File → Open**:



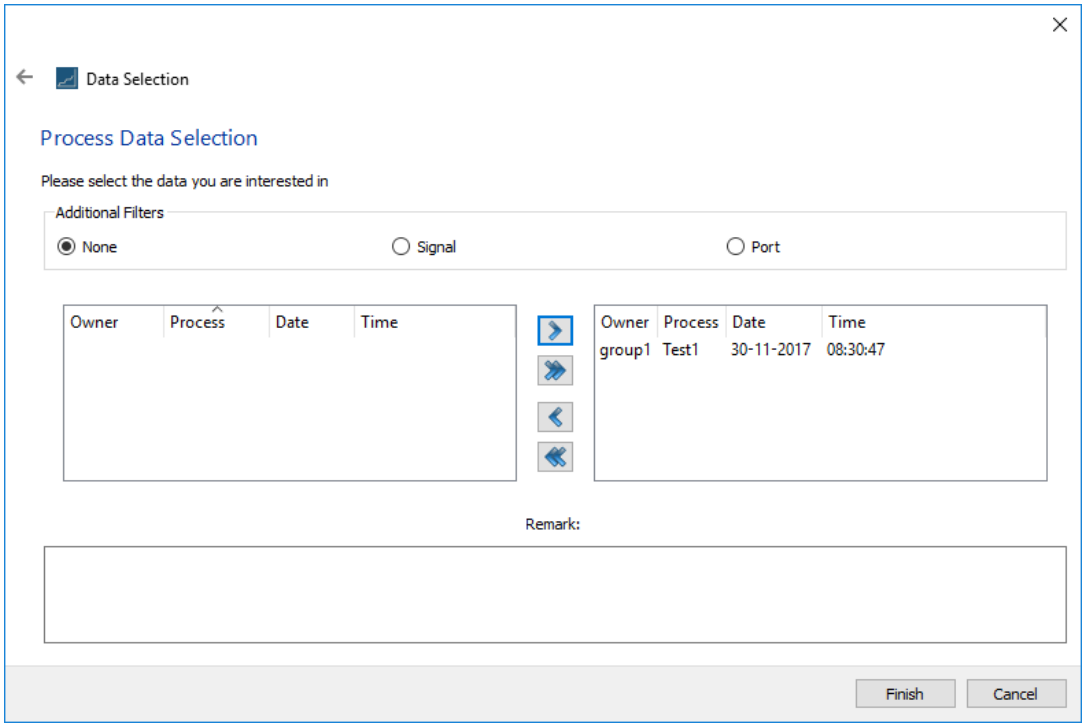
This invokes the **Data Selection** dialog:



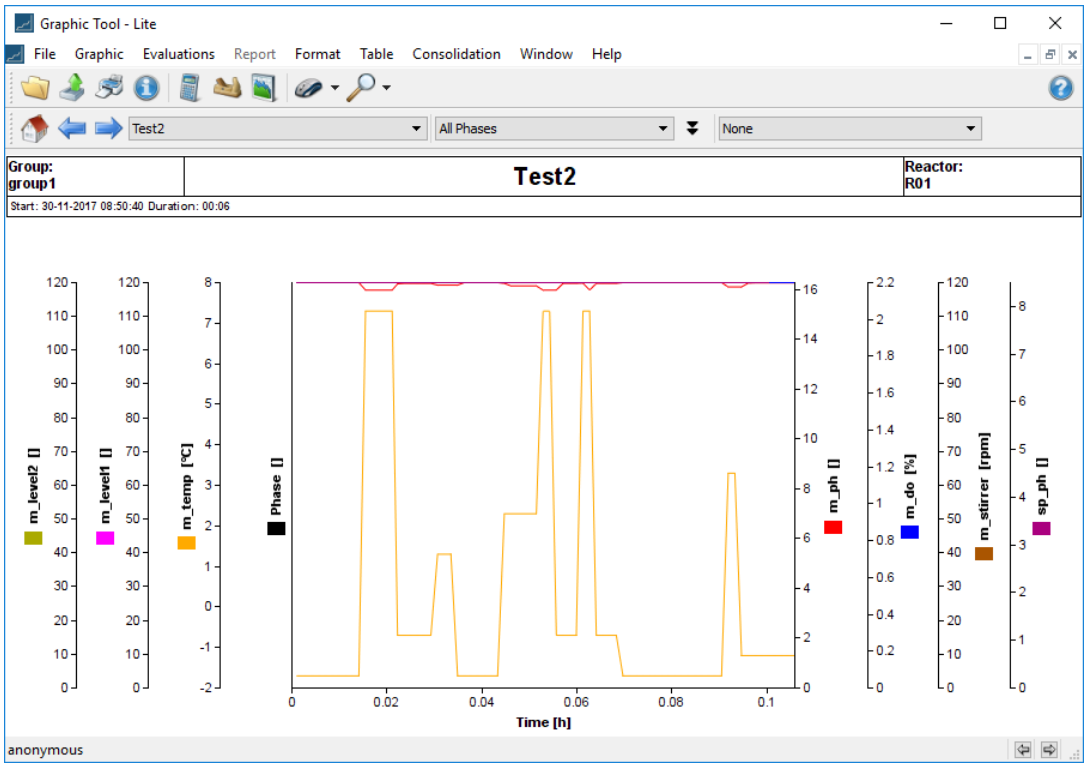
You can use the dialog to do a preselection (i.e. to filter the data sets). Otherwise, leave the fields blank and all data sets will be shown. Click on **Next**.



Select one of the entries in the list at left, press on “>” to transfer it to the right pane:



Here, process **Test1** has been selected. To complete the load, click on **Finish**:



The data has been loaded, and now one can perform analyses on it, as explained in the following sections.

3.3 USING THE CALCULATION TOOL

Once data has been loaded, one can perform operations on it. In the **Graphic Tool** one can invoke the **Evaluate** menu option (which holds all the Tools for Data Analysis) and from there, the **Calculate** menu option. **Calculate** allows one to perform mathematical operations on the loaded data, in general by composing expressions combining algebraic operations and mathematical functions.

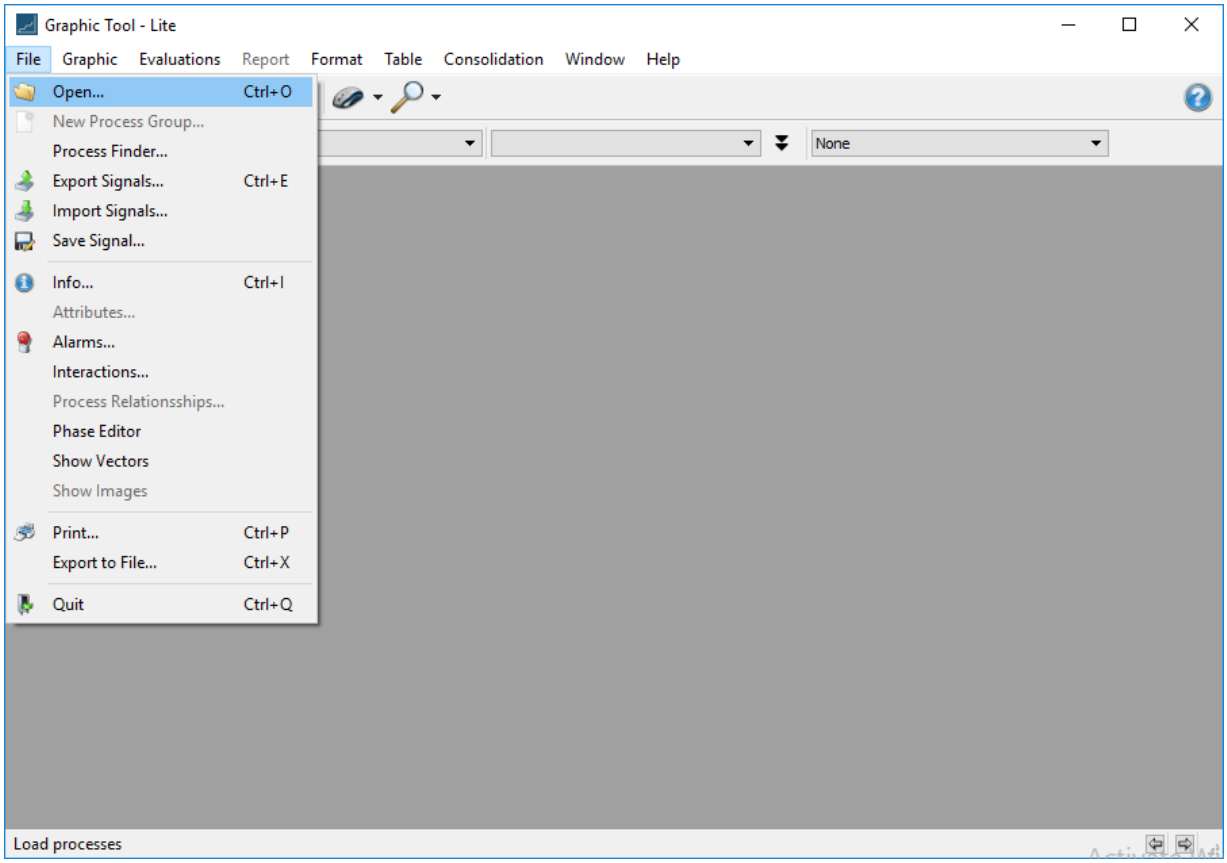
Operations	Functions
+	@abs(x)
-	@acos(x)
*	@asin(x)
/	@atan2(x,y)
%	@ceil(x)
^	@cos(x)
and	@exp(x)
or	@floor(x)
xor	@hypot(x,y)
not	@ln(x)
<	@log(x)
>	@pow(x,y)
=	@sin(x)
==	@sqrt(x)
!=	@tan(x)
	@rnd(x)
	@round(x,y)
	@integ(x)
	@deriv(x)
	@derivb(x)
	@positiv(x,y)
	@sum(x)
	@prod(x)
	@avg(x)
	@stddev(x)
	@count(x)
	@max(x)
	@min(x)
	@first(x)
	@last(x)
	@const(x)
	@ref(x)
	@if(A;B;C)
	@exists(x)
	@numberof(x)
	@numberof(x,h)

Examples of valid formulas are:

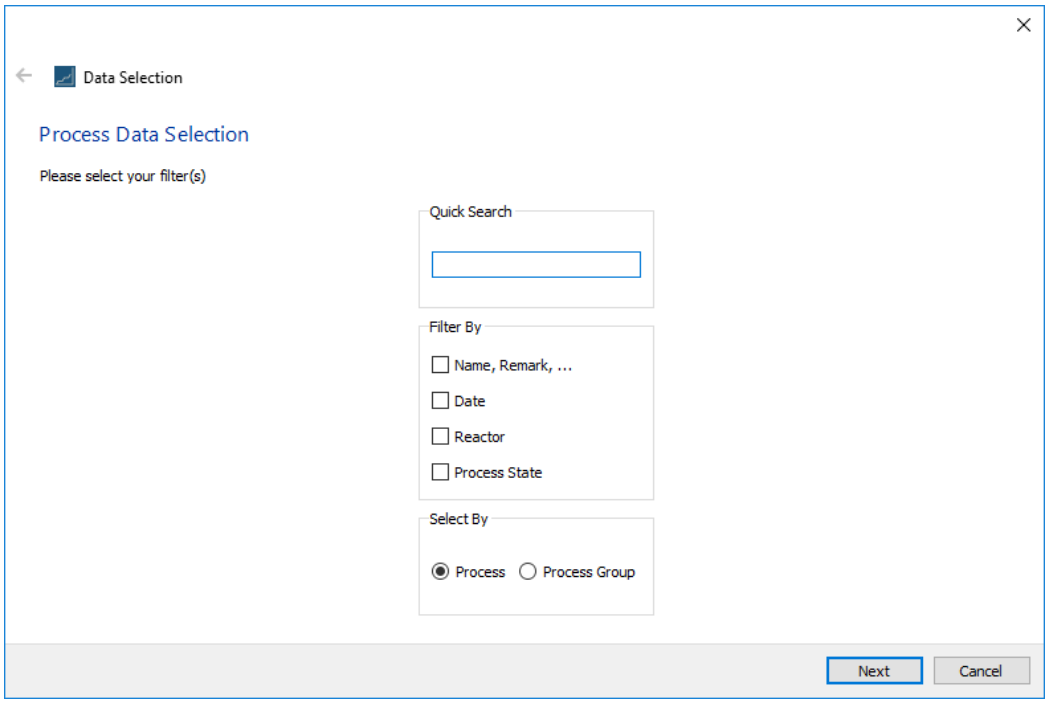
```
oxid := @sin(P0 * P1(0:10)) * @sqrt(P2(10:20))
actuation := @exp(T)

K := 32
TEMP_F   := 9/5*temp_c + K
```

Proceed as follows. Launch the **Graphic Tool** and if no data has been loaded yet, you can load it now via **File**

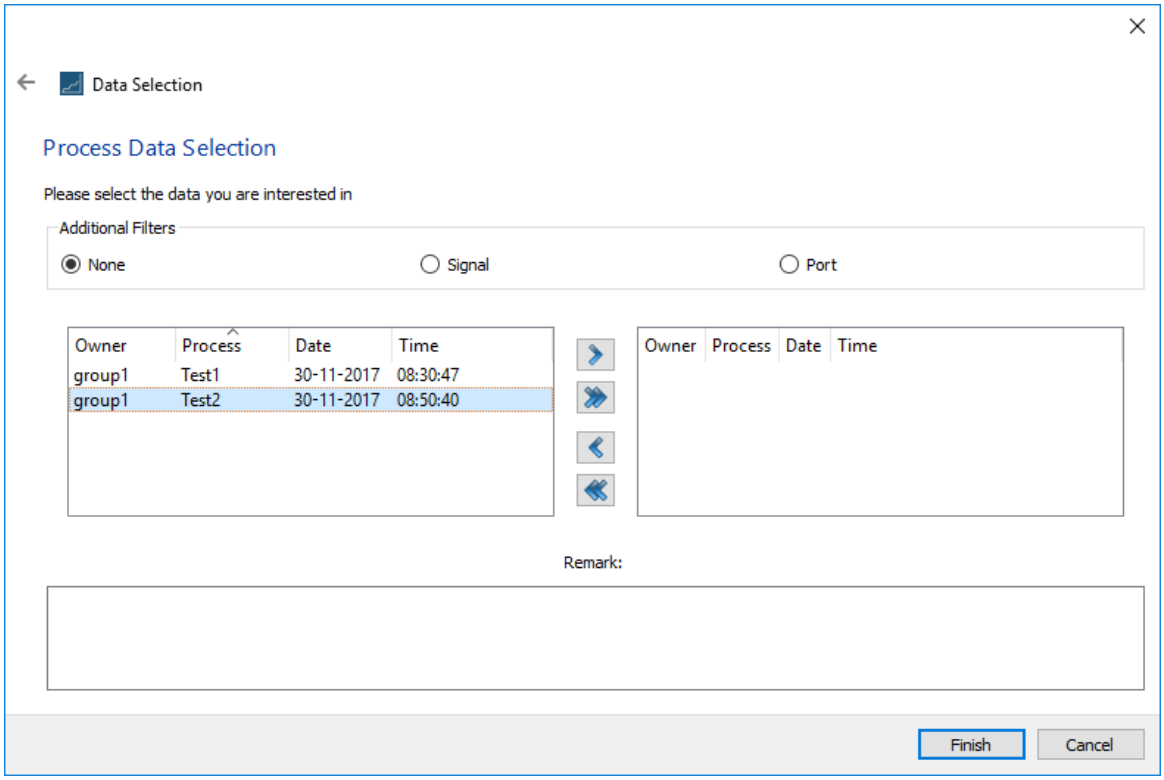
→ **Open**:



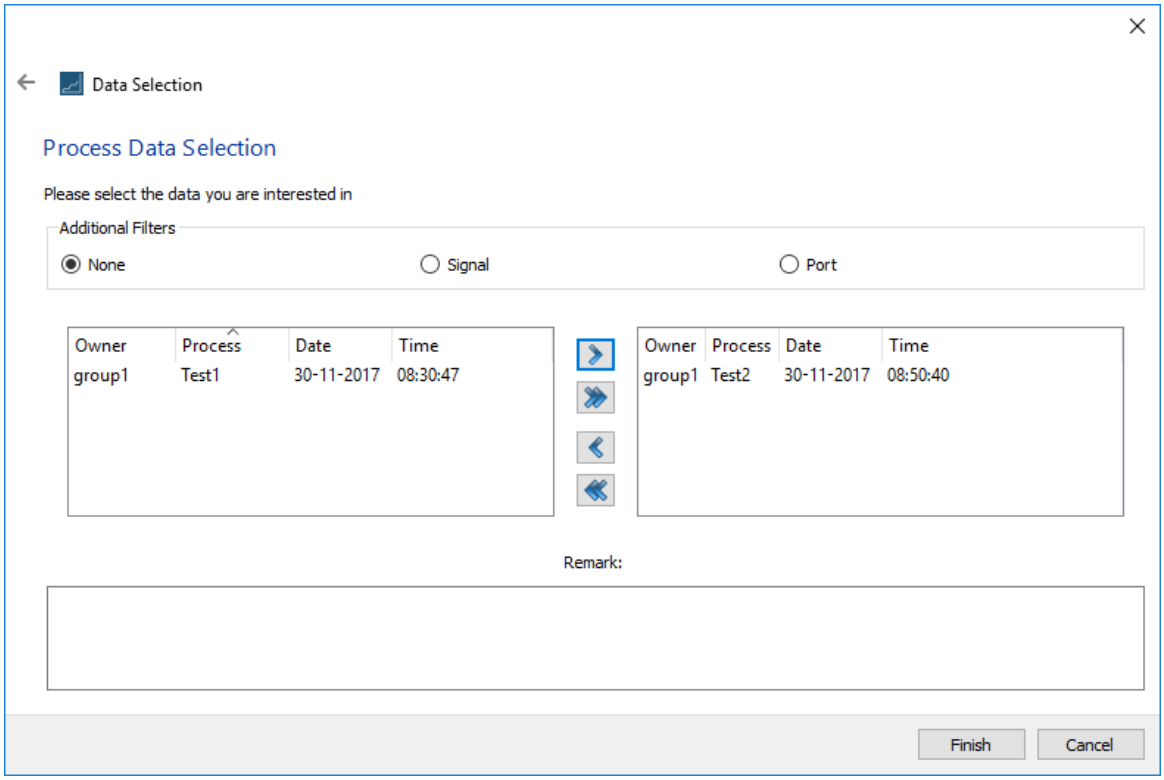
This will invoke a data set selection dialog:



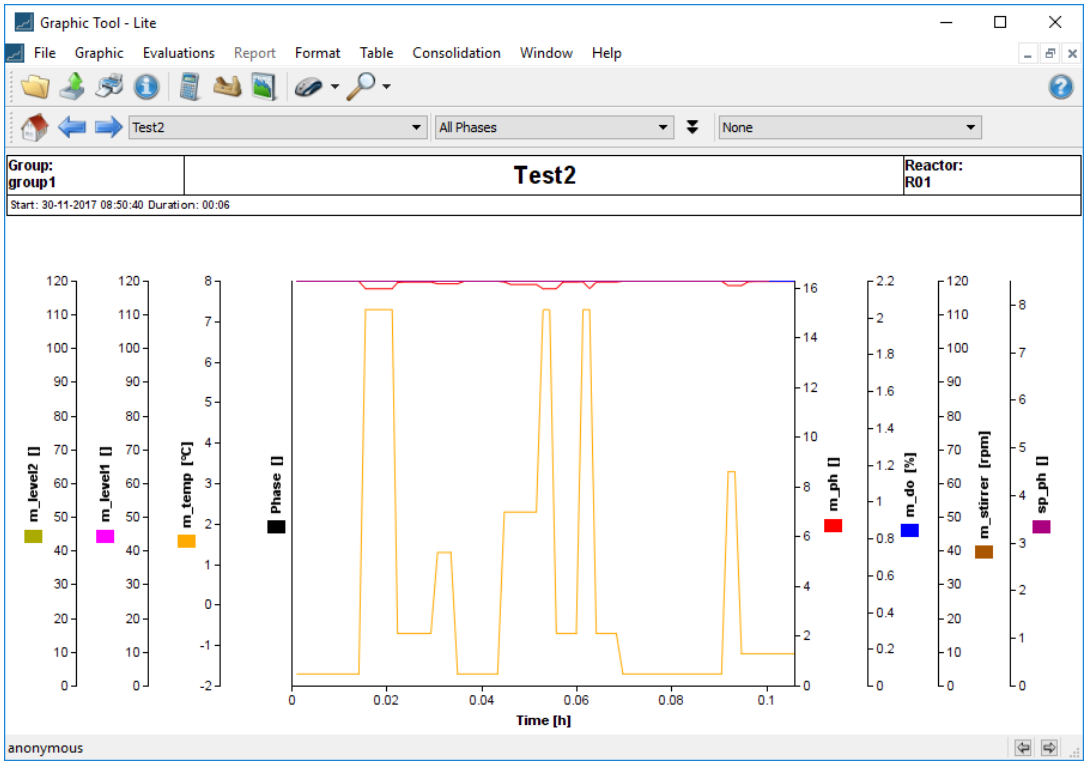
Use the above criteria to filter the collection, otherwise just leave everything blank – and this will display all data sets ever created on the system:



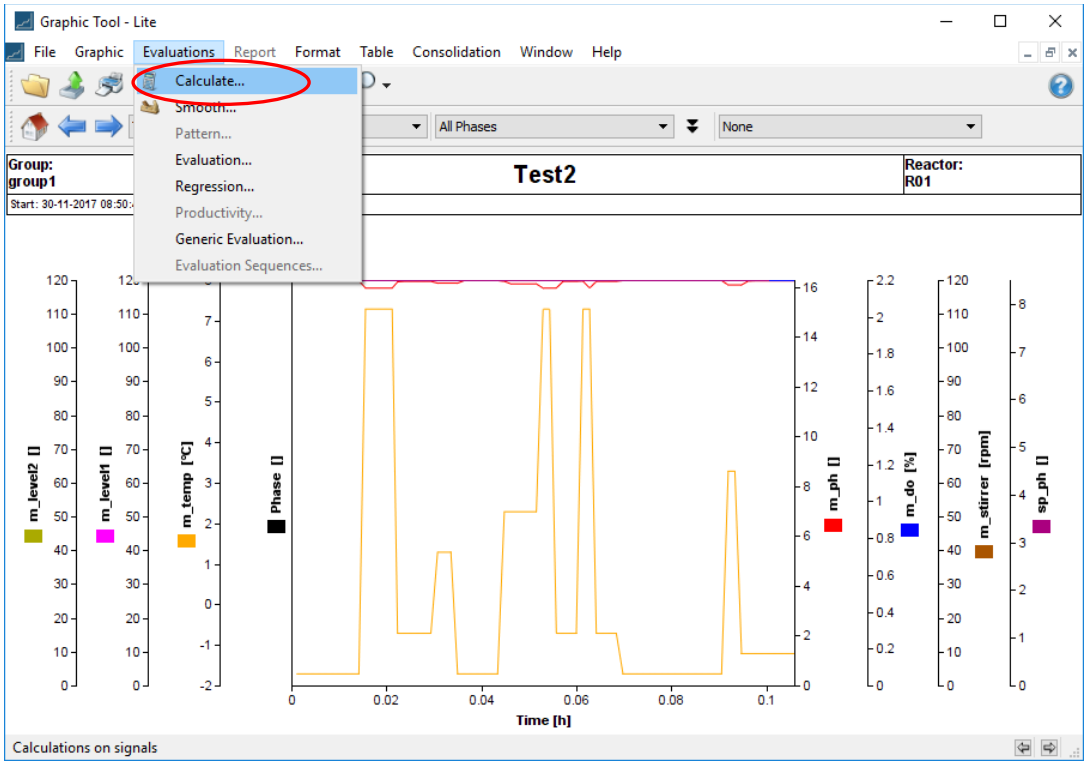
The above shows “Test2” being selected. Click on ‘>’ to transfer it to the right pane:



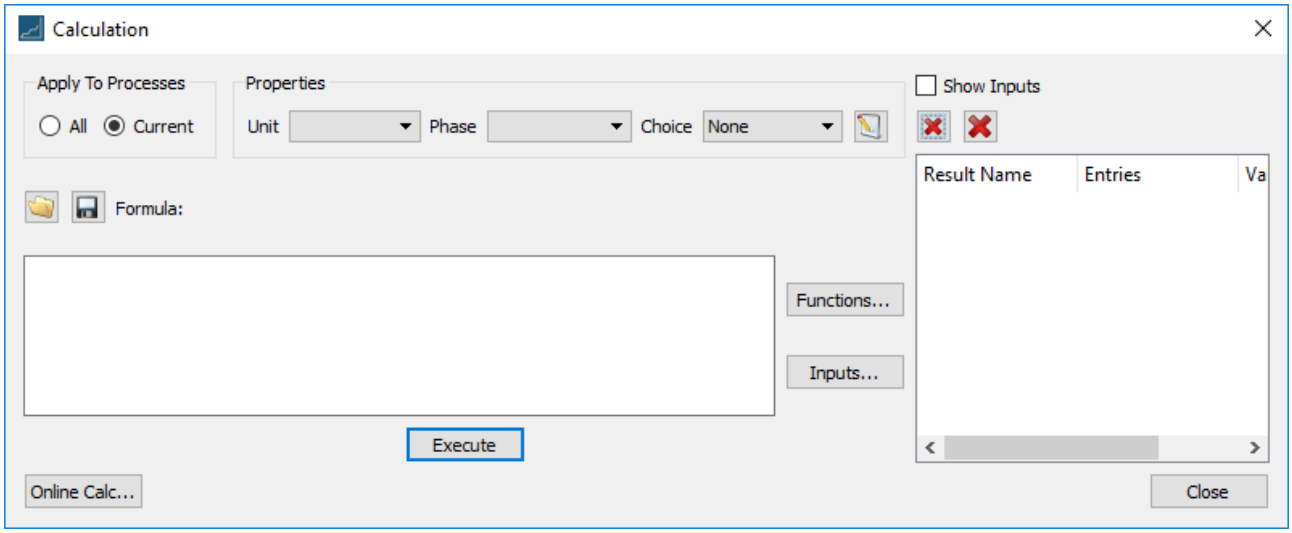
Then click on finish and a trend graph of the selected data is shown:



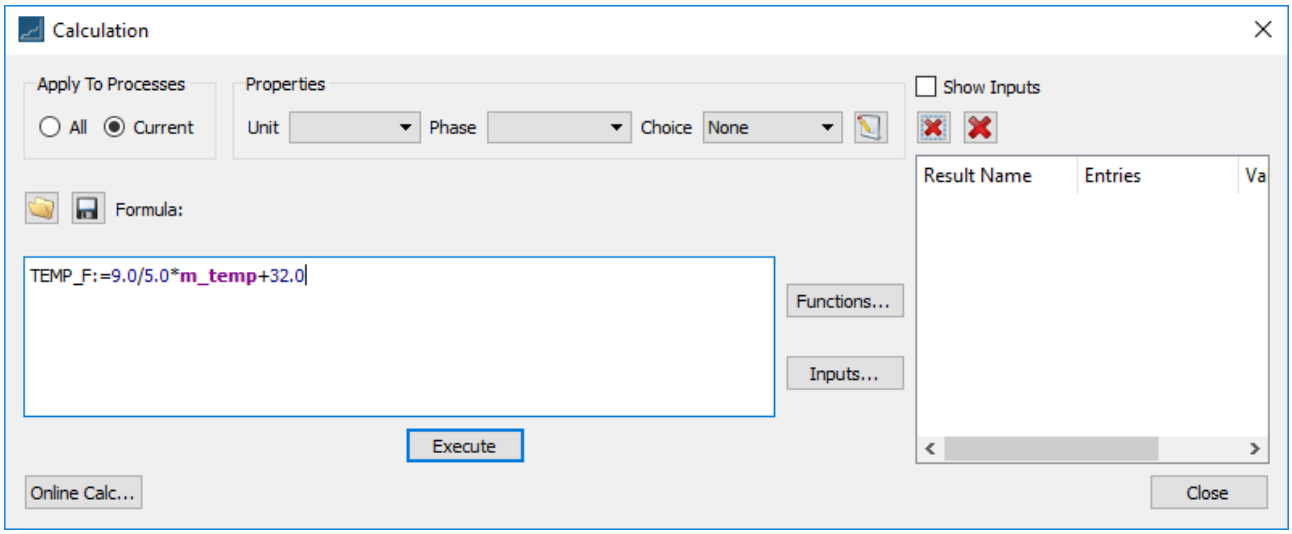
Now we define a calculation. Navigate to **Evaluations → Calculate**.



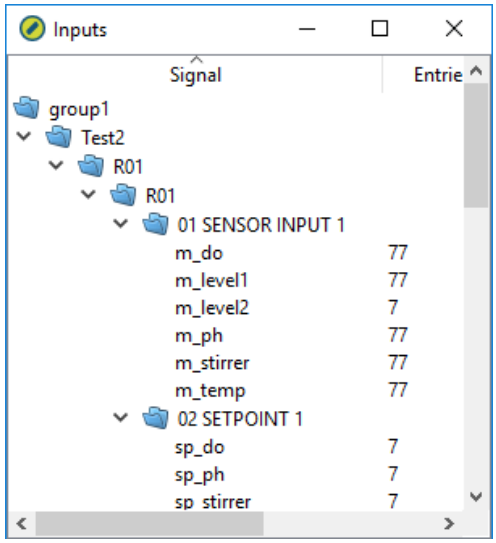
Select **Calculate** and this displays the following screen:



Enter the Celsius to Fahrenheit conversion formula in the left *formula* input field as follows:



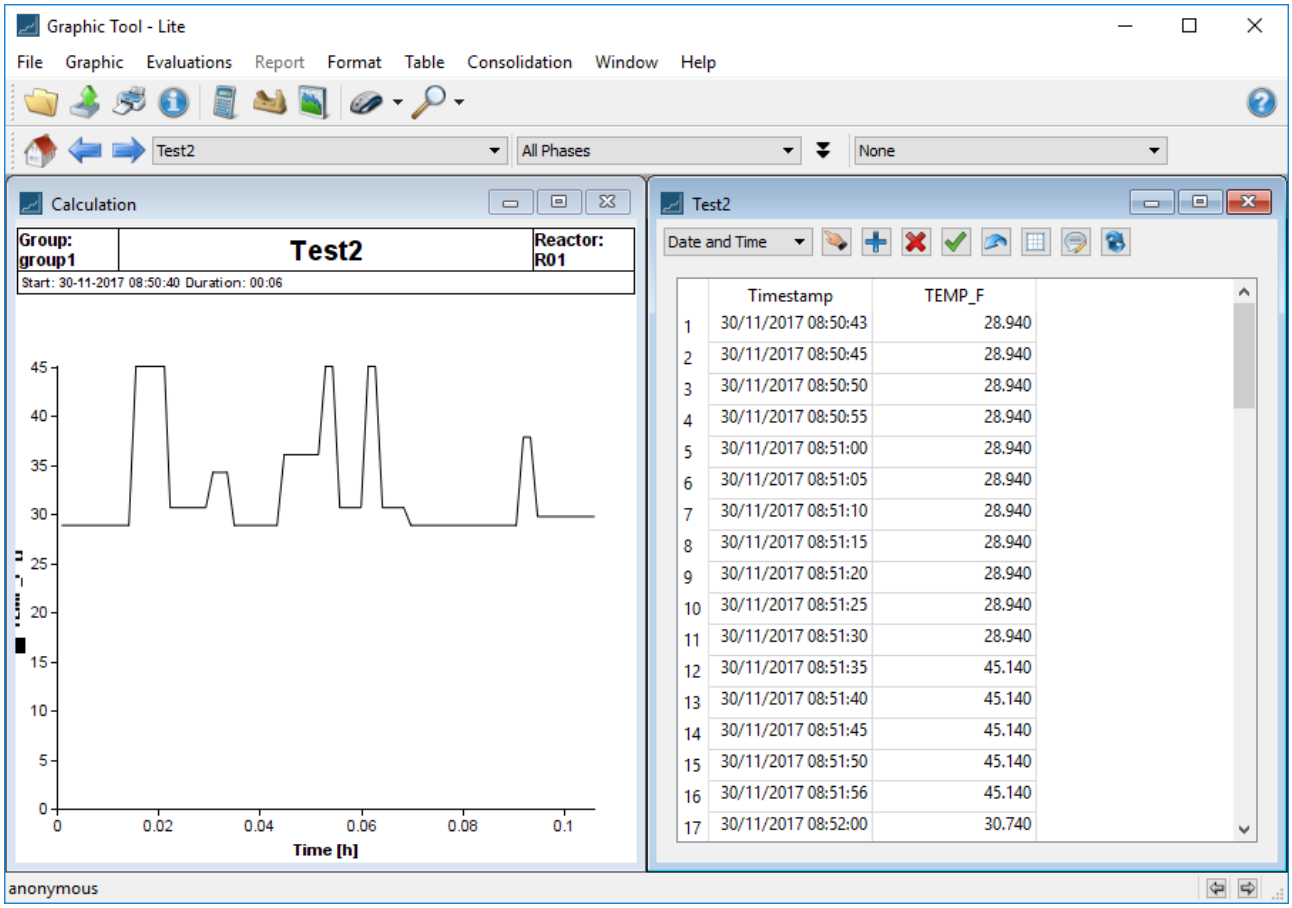
In writing the above formula, we needed to determine the name of the measured temperature variable. Clicking on the **Inputs …** button displays all known variables. In the 01 SENSOR INPUT 1 "folder" for sensor inputs, we see **m_temp** which must be the temperature process variable.



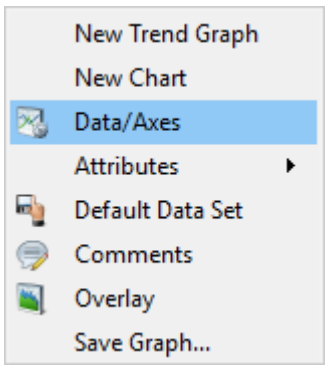
Temperature is measured by default in Celsius and we wish to display the temperatures trend in Fahrenheit. So we enter the formula as shown above, which is the basic conversion from Celsius to Fahrenheit (note: you can transfer m_temp to the formula by doueble-clicking on it in the Inputs window).

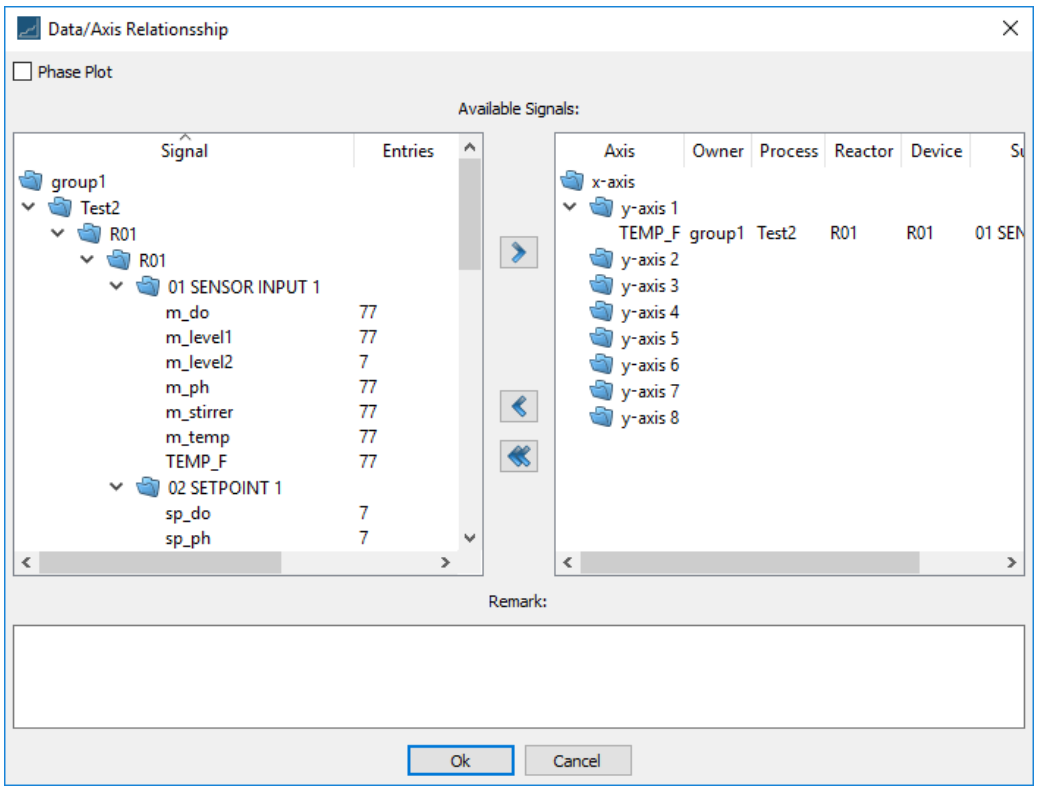
Then click on **Execute**.

This will create the variable **TEMP_F**, calculate its values and will automatically display by default the values of the new variable (or variables) in graphical and tabular form:

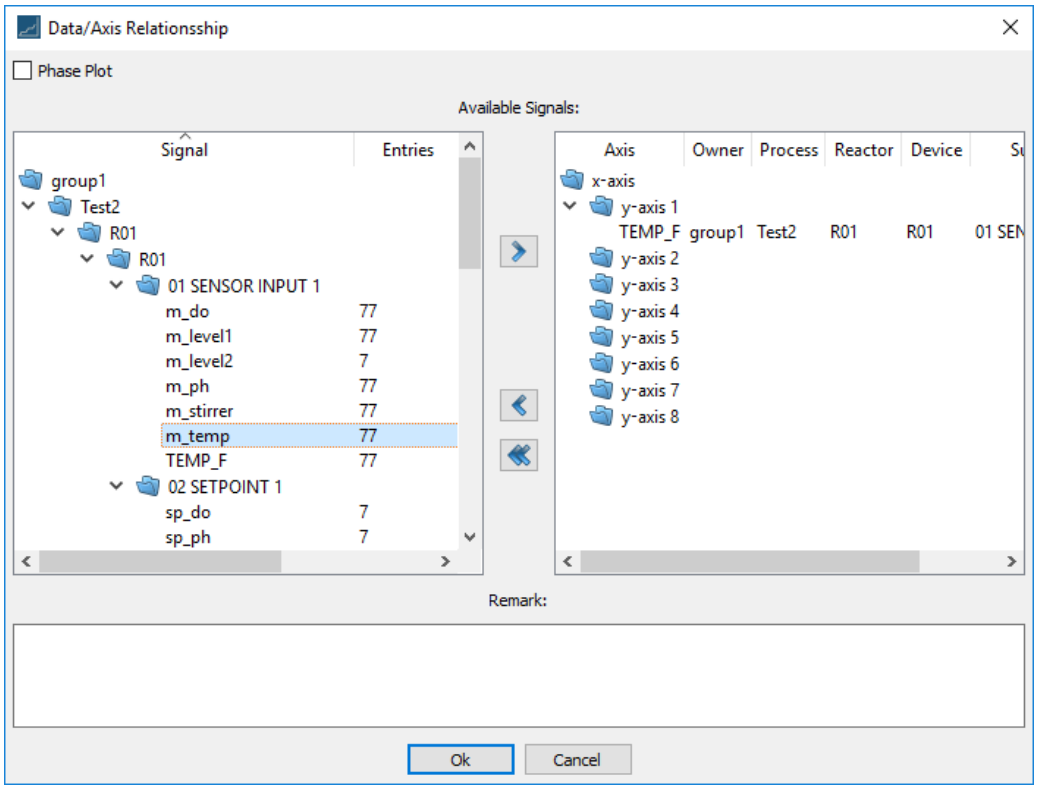


You can now choose to include the original measured values. Navigate to **Graphic → data/axes**:

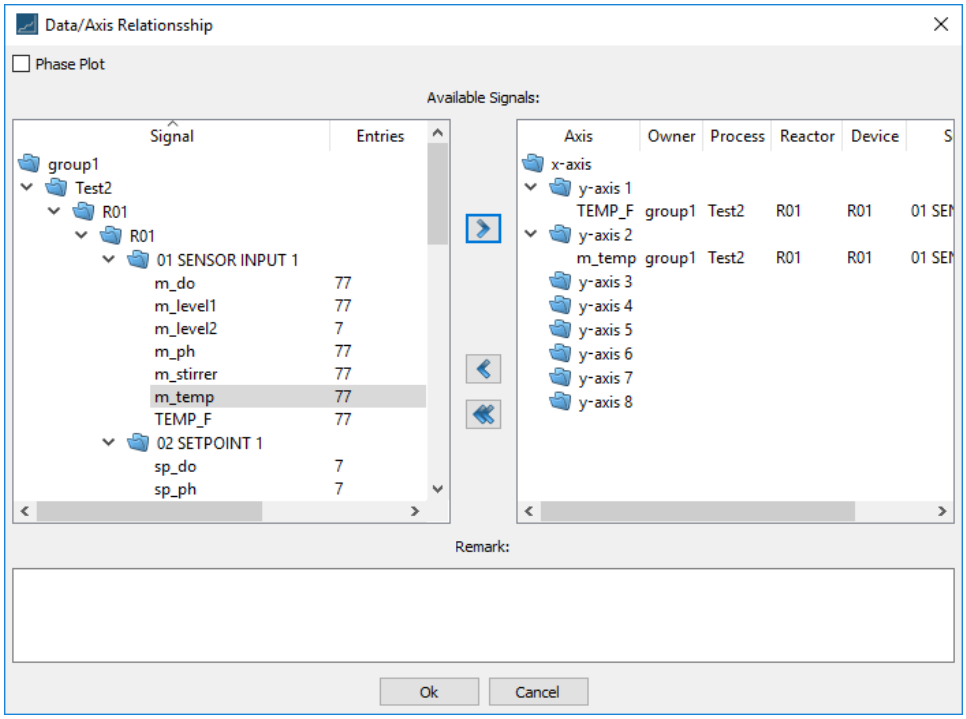




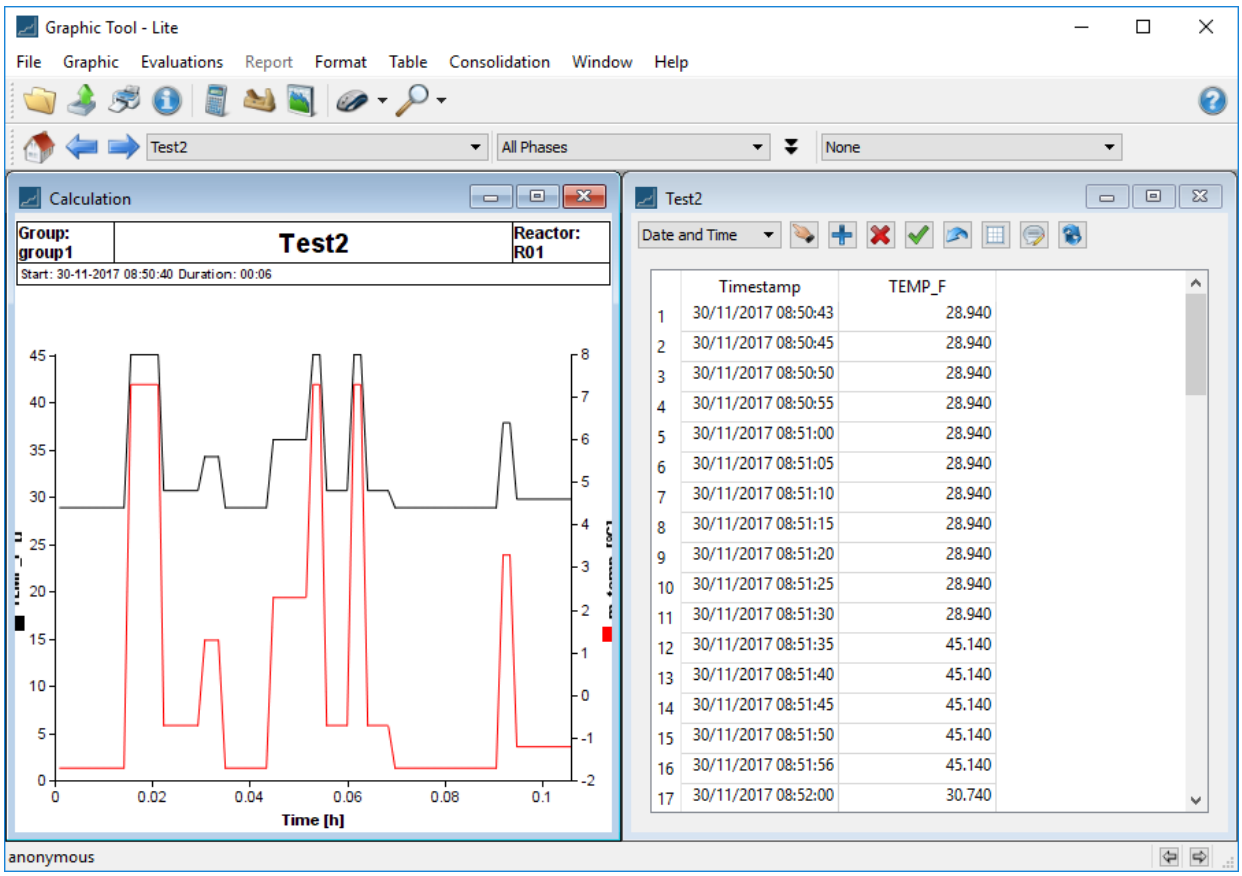
The calculated variable **TEMP_F** is shown. Select **m_temp** in the left pane:

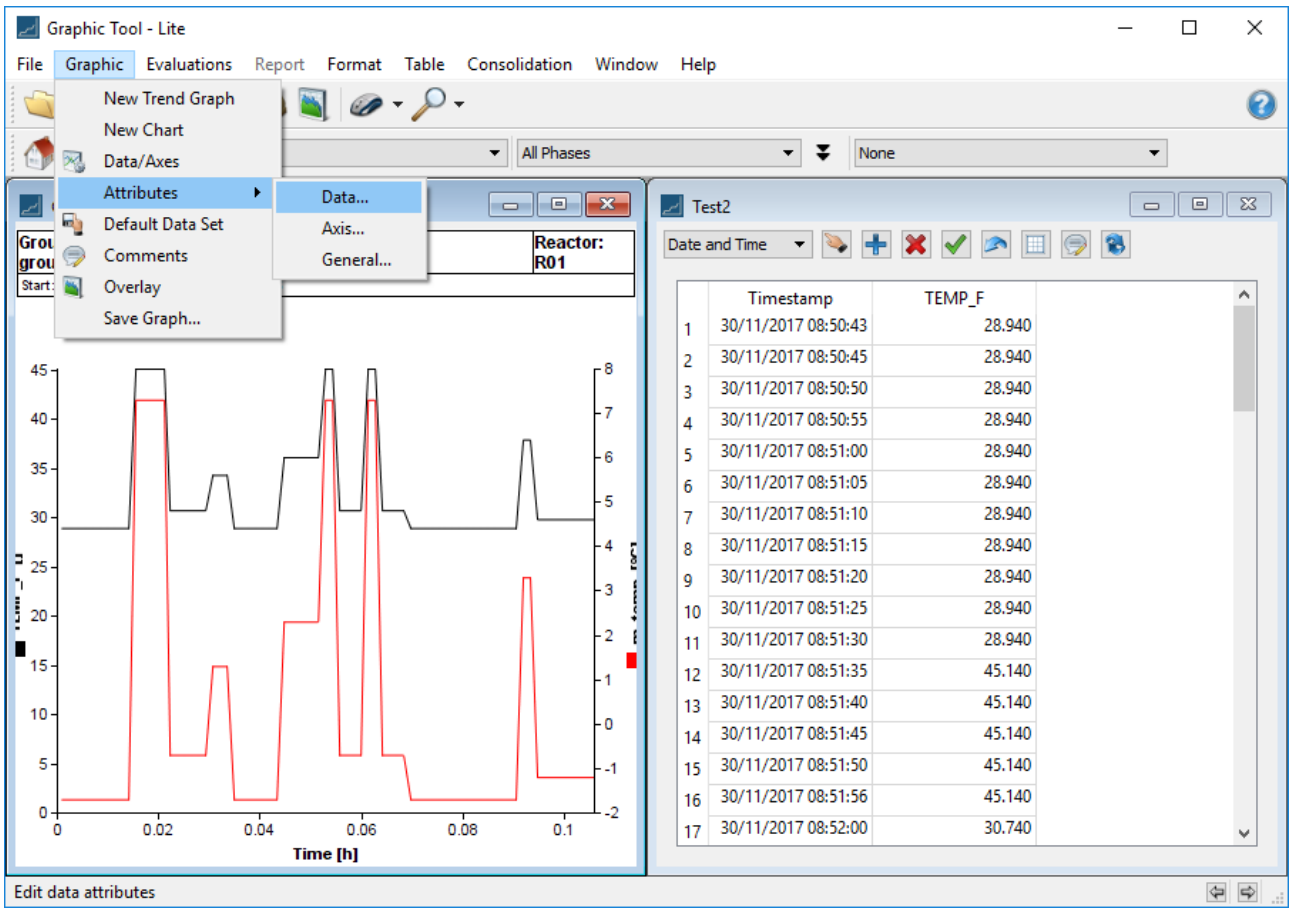


Click on '>' to transfer **m_temp** to y-axis 2:



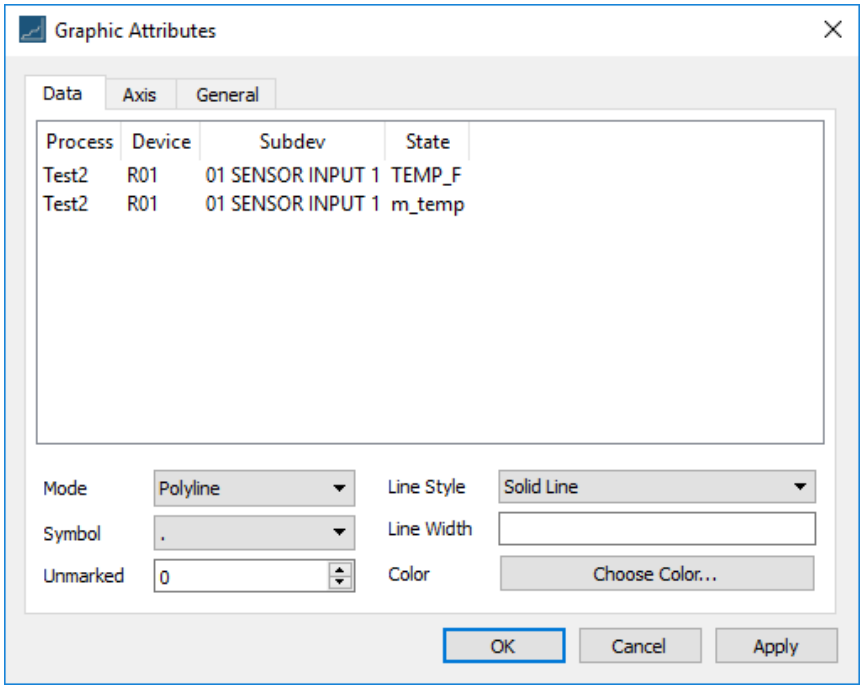
Click on **OK**, and now both variables are displayed:



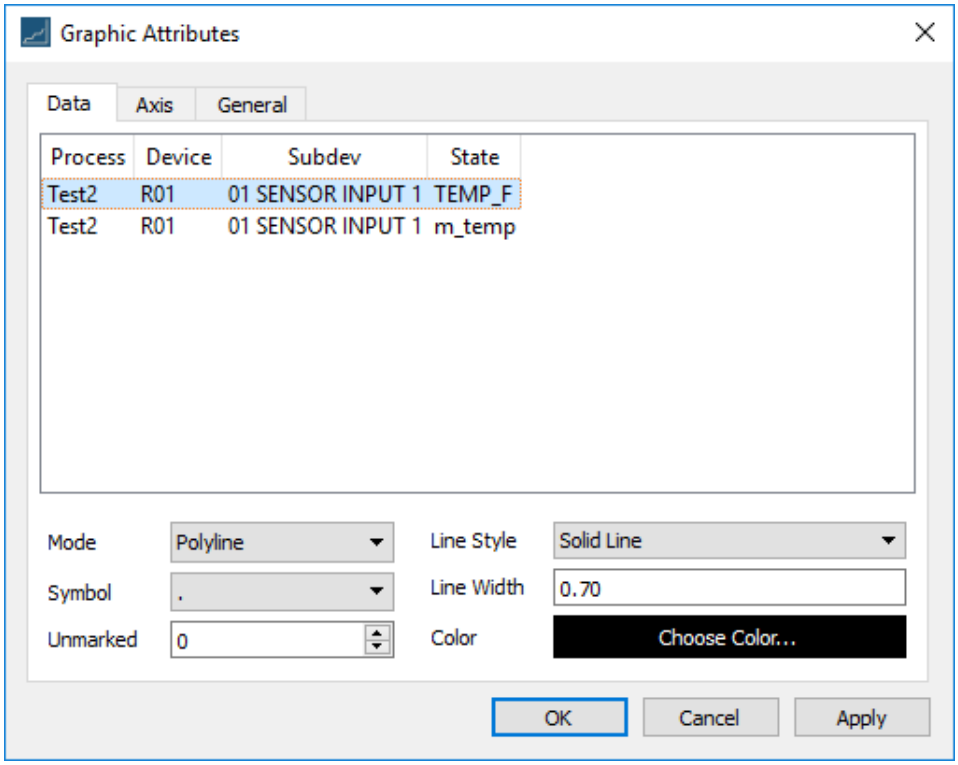


Here we wish to change the colors of the data plots. So Navigate to **Graphic → Attributes → Data** as shown above.

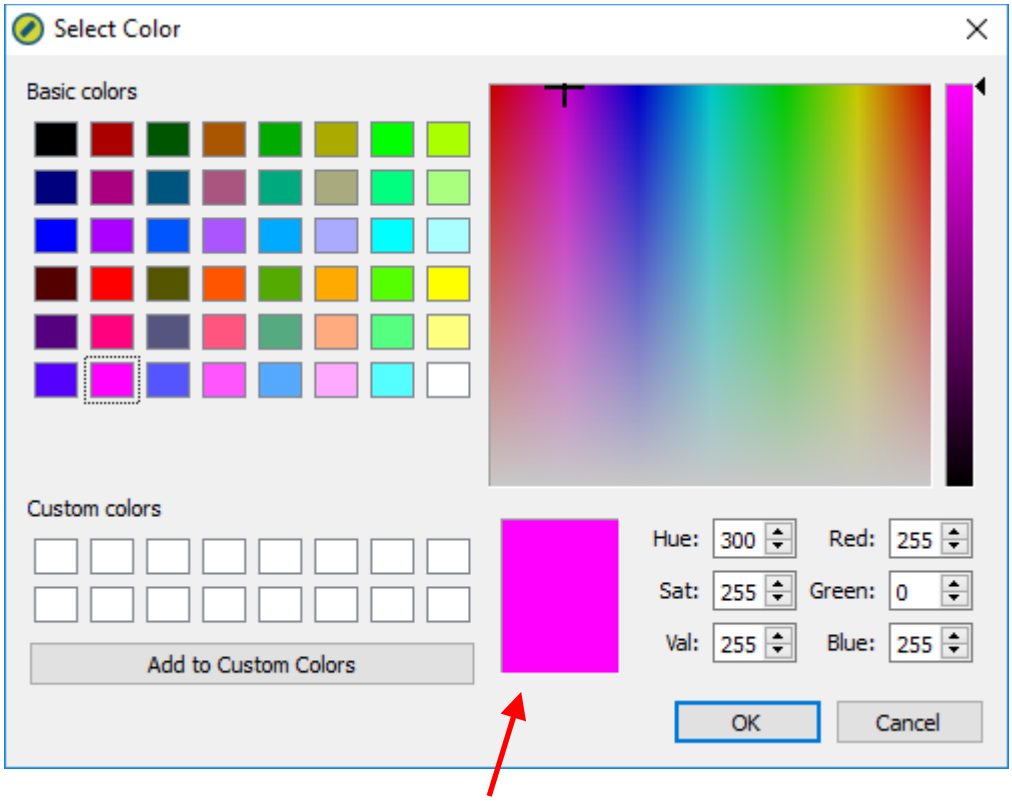
This invokes the dialog shown at right where one can change attributes of the plot for the listed data sets.



Select the TEMP_F data set:

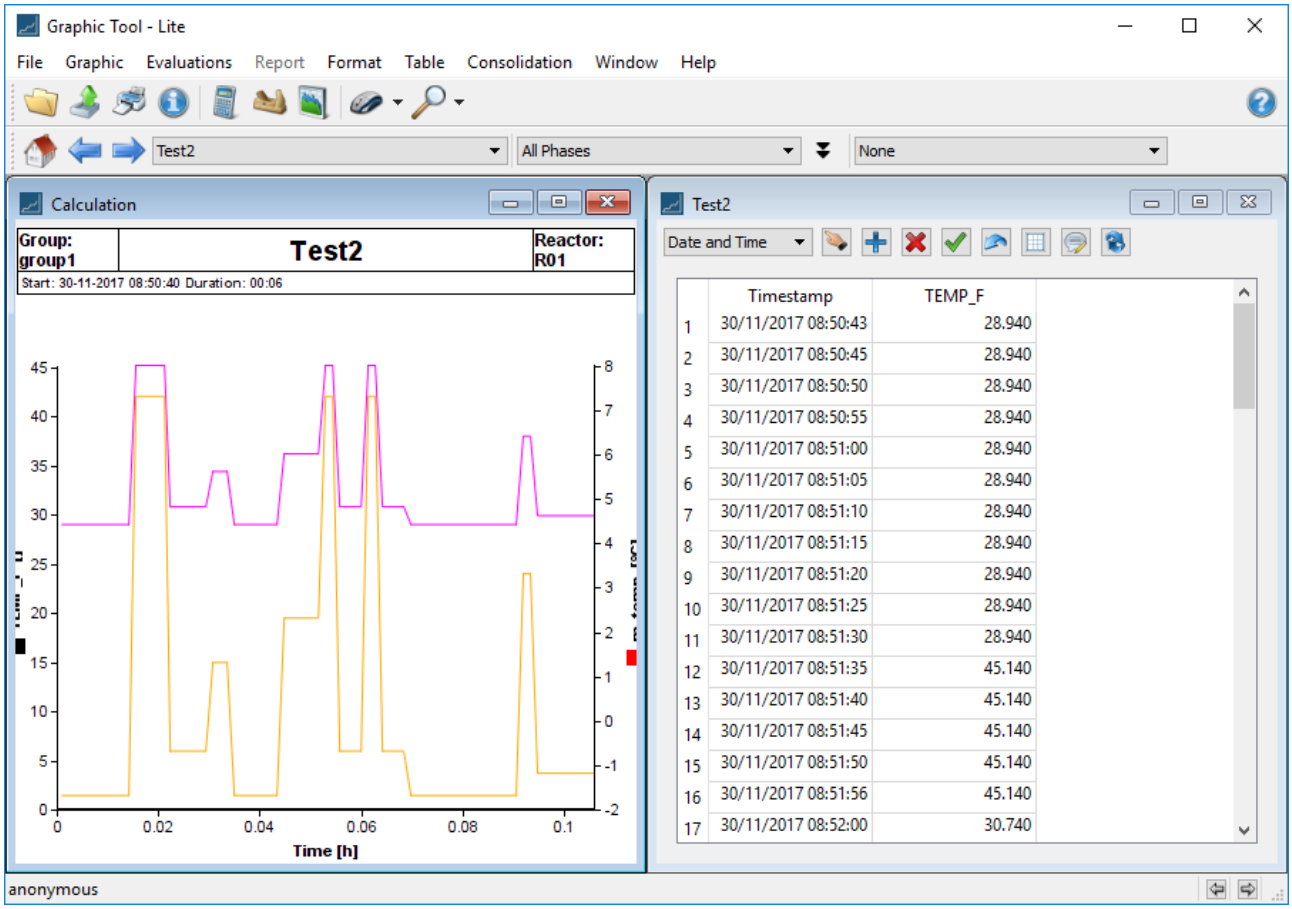


And the color button takes on the current color of the **TEMP_F** plot. Click on **Choose Color …** :

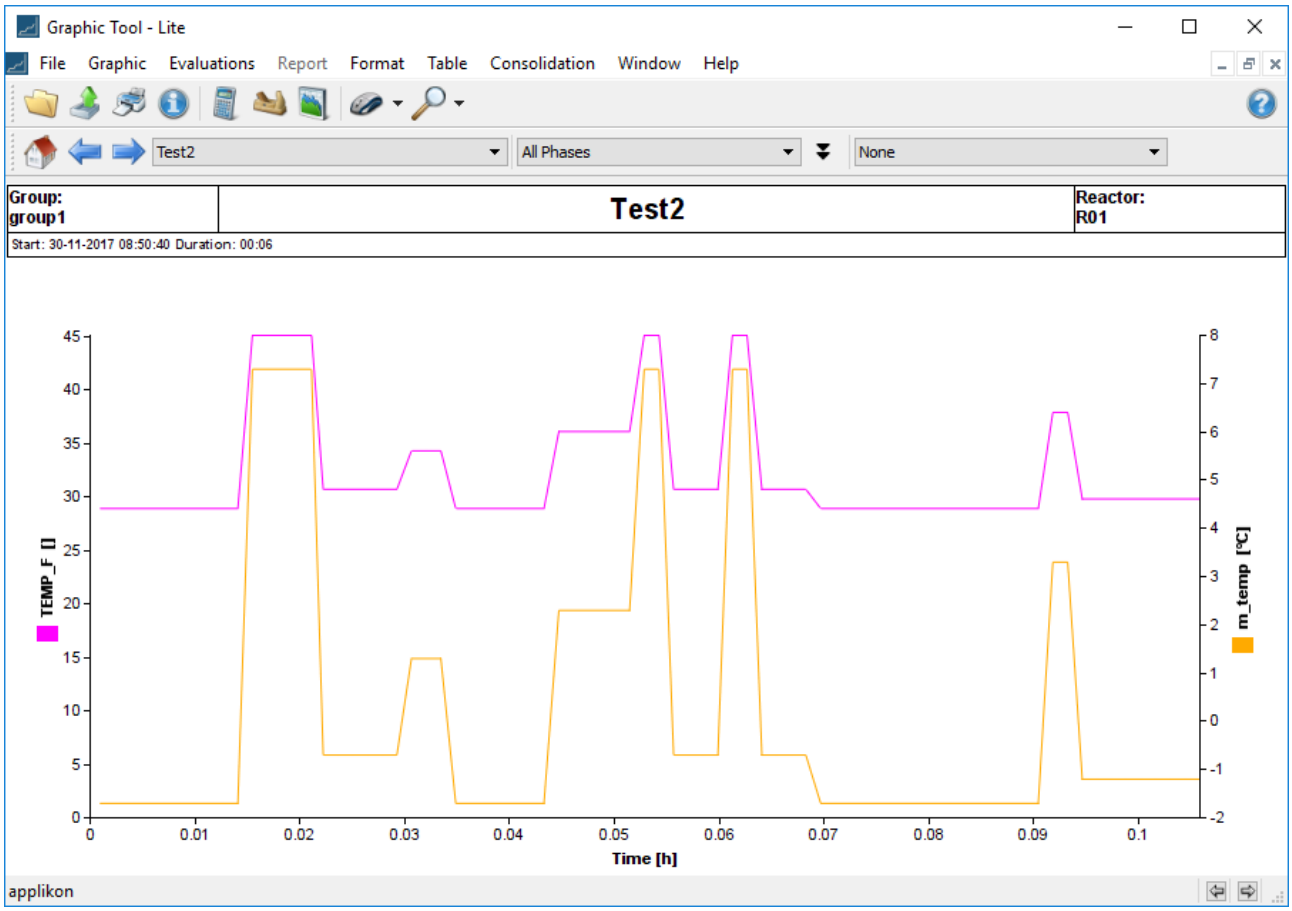


We select pink in the **Select Color** dialog. The selection is confirmed by the colored rectangle (indicated by the red arrow). Click on **OK** and this makes our TEMP_F curve pink.

Then we perform the same action for **m_temp** and change its color to orange. Click on **OK** to display the result:



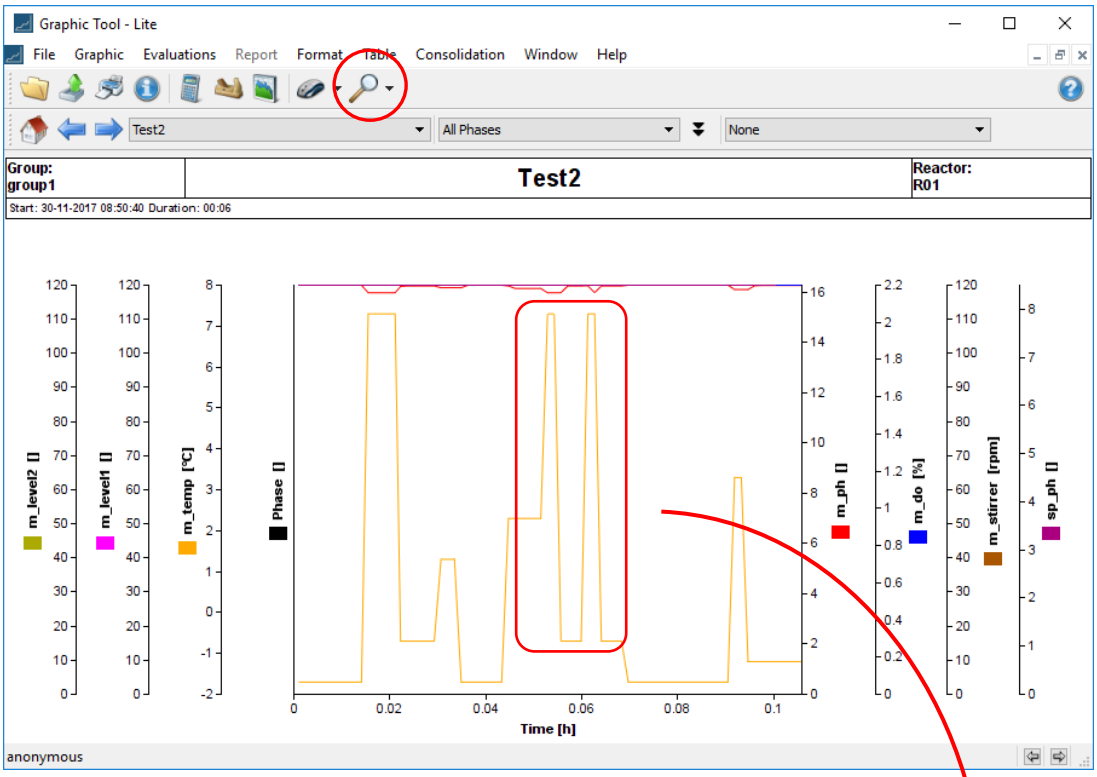
Finally, close the tabular display and maximize the trend:

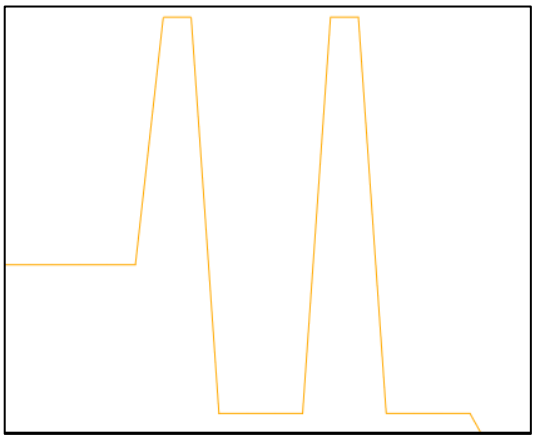


The above now shows the temperature measurement in Centigrade (orange) and the corresponding Fahrenheit value (pink). This is admittedly a simple example but the calculation capability allows you to make far more complex computations.

3.4 SMOOTHING GRAPHS

Smoothing is useful function for showing a general trend by smoothing out fluctuations in data. Proceed as follows. Launch the **Graphic Tool**. If no data has been loaded yet, you can load it now via **File → Open**, as was done in the previous section.

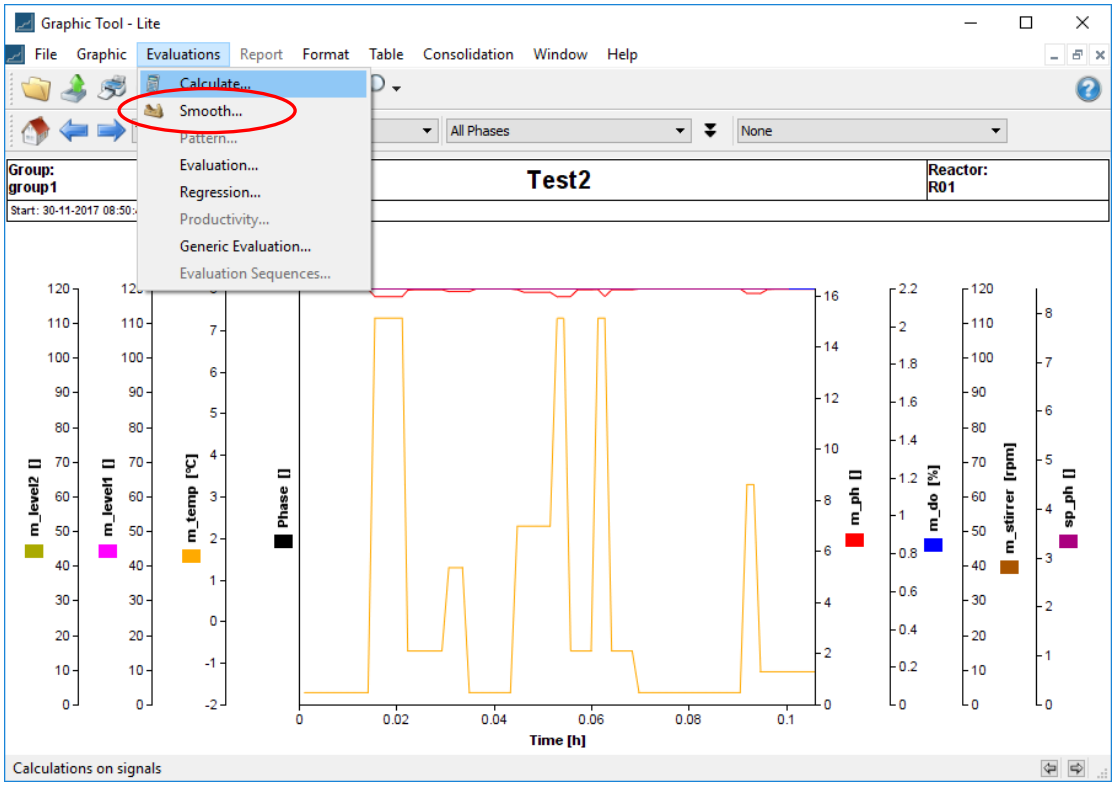




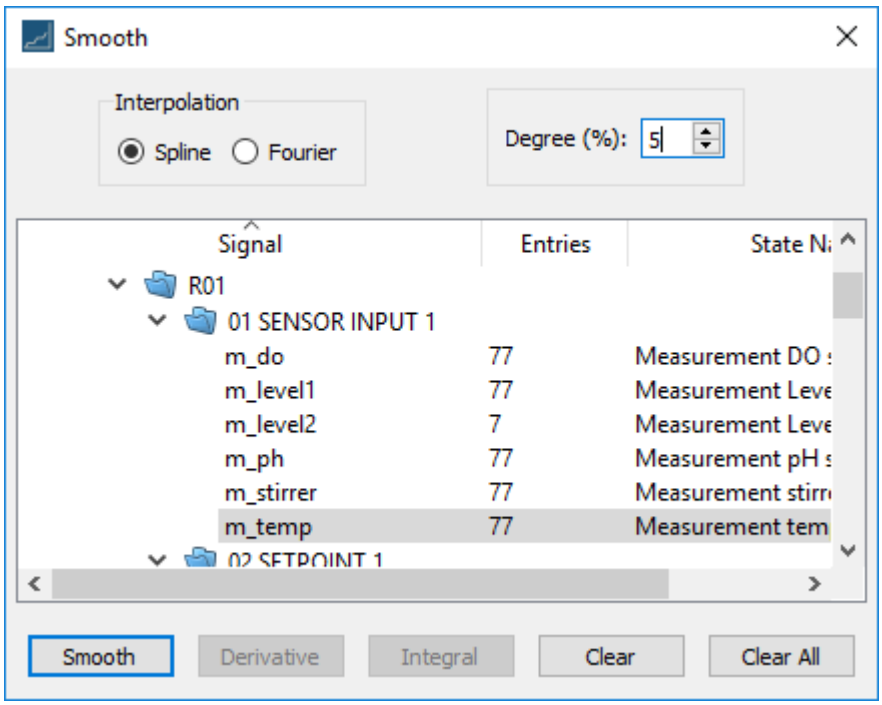
If necessary, zoom into a section of the graph in order to more clearly demonstrate the effect of smoothing and smoothing will be applied to the selected portion.

Use the hourglass button (circled) to zoom in. The cursor changes to a magnifying glass and you drag with the left mouse button to make a rectangular selection which is then zoomed into. If no selection is made, the smoothing will be applied to the entire data set.

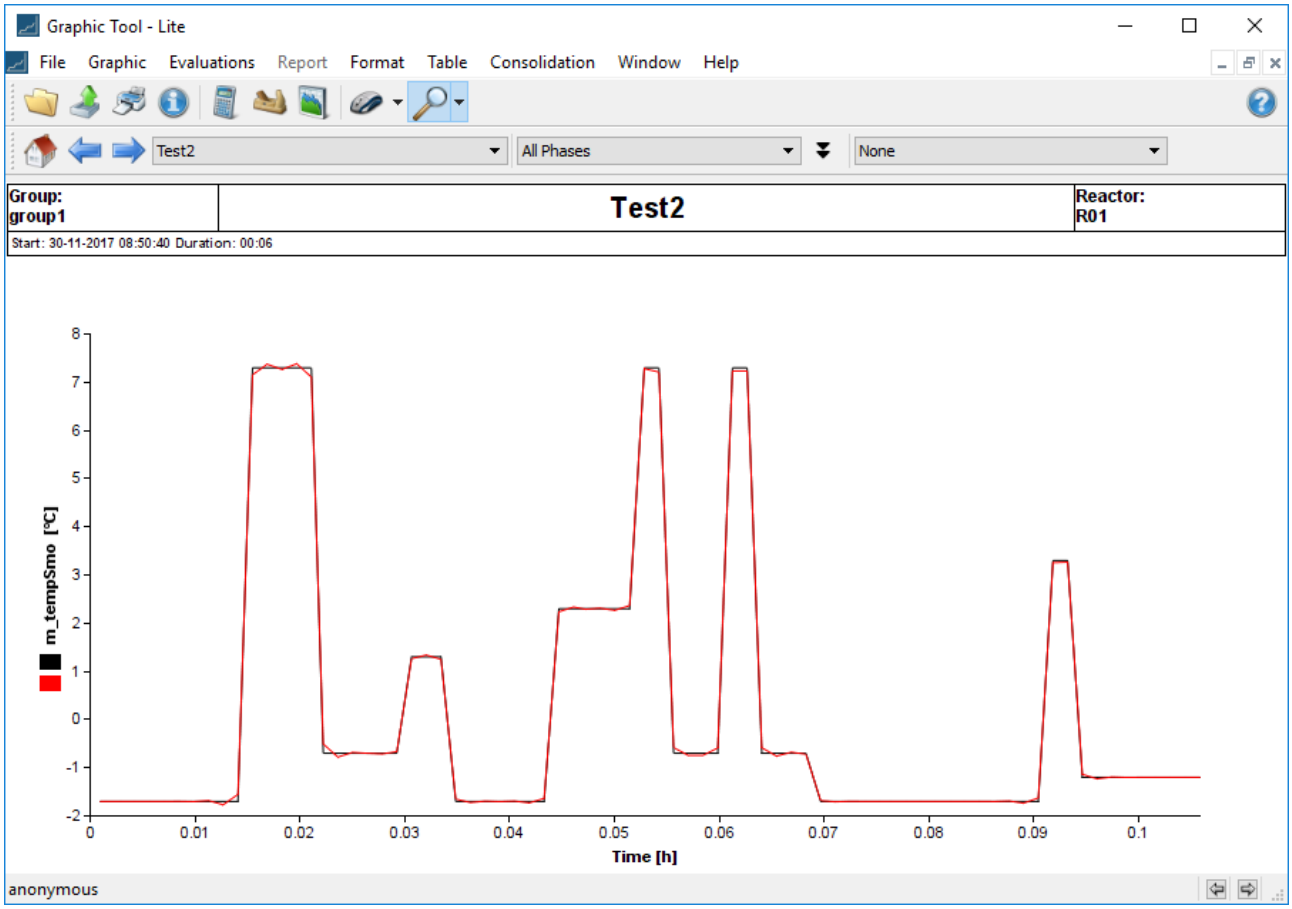
Navigate to **Evaluations → Smooth**.



Clicking on Smooth will display the following dialog:



We can select between Spline or Fourier methods for smoothing. Here we select **Spine** and set the **Degree** of smoothing to 5%. Select **m_temp** and click on **Smooth**.
The result of this is the following:



The original curve in black has been overlaid by the smoothed curve in red. As one can see here, 5% does not do very much smoothing.

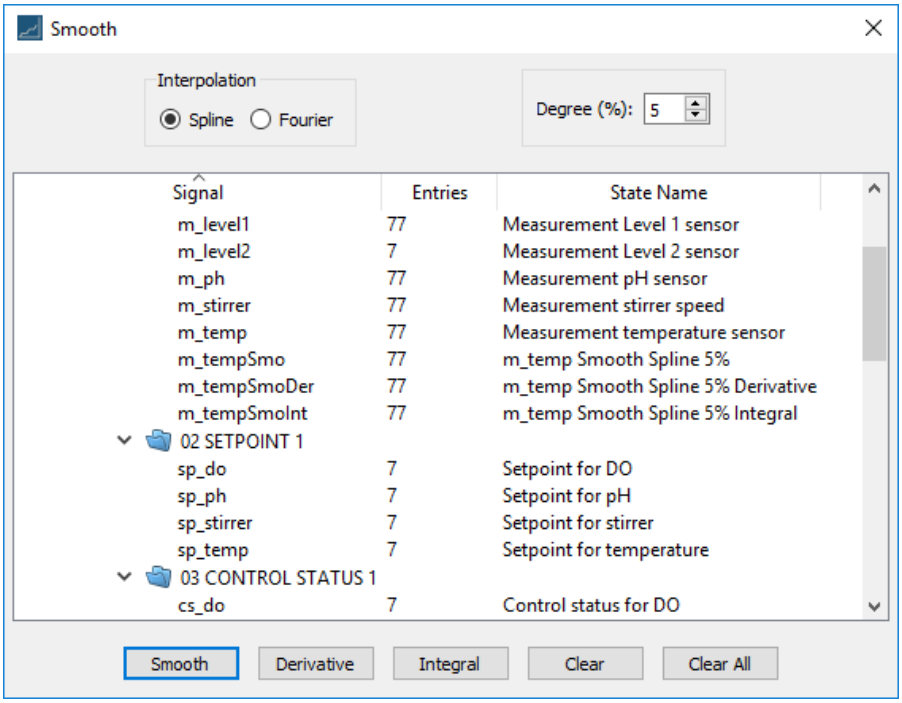
Invoke again the Smooth dialog.

Note that the smoothing operation created the 3 extra variables:

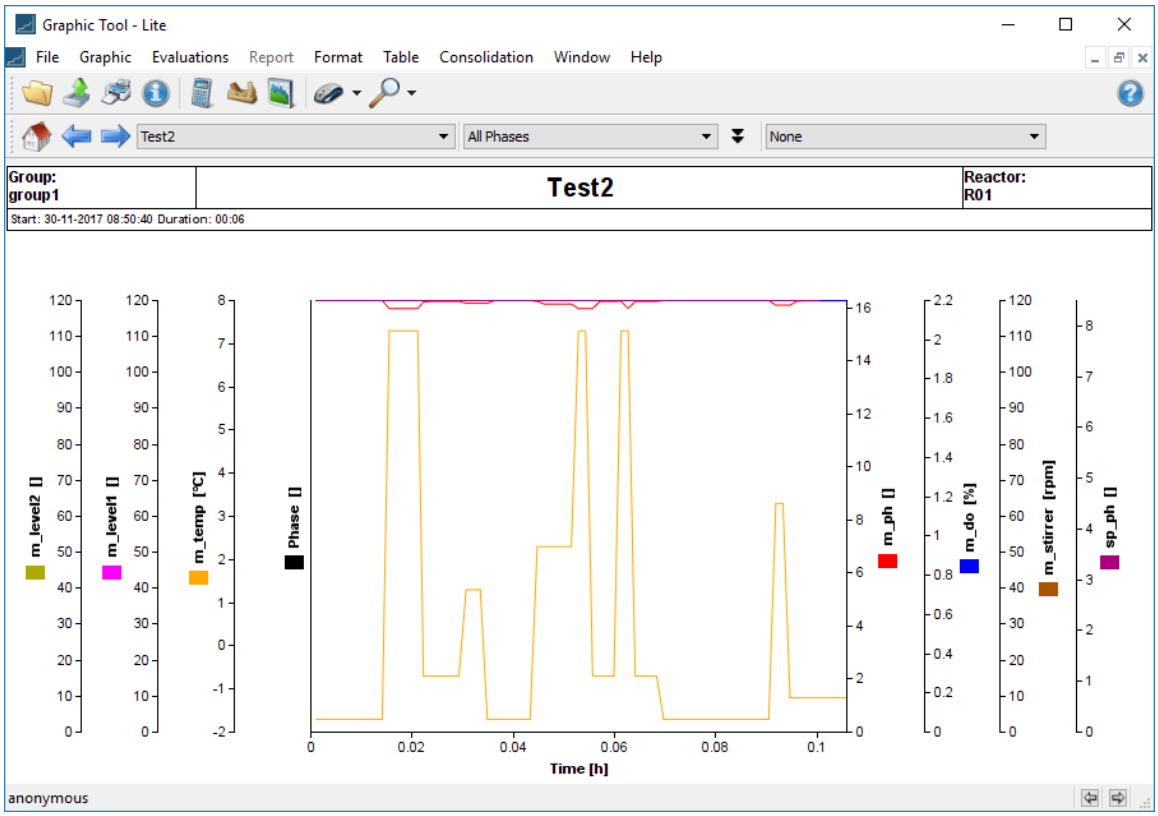
m_tempSmo
m_tempSmoDer
m_tempSmoInt

In the Smooth dialog. m_tempSmo holds the smoothing data. The other two variables hold the derivative and integral of the smoothed data.

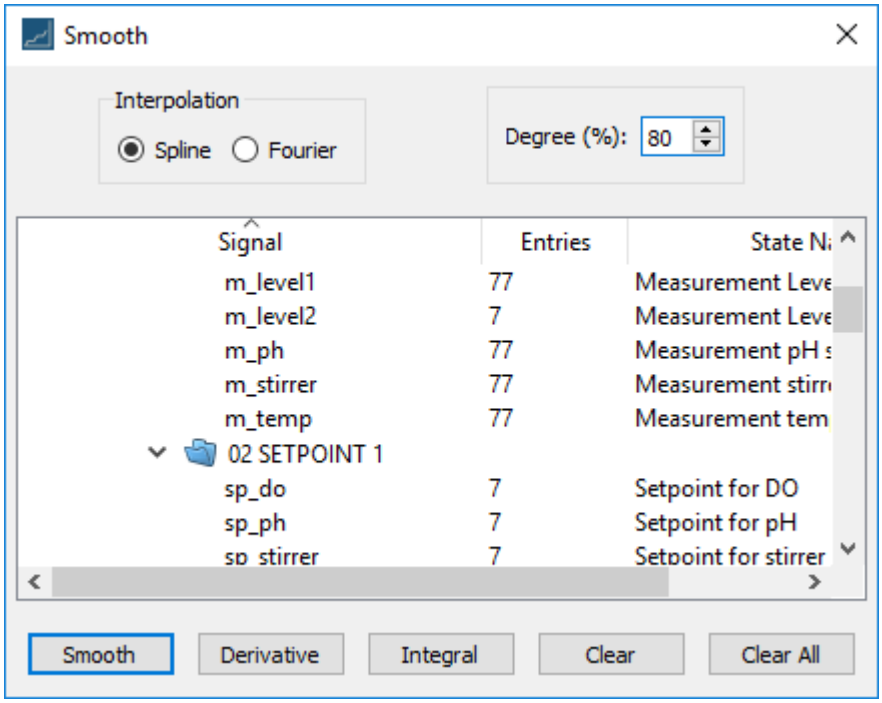
To start over, click on **Clear All** in the dialog at right and



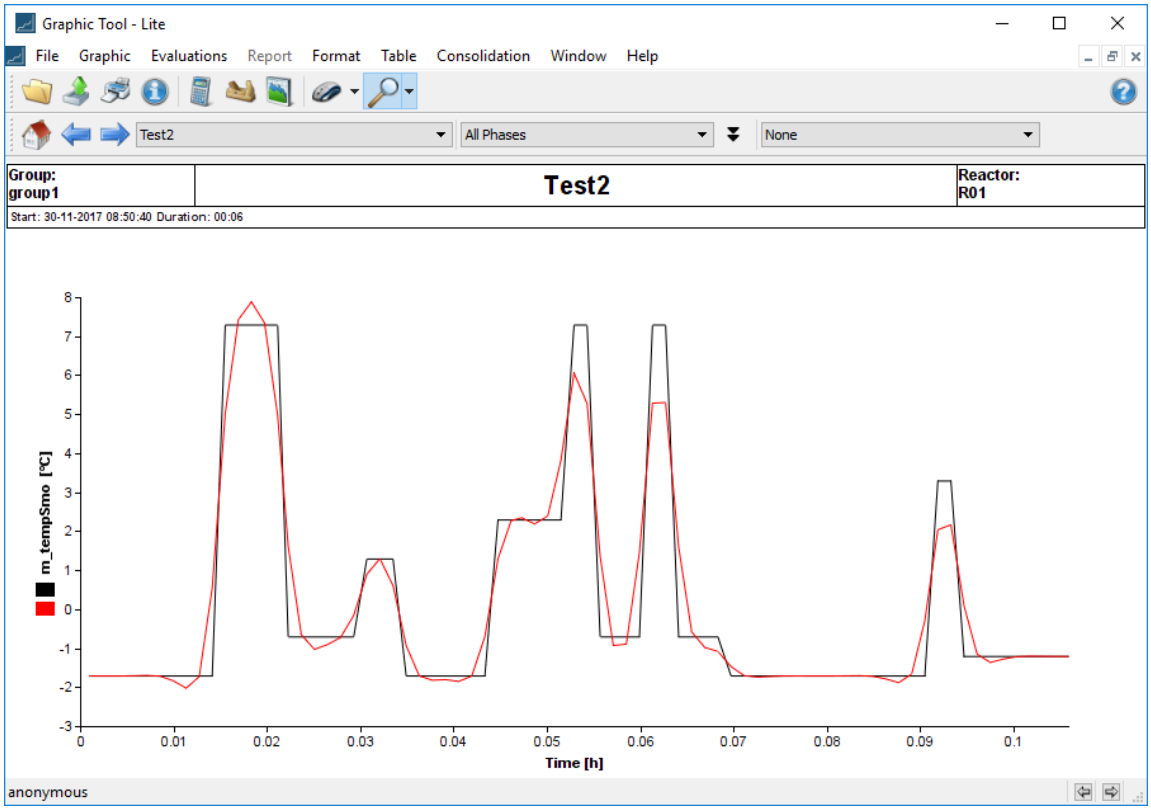
close the plot in the Graphic window via the button. This clears all generated variables and restores the original plot:



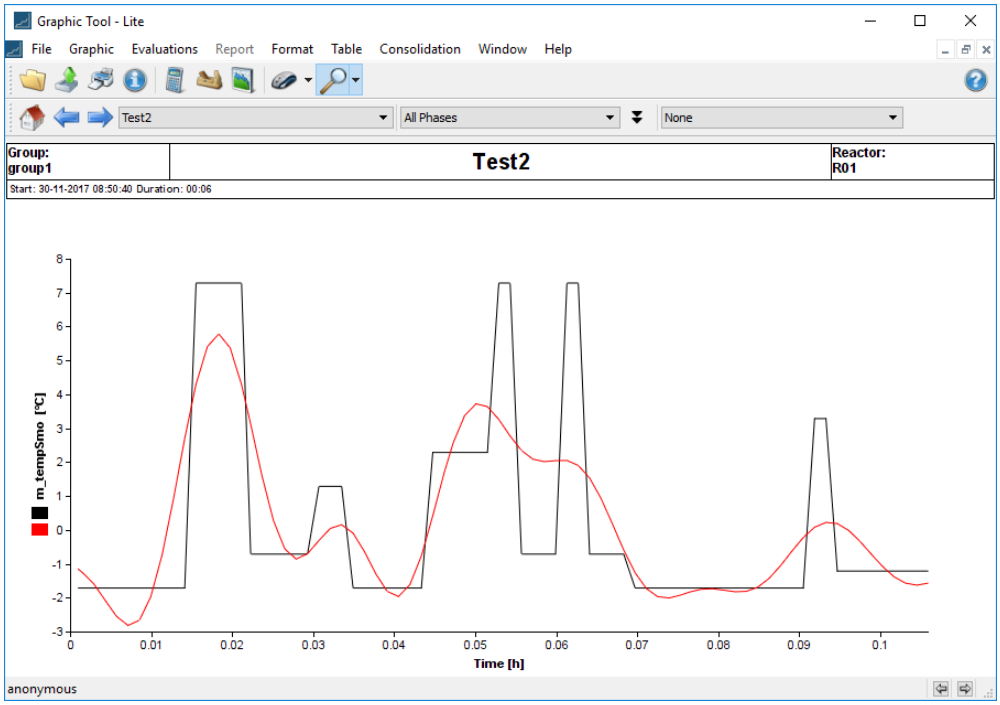
With the **Smooth** dialog still open, now raise the **Degree** value to 80%.



This produces the following result:



This shows somewhat better smoothing. If you now choose **Fourier** smoothing with **Degree** 25, in order to approximate the data with sinusoidal components, the result would be:



which clearly shows the sinusoidal nature of Fourier smoothing.

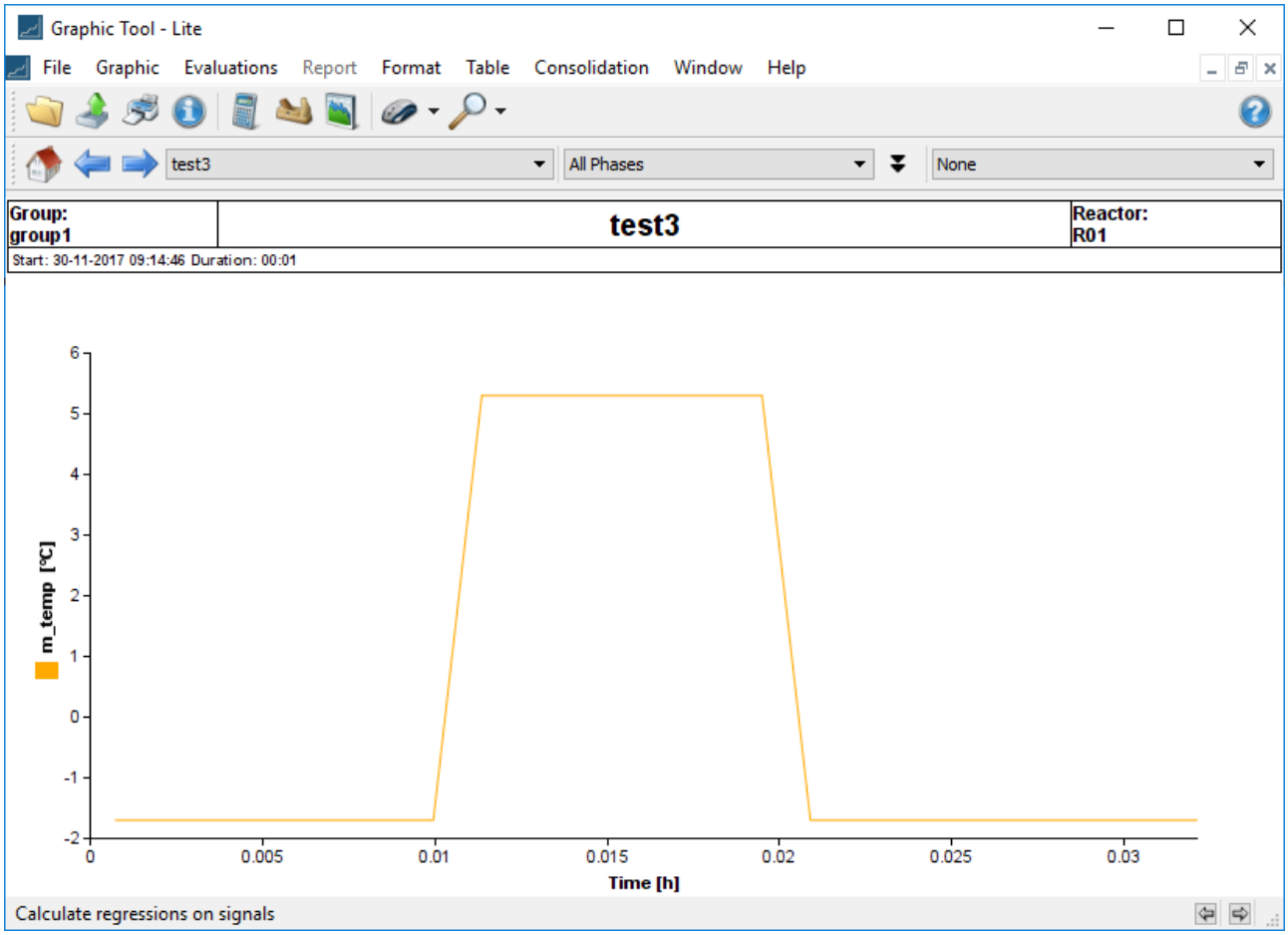
3.5 PERFORMING REGRESSION ANALYSIS

While the previous section illustrated data *smoothing*, this one illustrates data *fitting*. Recall the distinction: *smoothing* abandons an exact data match in order to produce a smooth curve that approximately follows the data; *fitting* tries to find the mathematical expression that most closely matches the data.

To fit data using regression analysis, one tries to determine a mathematical expression that best matches the data. The mathematical expressions used here are:

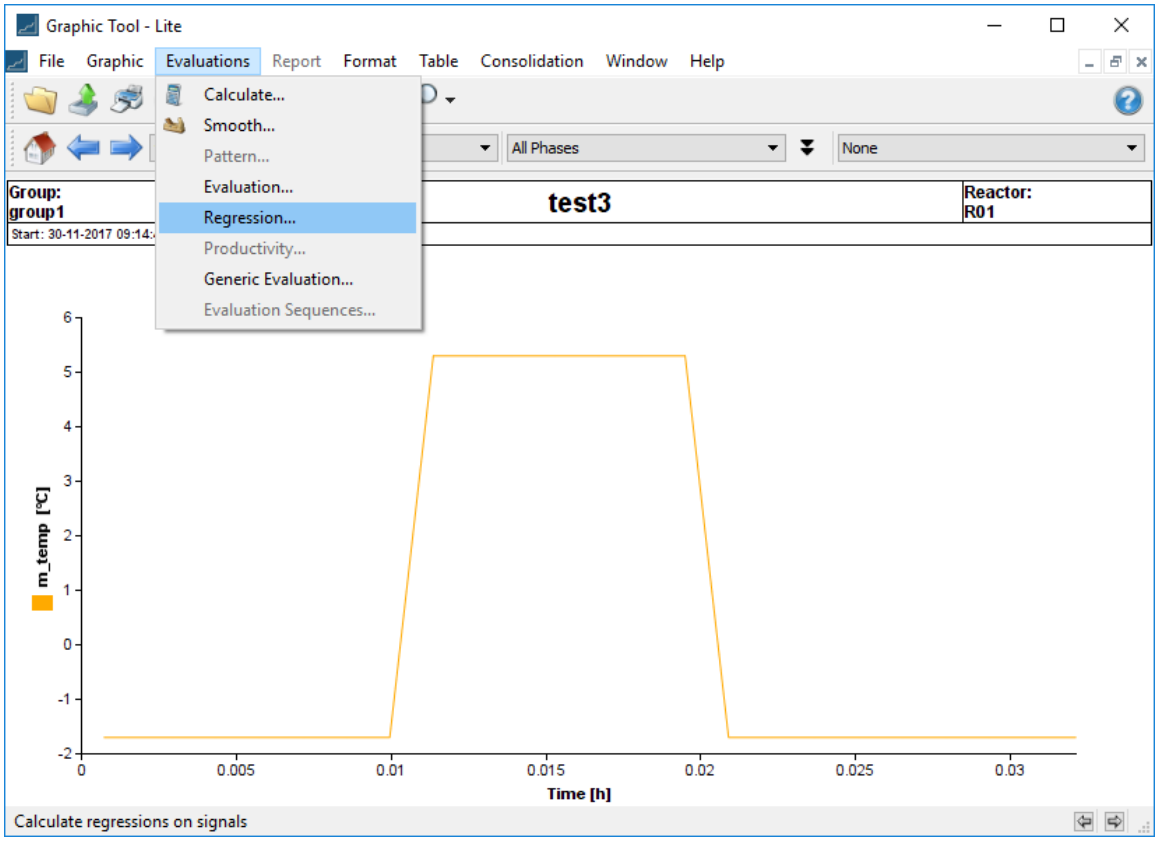
- Linear expressions
- Polynomial expressions
- Exponential expressions
- Logarithmic expressions
- Sigmoidal expressions

We will illustrate the first two, but the general approach applies to the others as well. By following the same procedure as in the previous section once again, start the Graphic Tool and open a data set selected from the collection of all data sets:



This is a fairly simple data set but is useful in illustrating how the regression analysis performs.

Navigate to **Evaluations → Regression:**

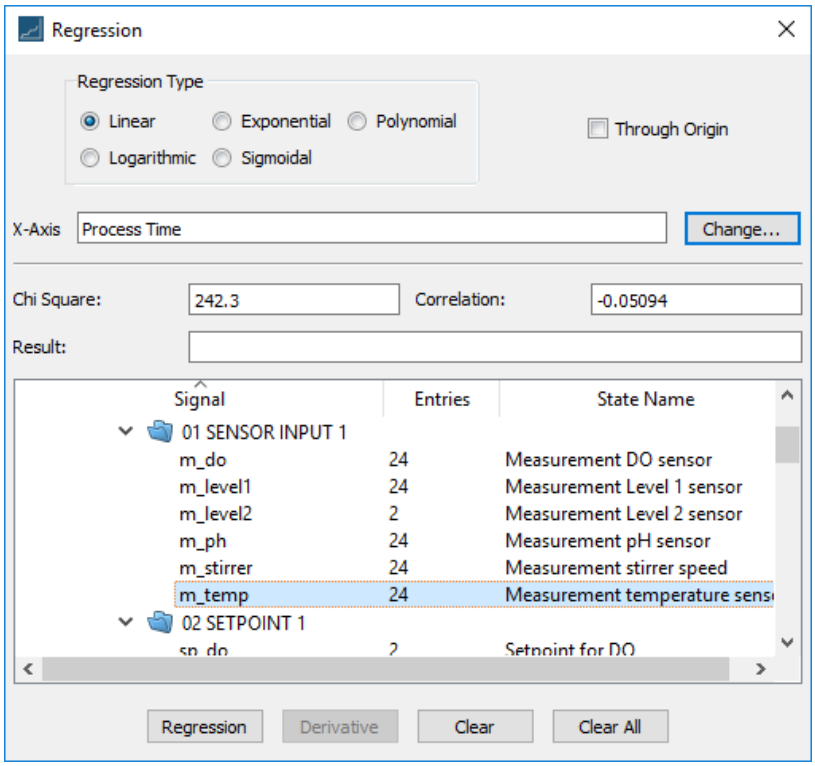


This invokes the following dialog shown at right.

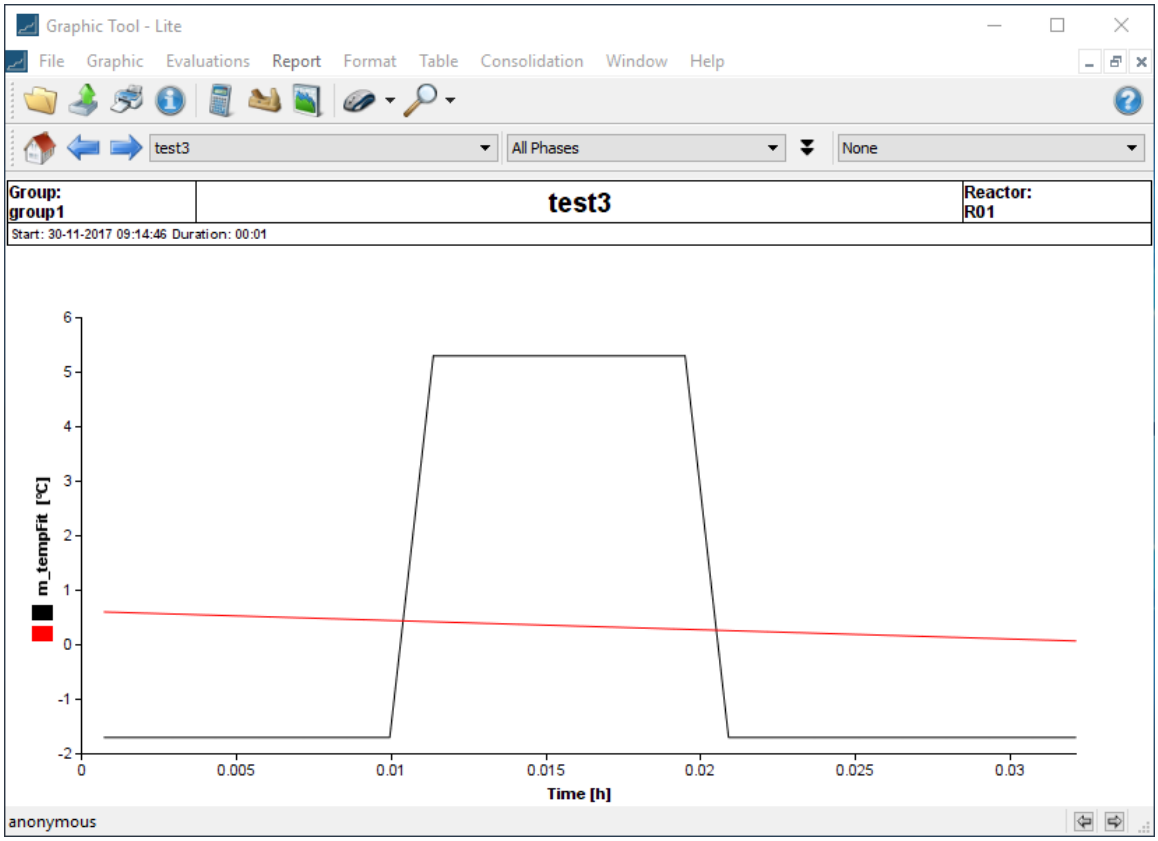
Select the signal **m_temp** and for the **Regression Type**, select:

- Linear

Whether the data is meant to go through the origin is a judgment you have to make based on your knowledge of what the data should be at time zero; here we choose not to, so we leave **Through Origin** unchecked.



The result is as shown below.



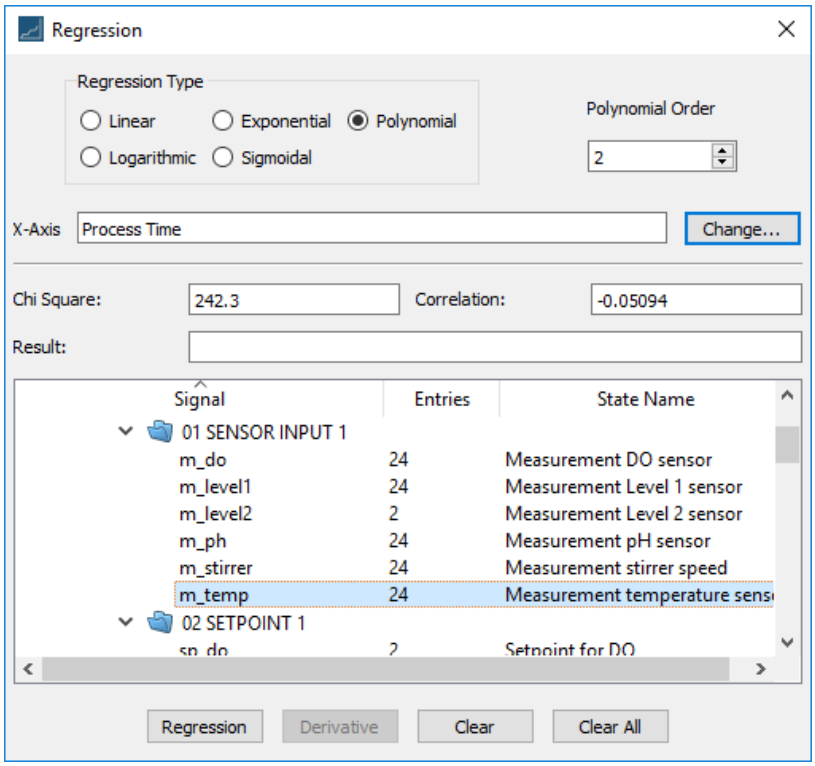
The data has been approximated by the red line shown above.

To start again, click on **Clear All** and close the regression plot using the button

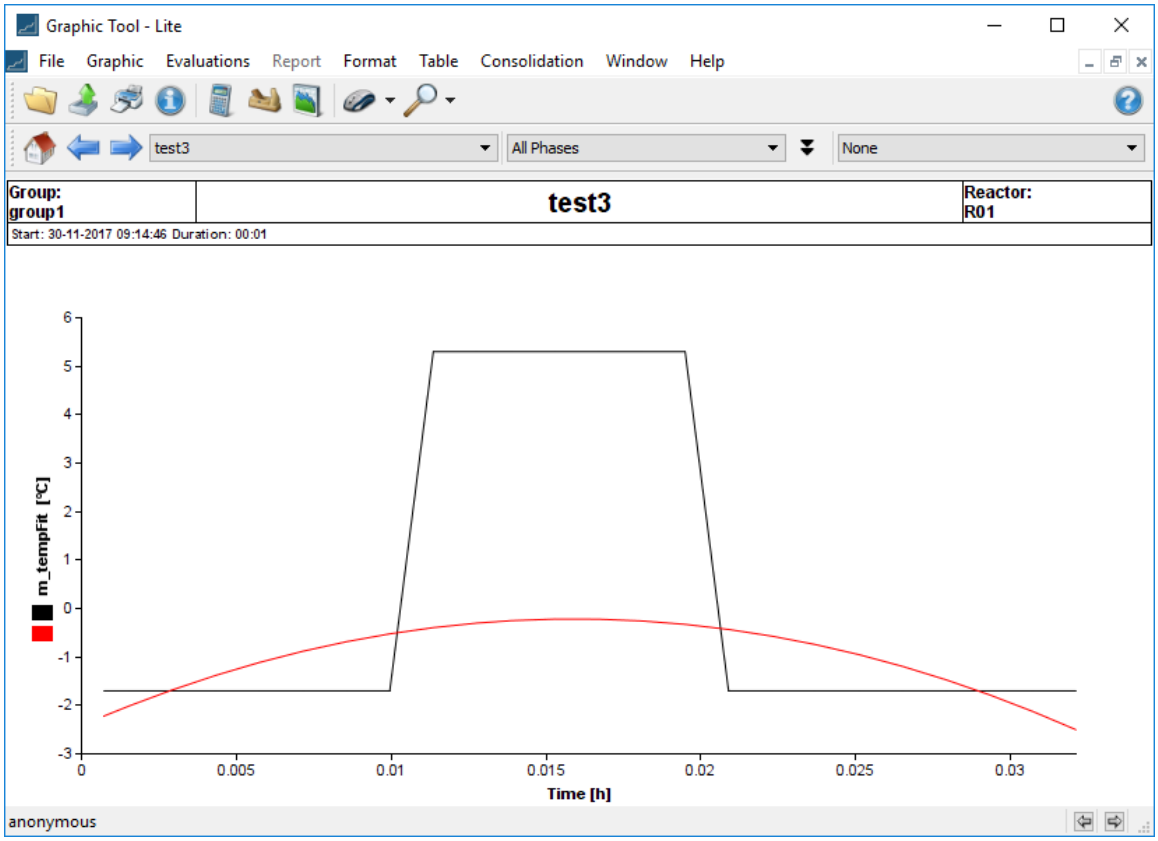
Let us now choose the following **Regression Type**:

- Polynomial
- Polynomial Order 2

Select **m_temp** and click on **Regression**.



The result is as shown below:



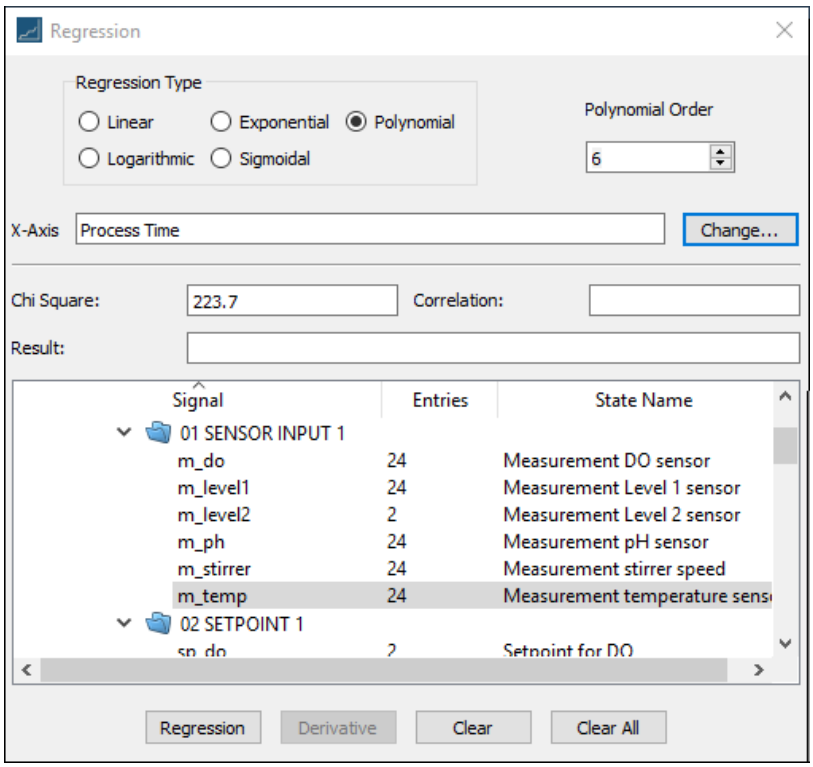
The data has now been approximated by a quadratic.

To start again, click on **Clear All** and close the regression plot using the button

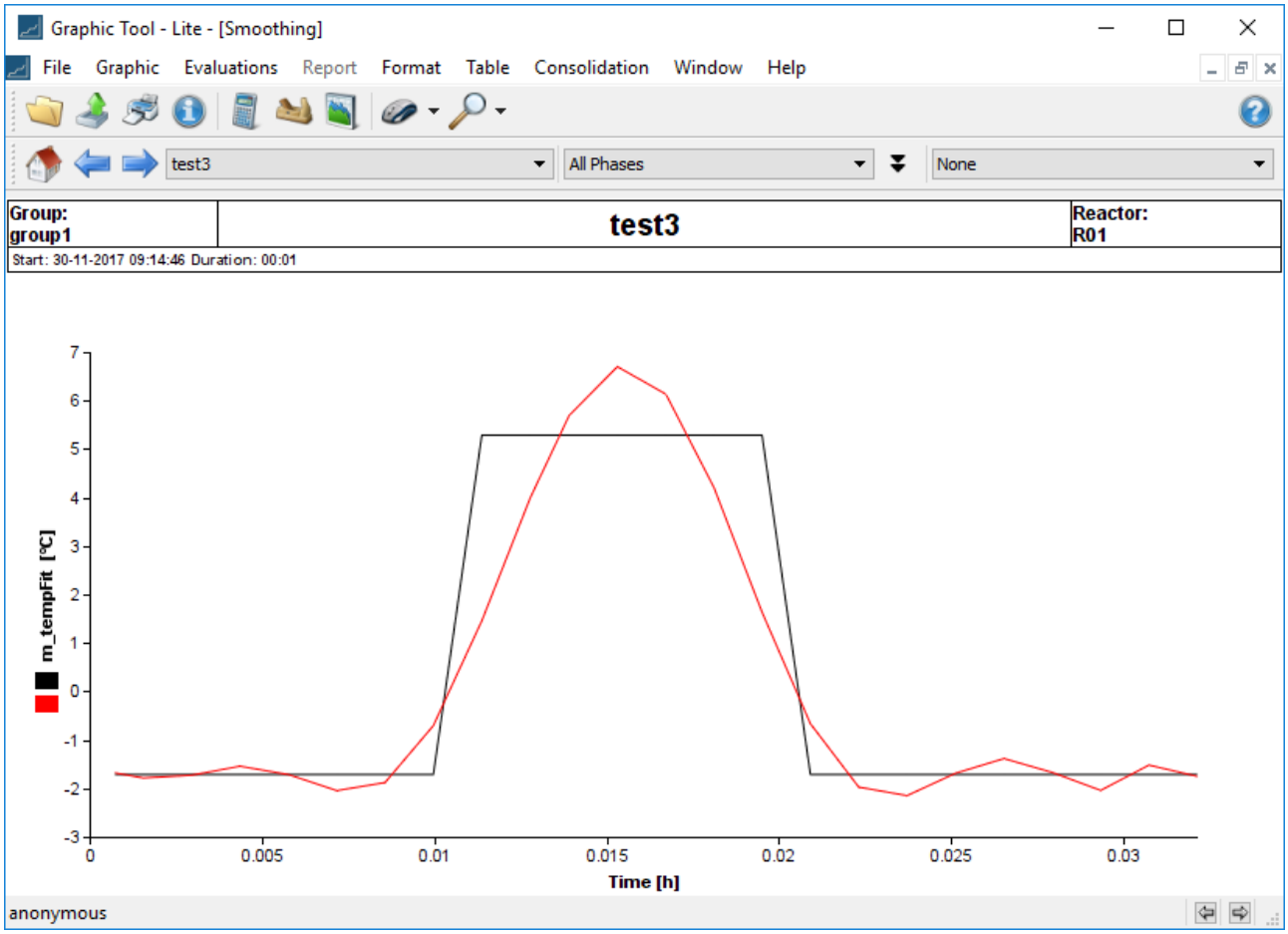
Let us now choose the following **Regression Type**:

- Polynomial
- Polynomial Order 6

Again, select **m_temp** and click on **Regression**.



The result is shown below:

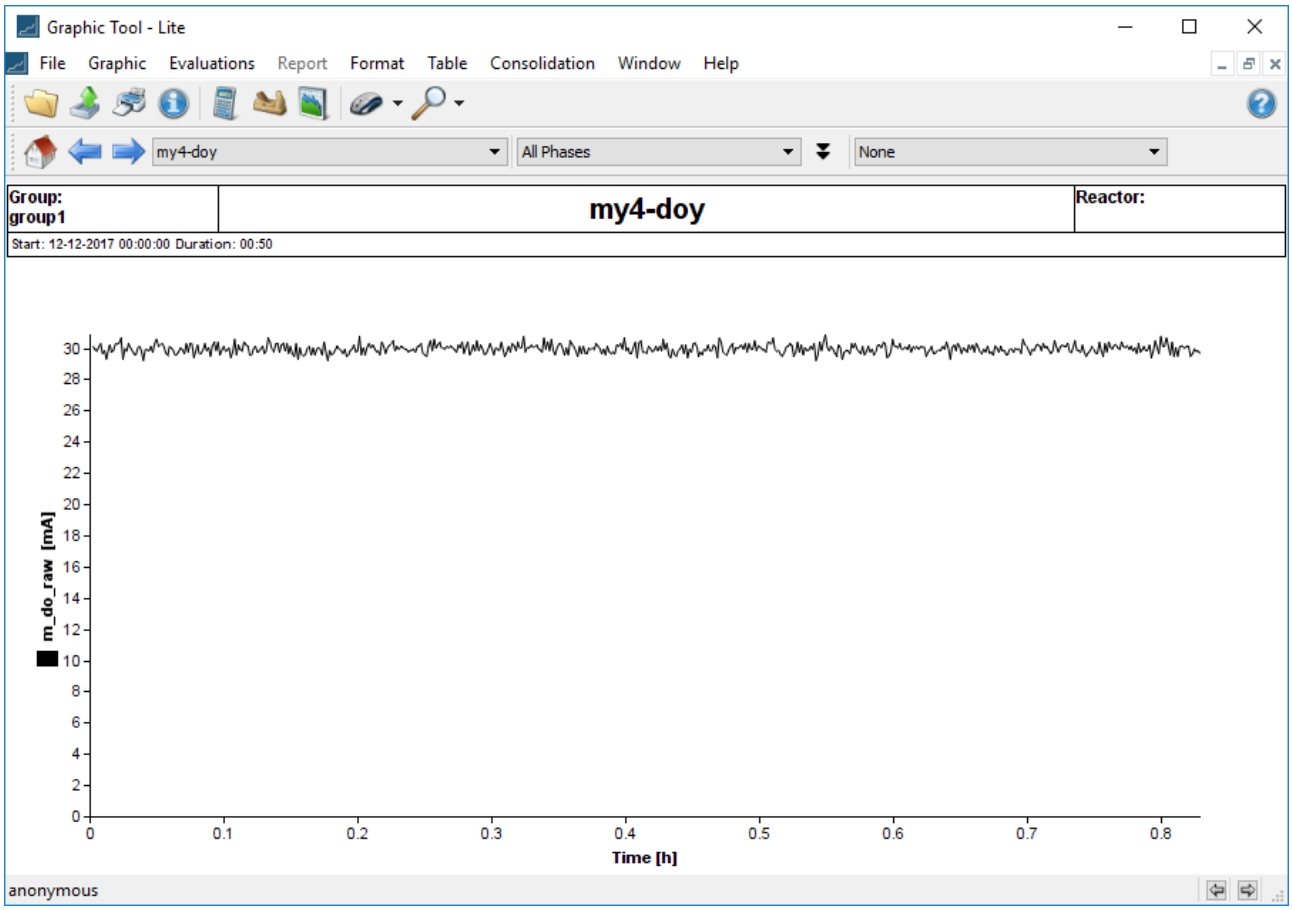


And one can see that for this data, a polynomial of sufficiently high degree can fit the data fairly closely.

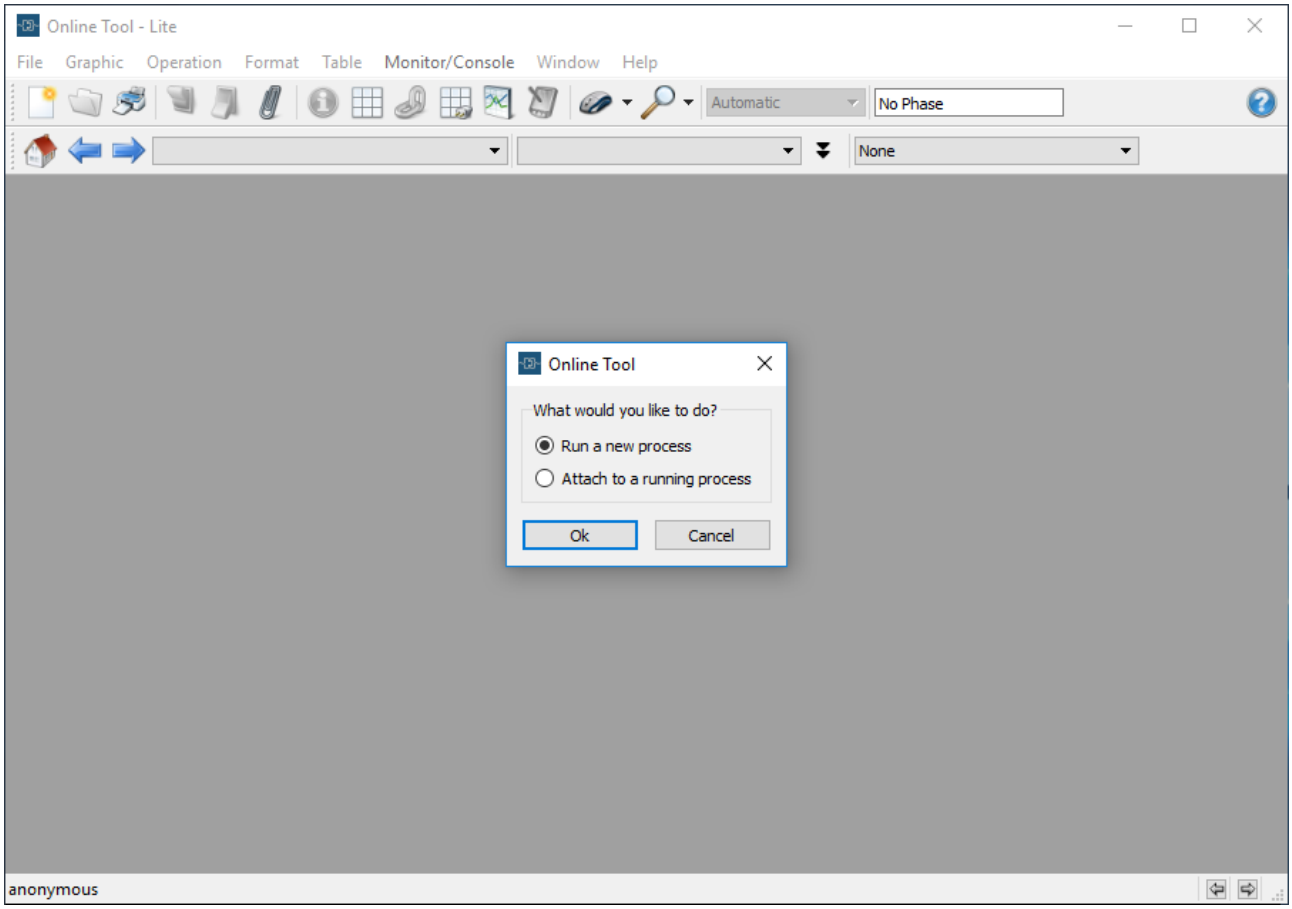
3.6 MAKING GRAPH OVERLAYS

When running an experiment one can overlay the data from past and stored experiments over the data of the currently running experiment. This is visually extremely useful for comparing trends.

To do this in Lucullus, one first starts an experiment and then selects one or more prior experiments to overlay. So start by examining your stored experiments to select the ones to overlay. You do this by invoking the **Graphic Tool** and then navigating to **File → Open**, and finally selecting an experiment. If it is not the desired one, you repeat these steps until you have found the desired experiment or experiments. For this example, we choose the following experiment:

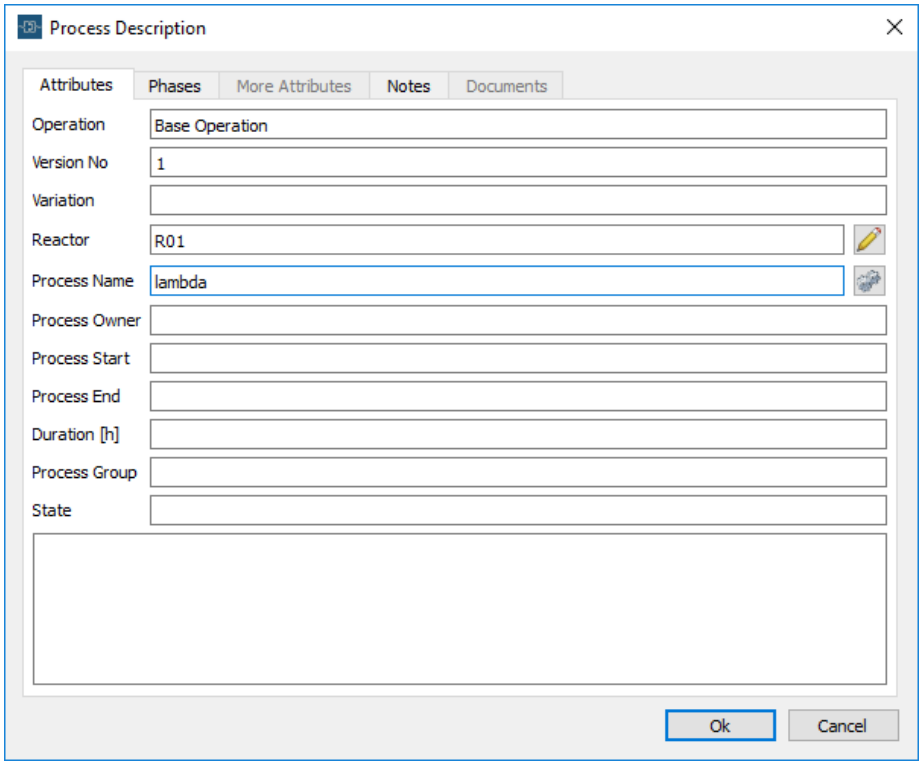


Then you start a run. We will run through the steps of starting a process again, as a refresher. Start the **Online Tool** and you get the following:

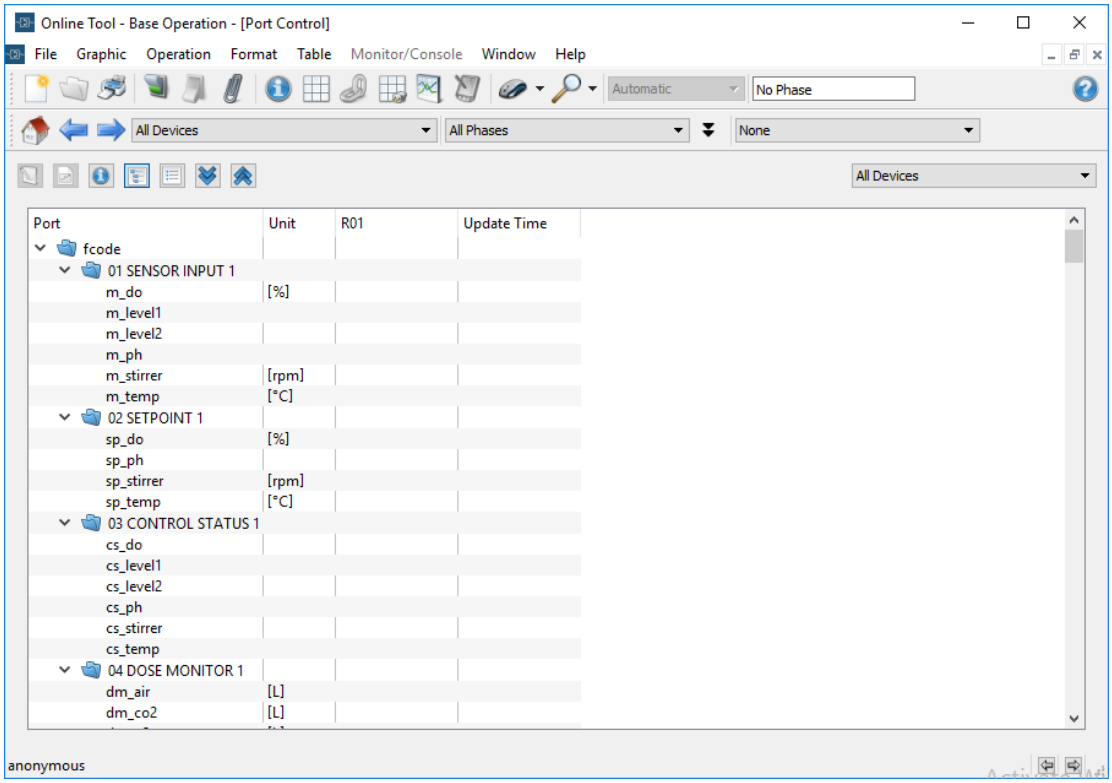


Select **Run a new process.**

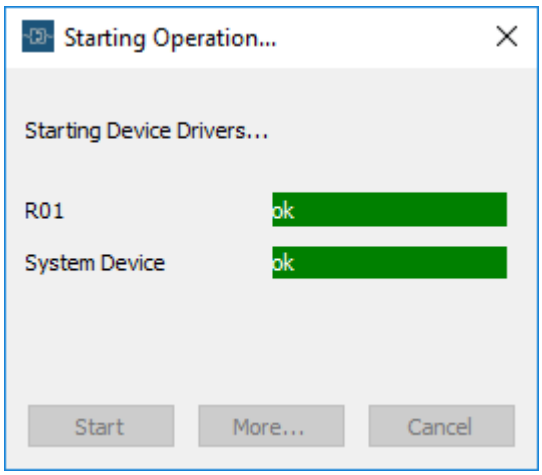
This will invoke the following screen:



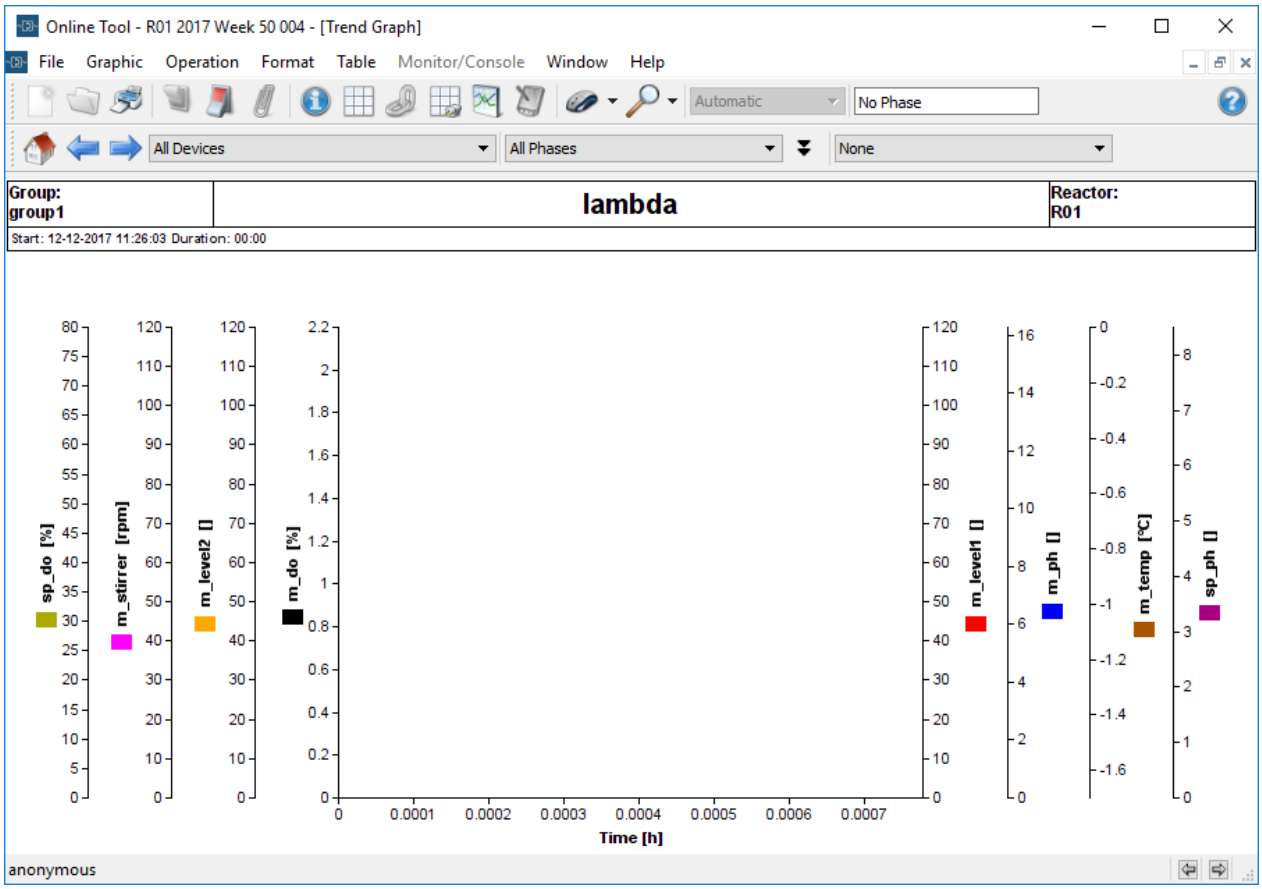
Enter the Operation and the process name (the minimum required entries) and click **OK** for the port display:



Assuming that the port configuration is ok, we proceed to start the experiment using the **Start** button:

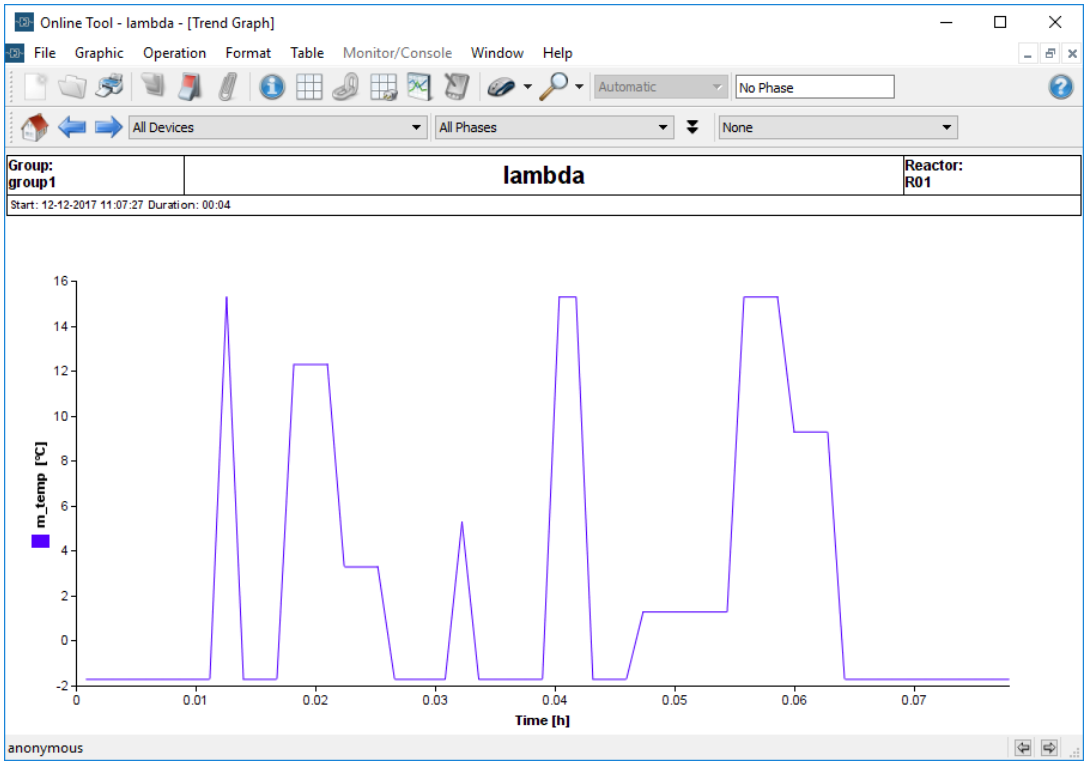


The **Starting Operation** popup automatically disappears and the plot after some time looks as follows:

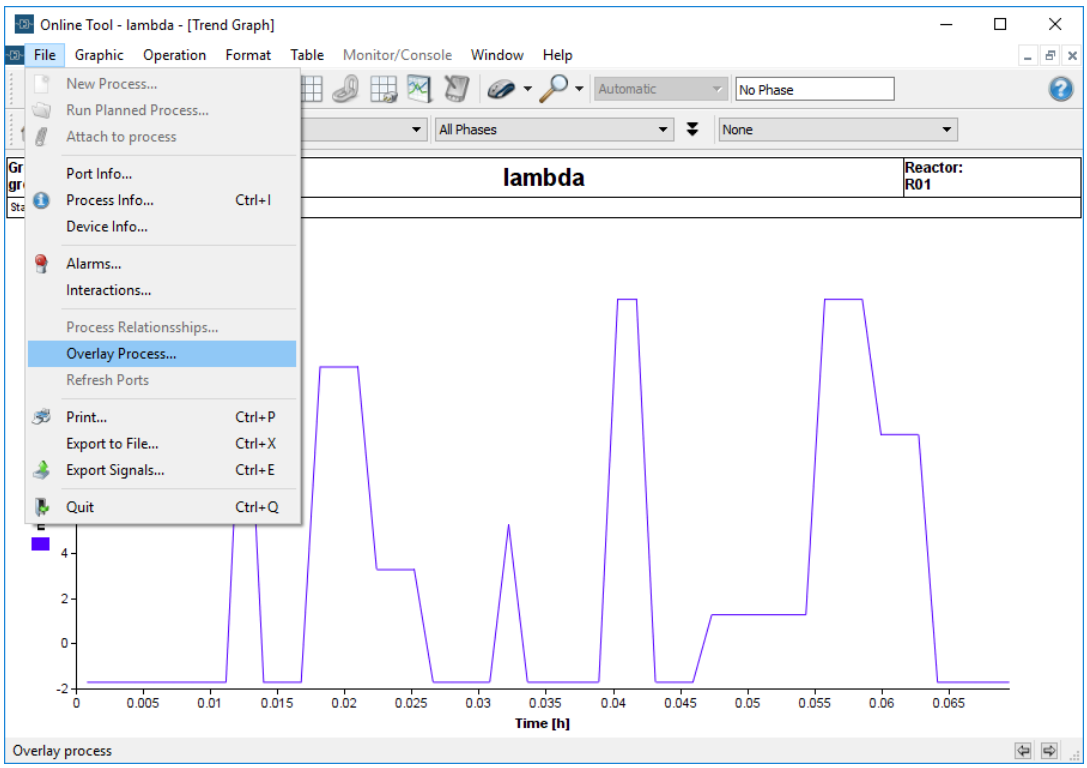


Since only m_temp is measured, we can remove all the other y-axes via **Graphic → data/axes**, or by loading a stored format.

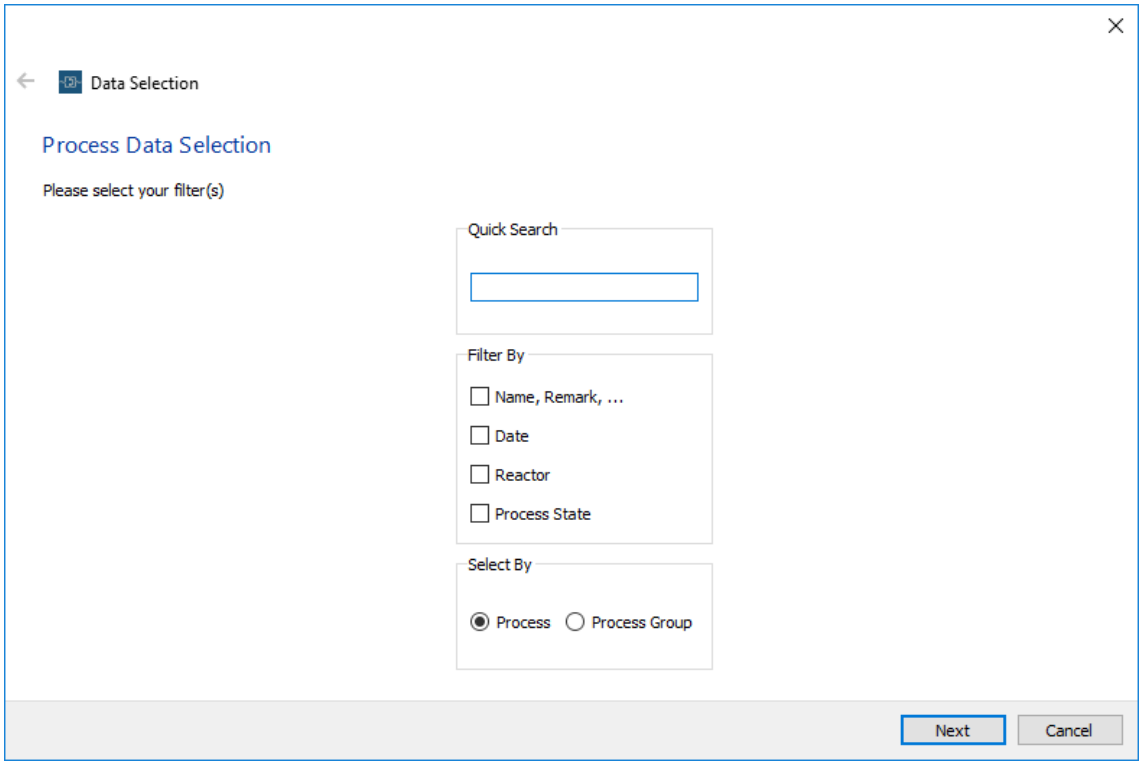
The result is a temperature plot for the current experiment:



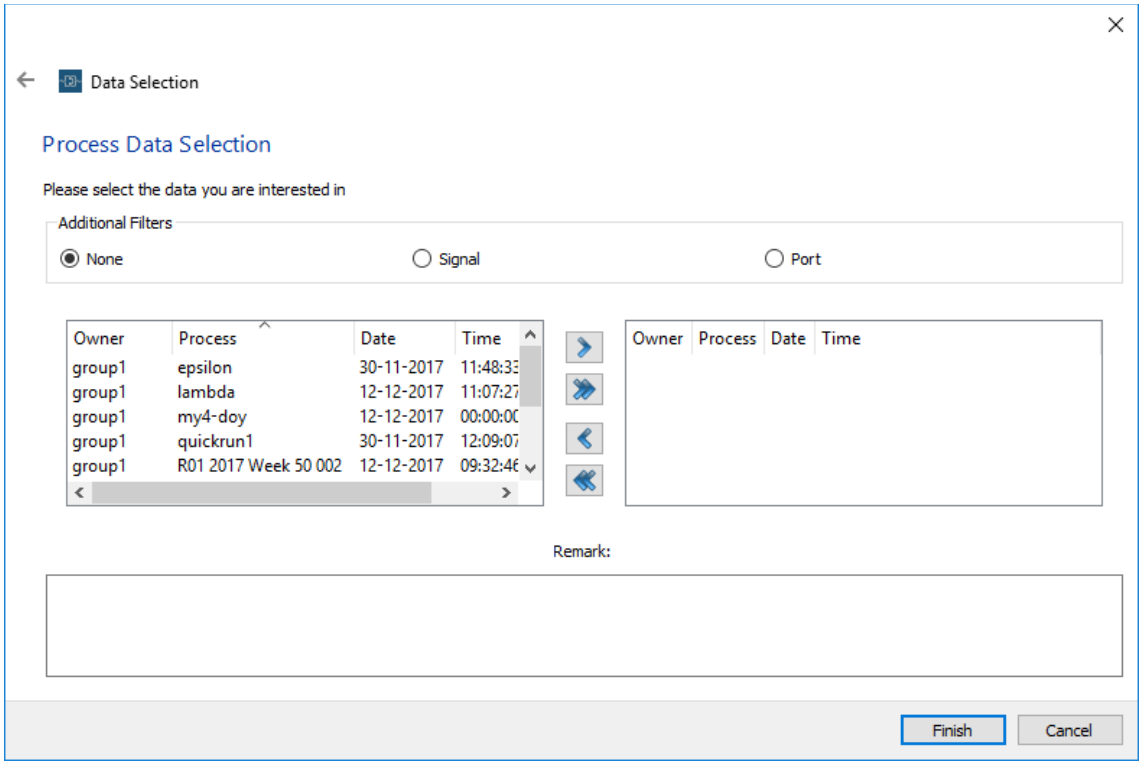
Now we can add an overlay via **File → Overlay Process**:



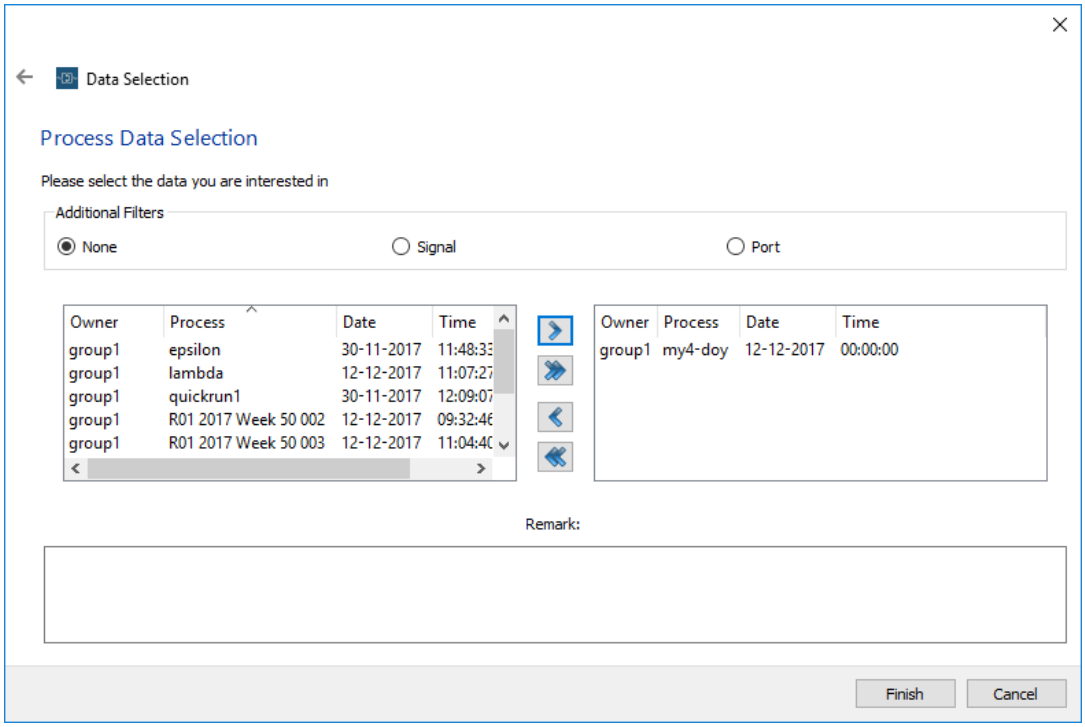
This invokes the screens where we can select from all prior data runs:



Leave all entries blank to select everything and click on **Next**:

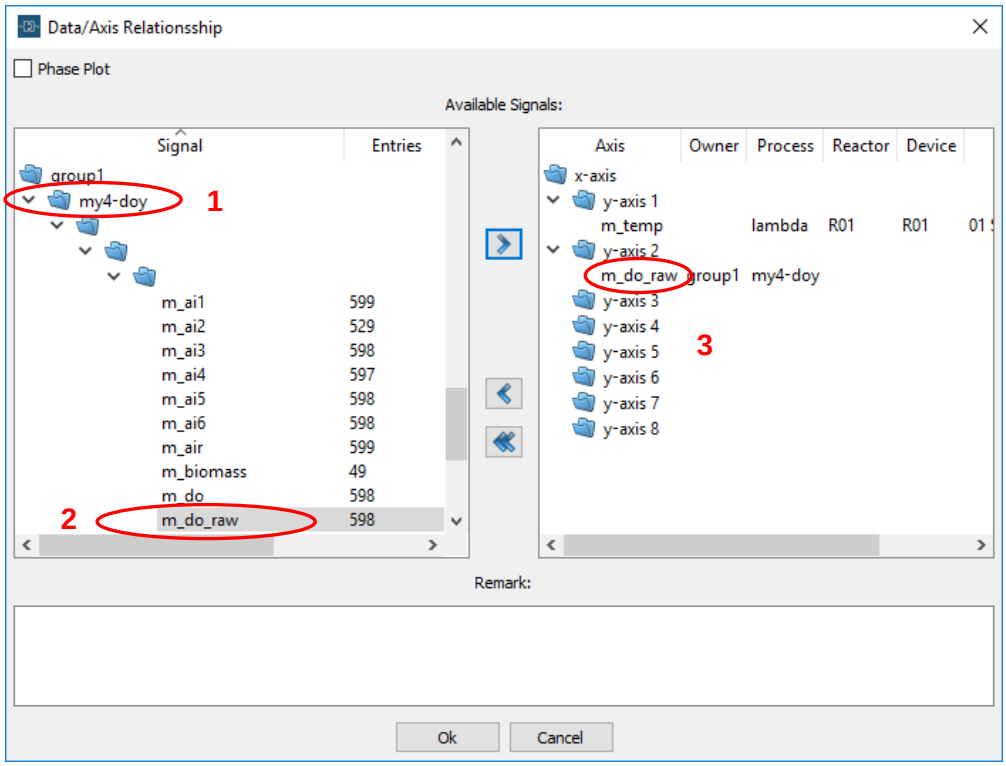


We scroll to the desired entry and select it, then click on '>' to move it to the right window:

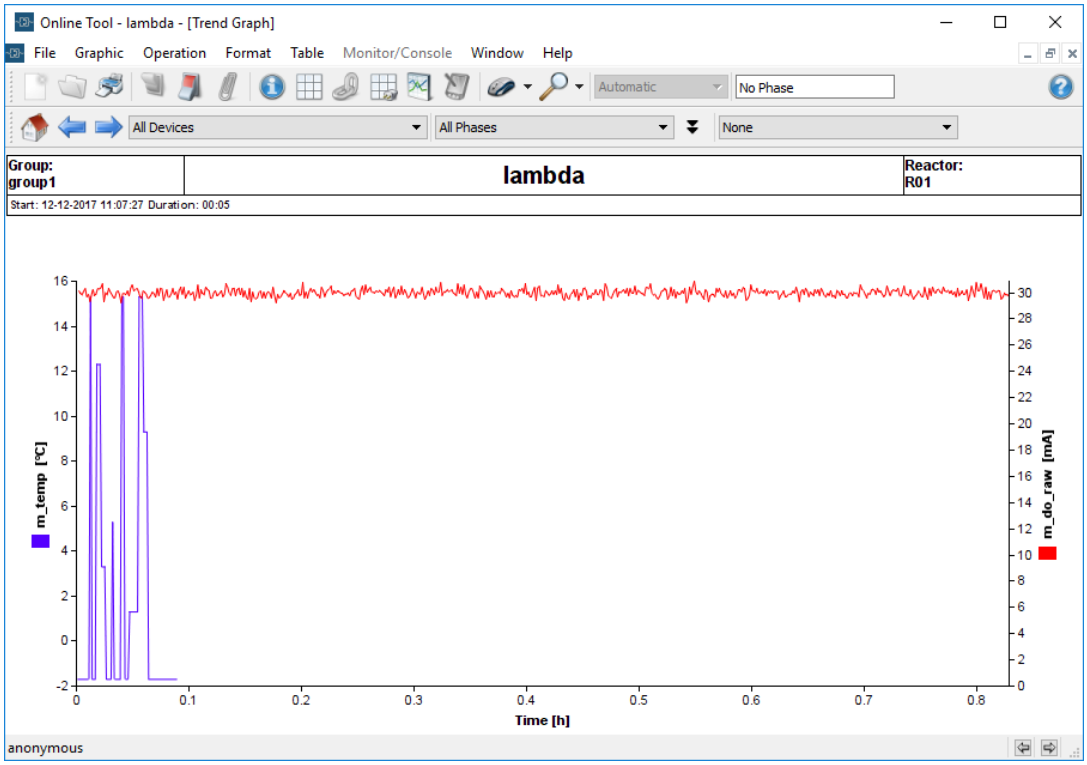


After clicking on **Finish**, we will probably not see the trend of this previous run yet; we still have to specify which of its parameters are to be plotted onto which axis. Navigate to **Graphic → data/axes**. Then scroll down in the left pane until you see the entry corresponding to the process you wish to overlay (alpha1).

1. Locate the desired process in the left pane
2. Locate the process value to overlay
3. Select the process value (m_do_raw here) and click on '>' to transfer it to the next available axis on the right pane

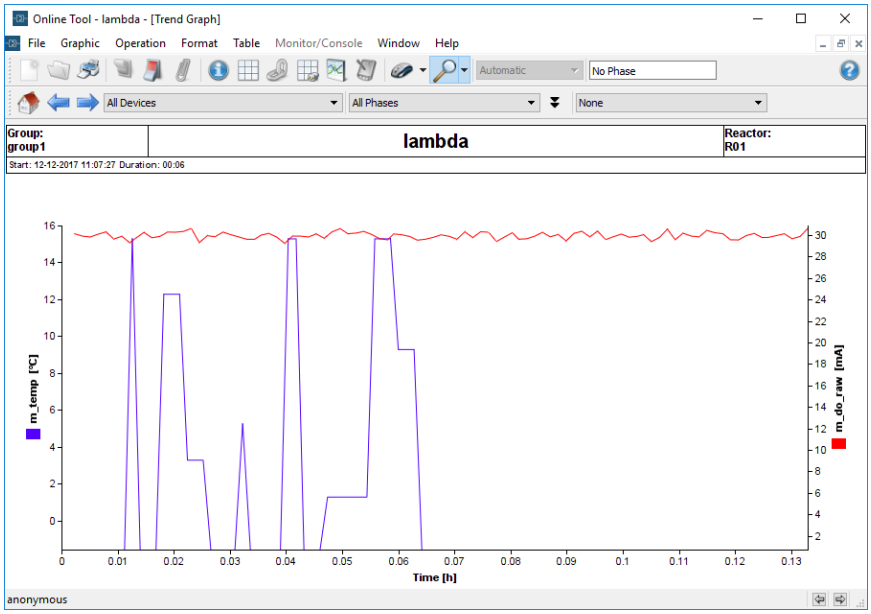


Press on **OK** and the overlay is displayed:



And here we recognize the trend plot for **my4-doy** in red that was shown at the start of this section, overlaid onto the trend in blue for the current run. The trend in blue is compressed as a result of horizontal rescaling in order to fit the larger number of data points in **my4-doy**.

You can rescale the axes to show the blue trend in more detail, or you could use the magnifying glass to zoom in on it. If you do this last, the result looks like the following:



3.7 EXPORTING DATA

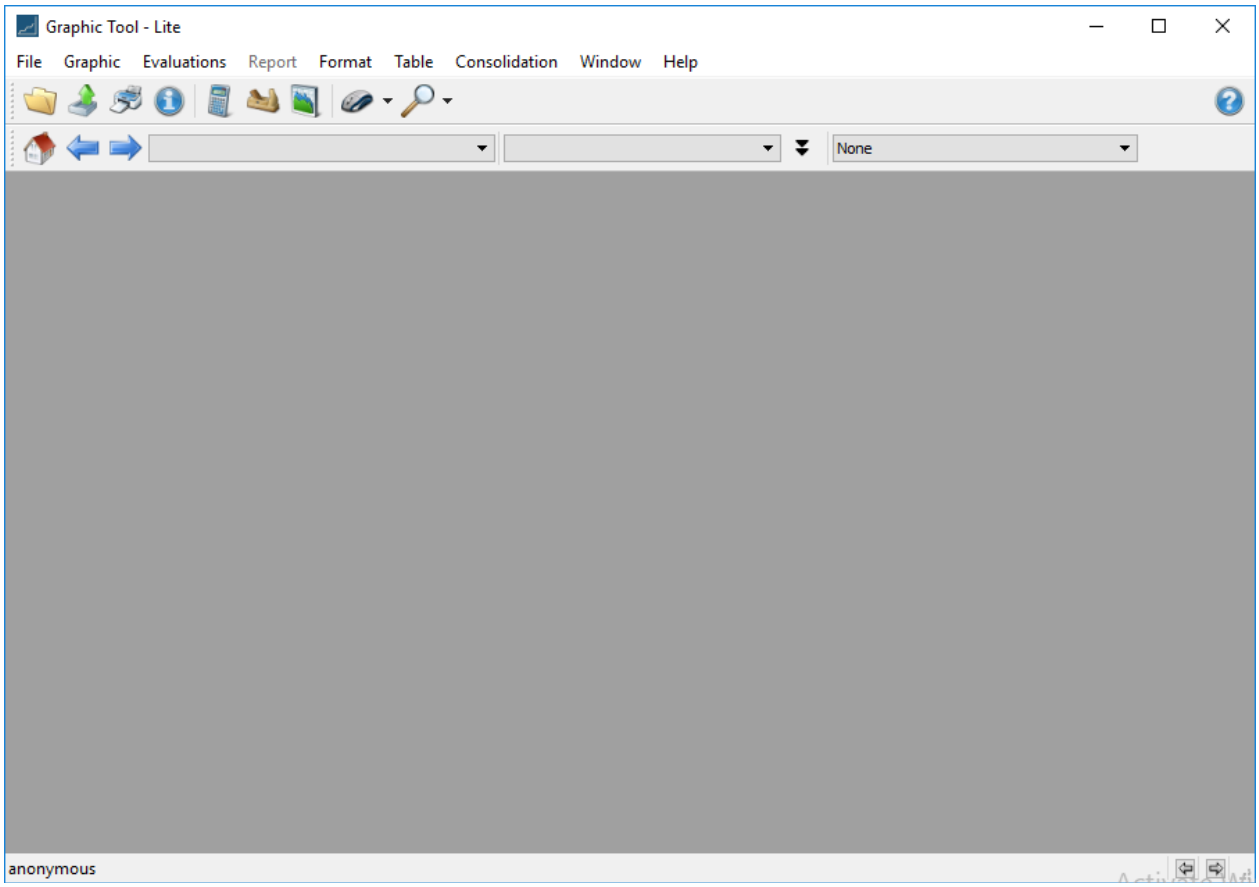
It is very convenient to be able to export data in order to further process it outside of the Lucullus environment.

Exporting from previous runs

When exporting data from previousprevious runs, the main steps consist of loading the desired data set, then selecting the signals from this data set that we wish to export. The steps in more detail are:

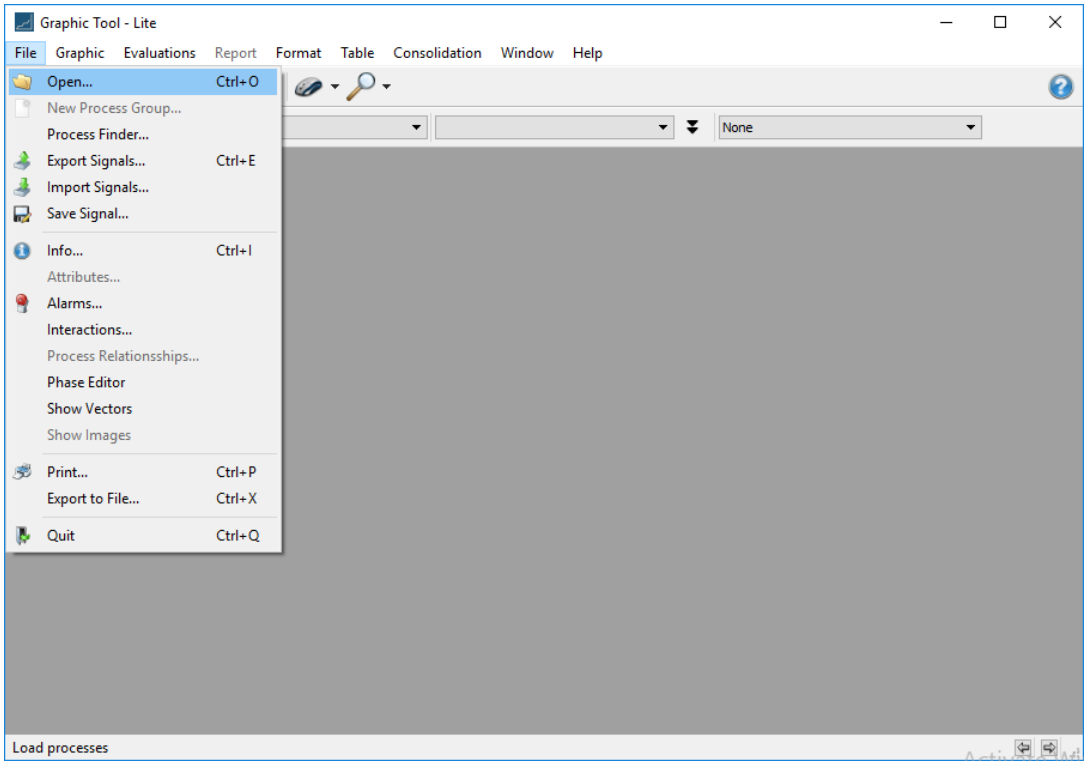
1. Start the **Graphic Tool**
2. Load the stored data set via **File → Open**
3. Select the desired data set
4. Start to export via File → **Export Signals**
5. Select the desired signal
6. Select the phases (select **all** if in doubt)
7. Select the mode for time intervals and interpolation options
8. Provide target folder and file name
9. Complete the export

We first start the **Graphic Tool**:

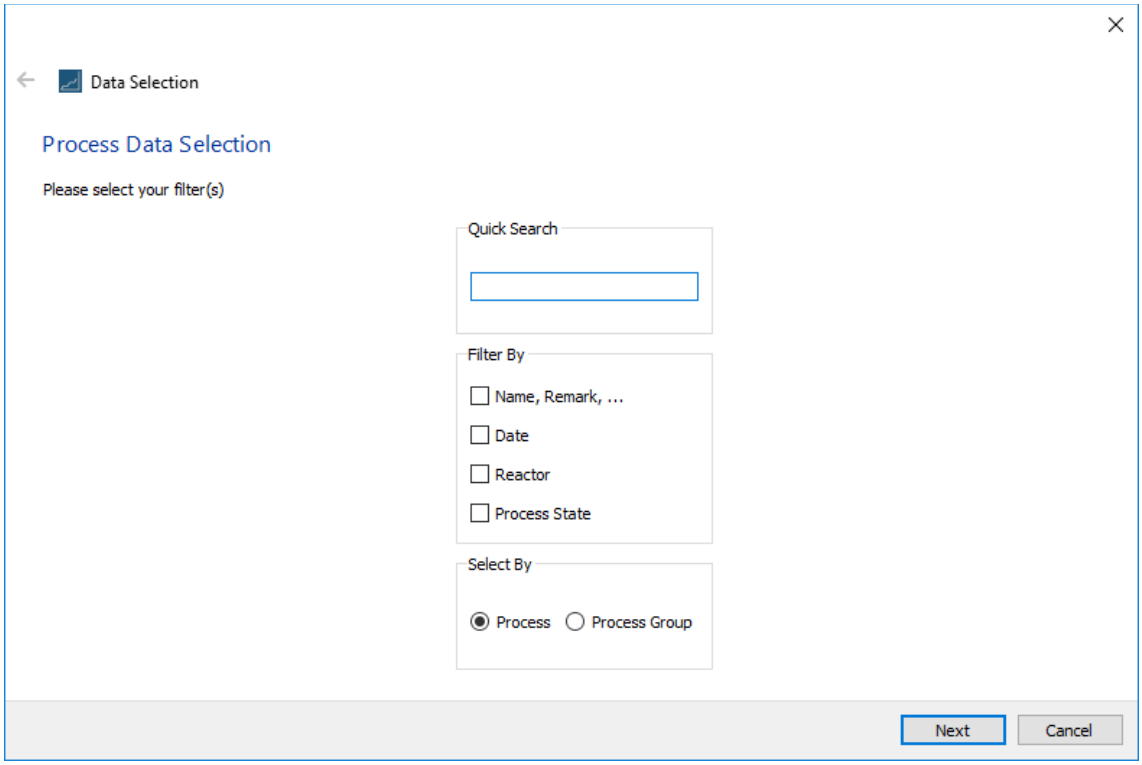


Then we follow the same steps as in previous sections to load the desired data.

Select **Open**:

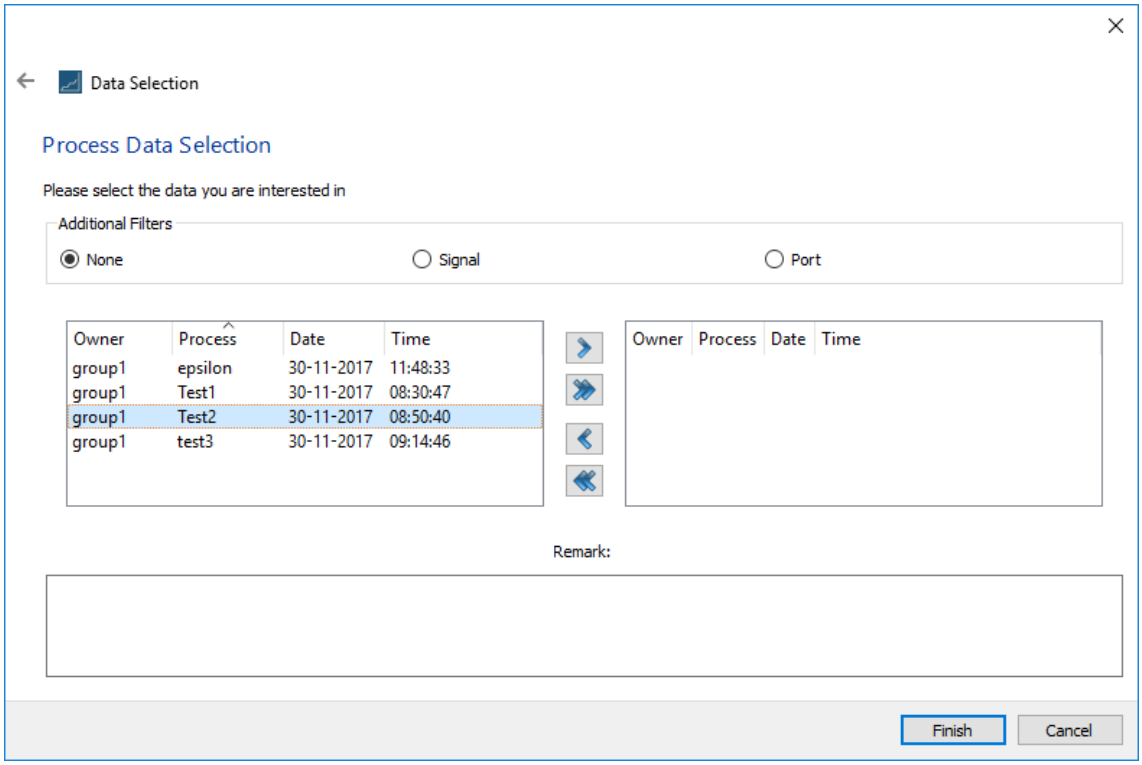


The Process Data Selection window appears:

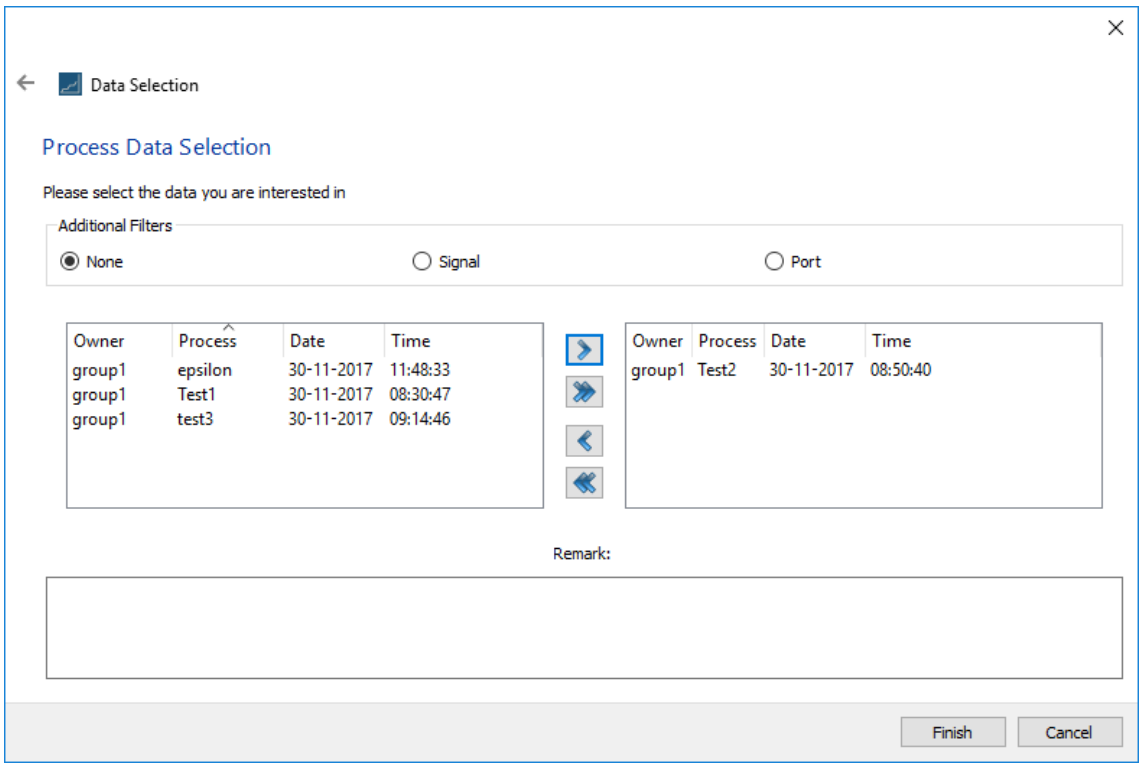


Adjust the filter criteria, or leave everything blank to select all data sets.

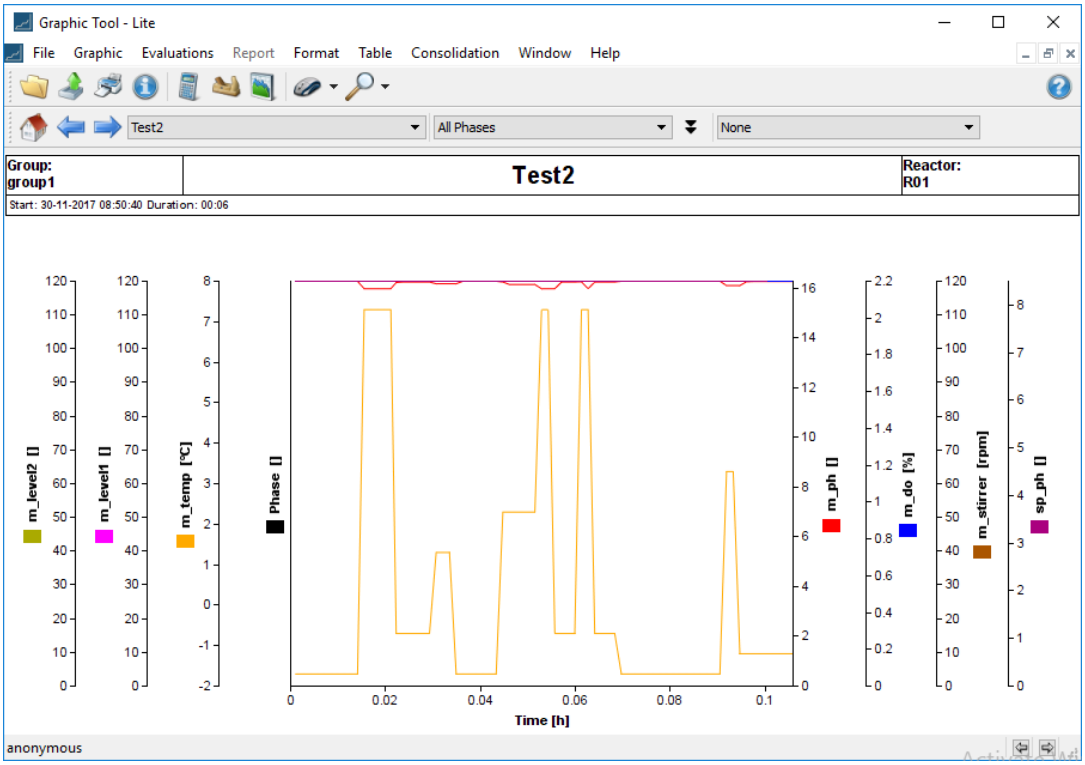
Scroll the data sets until the desired one is found. Highlight it:



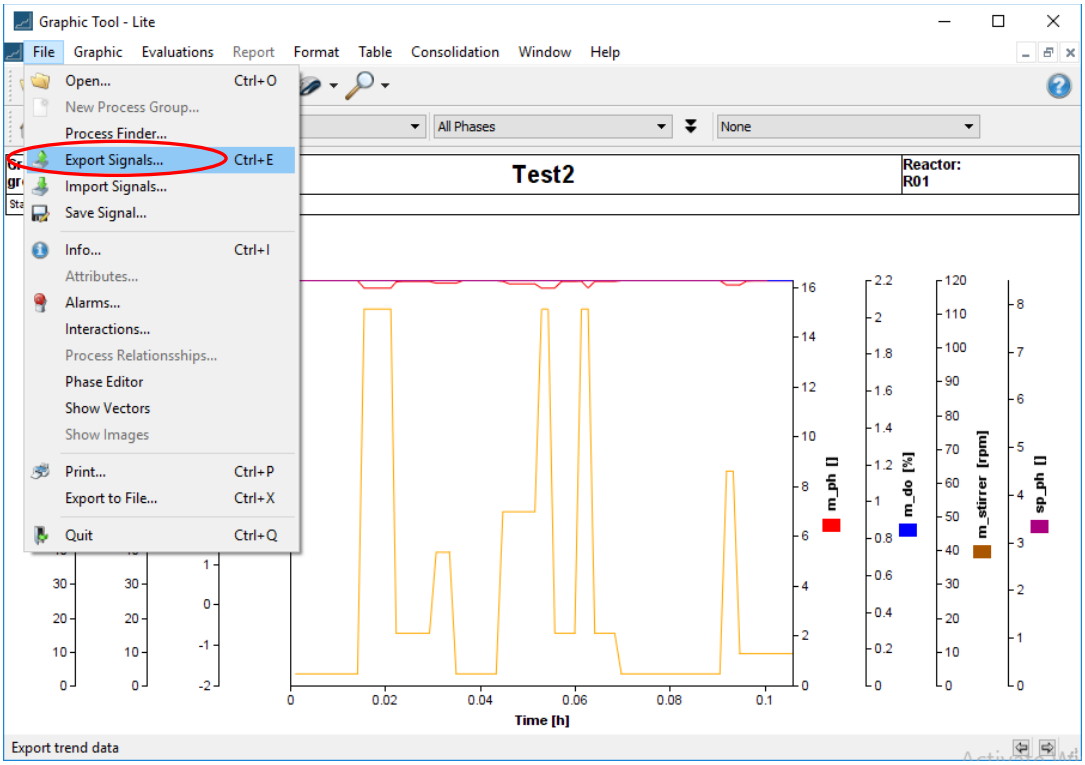
Select the desired data set, click on '>' to transfer it to the right pane and click on **Finish**:



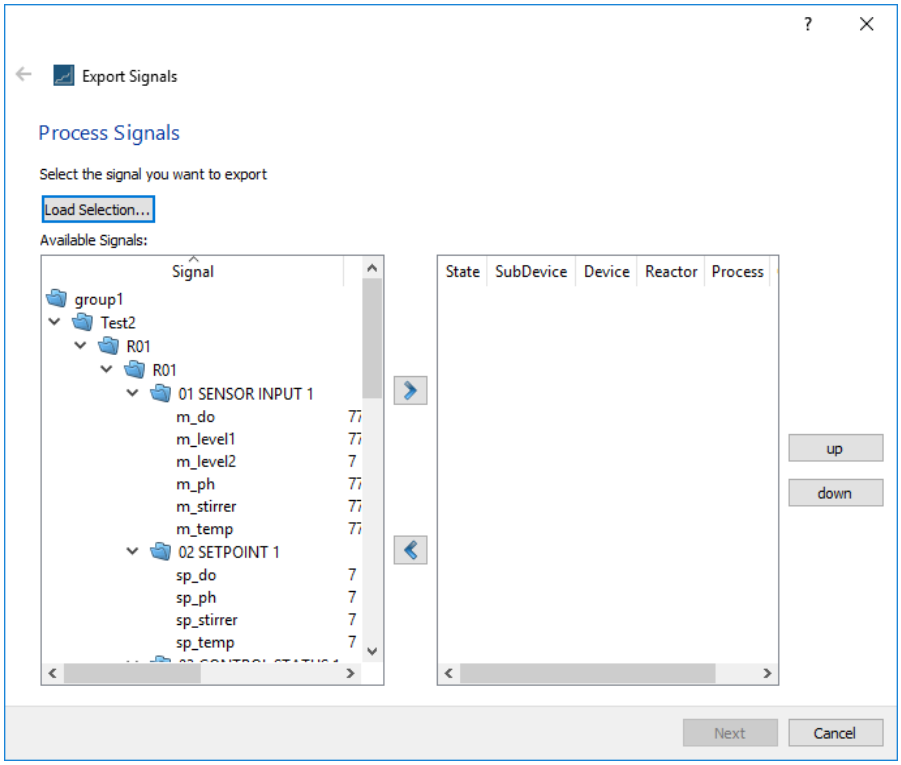
The desired data set (**test2** here) is displayed:



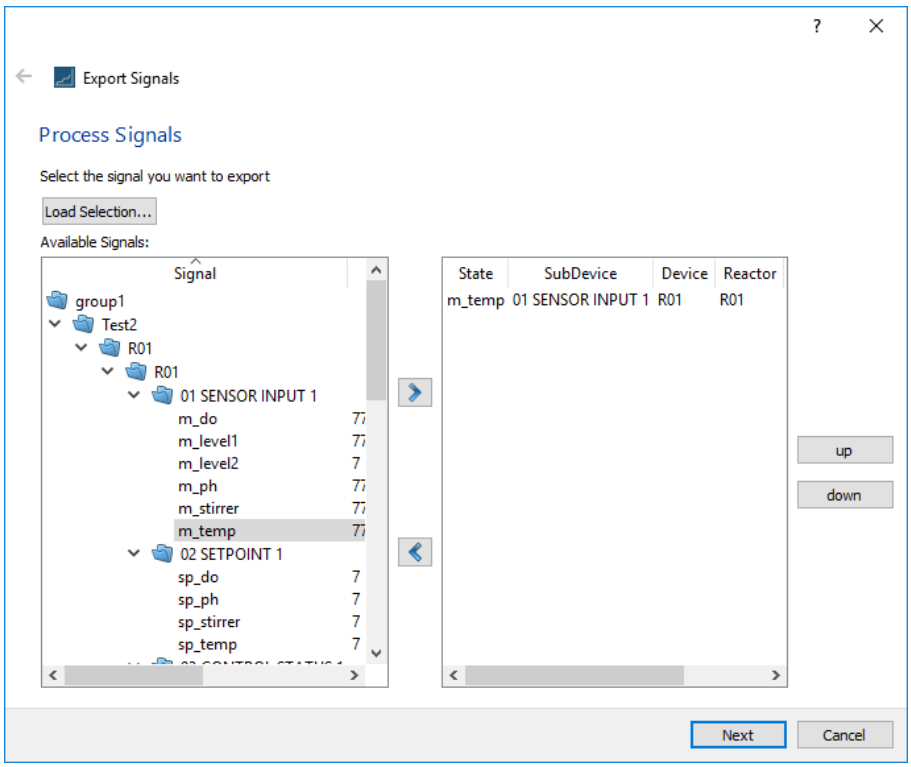
Now navigate to **File → Export Signals …**



The signals available for export are displayed.

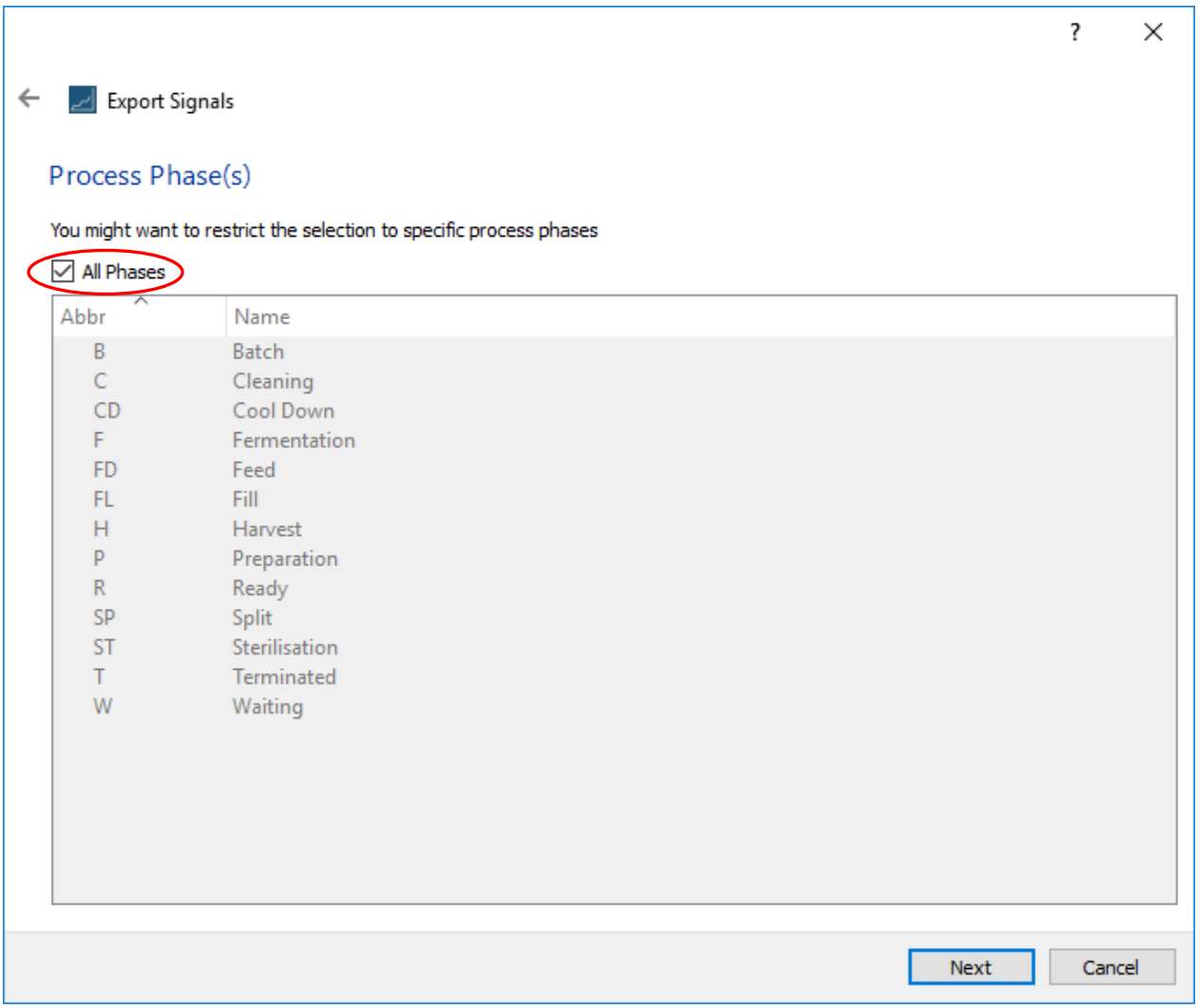


Select, for this example, **m_temp** to export.

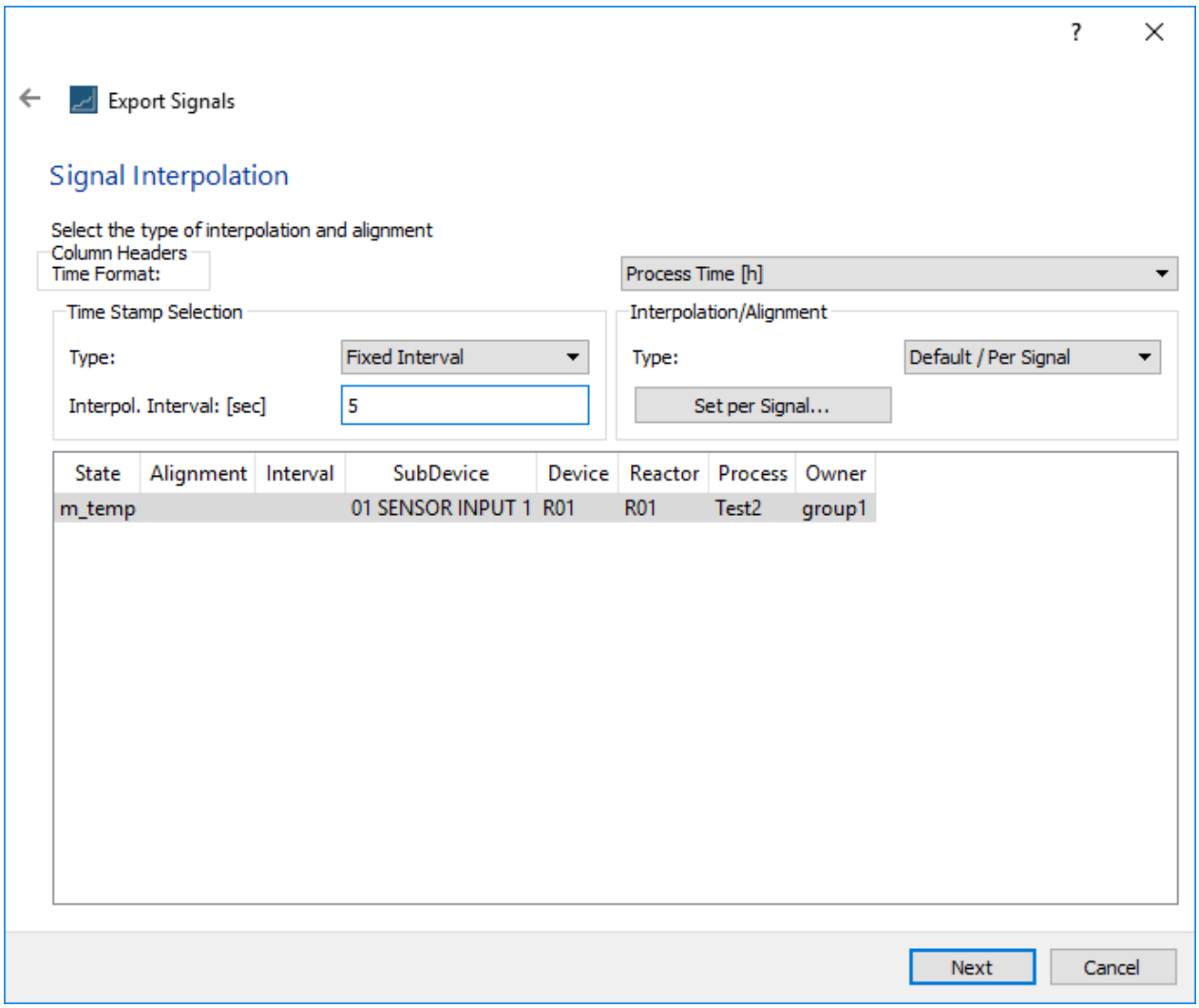


Click on '>' to transfer it to the right pane. Then click on **Next**.

This invokes the phase screen:

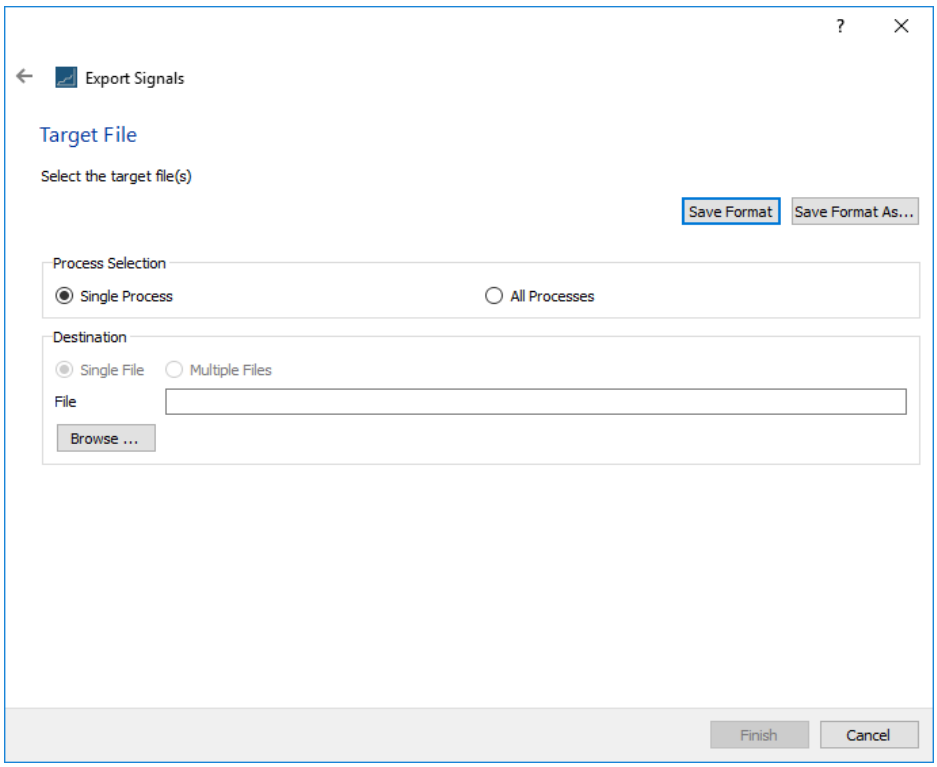


In this window you select for which phases of the process signal data is to be exported. In this case we select **All Phases** (circled).



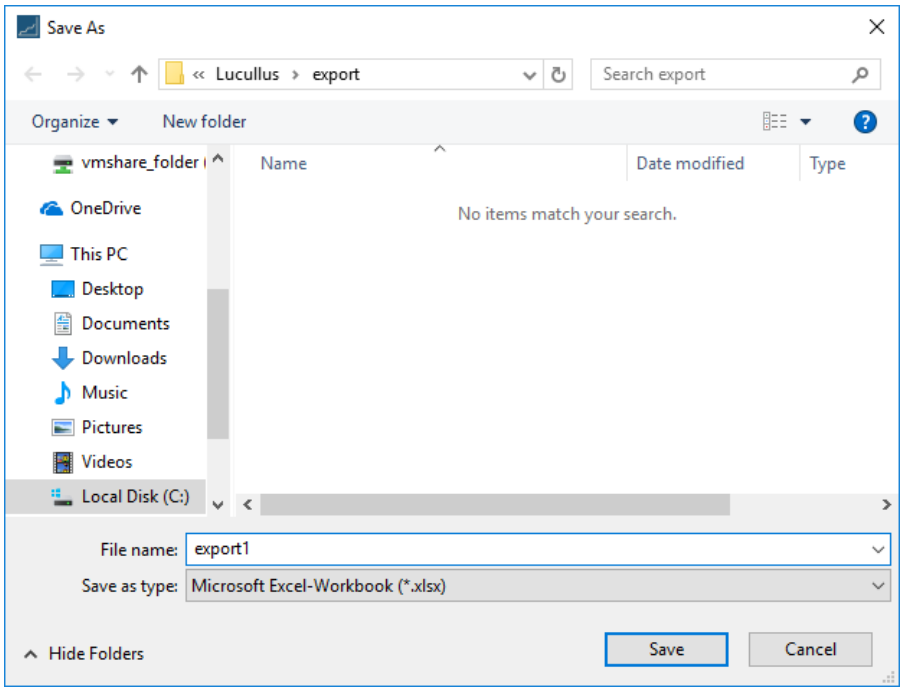
In this next window, we can take the data with the time stamps as they are, or we can force to have equispaced data, in which case the system supplies the intermediate values between two existing consecutive timestamps by linear interpolation. Here we select fixed interval time stamps with an interval of 5 seconds.

Select the signal to export from the list of signals and press **Next**:

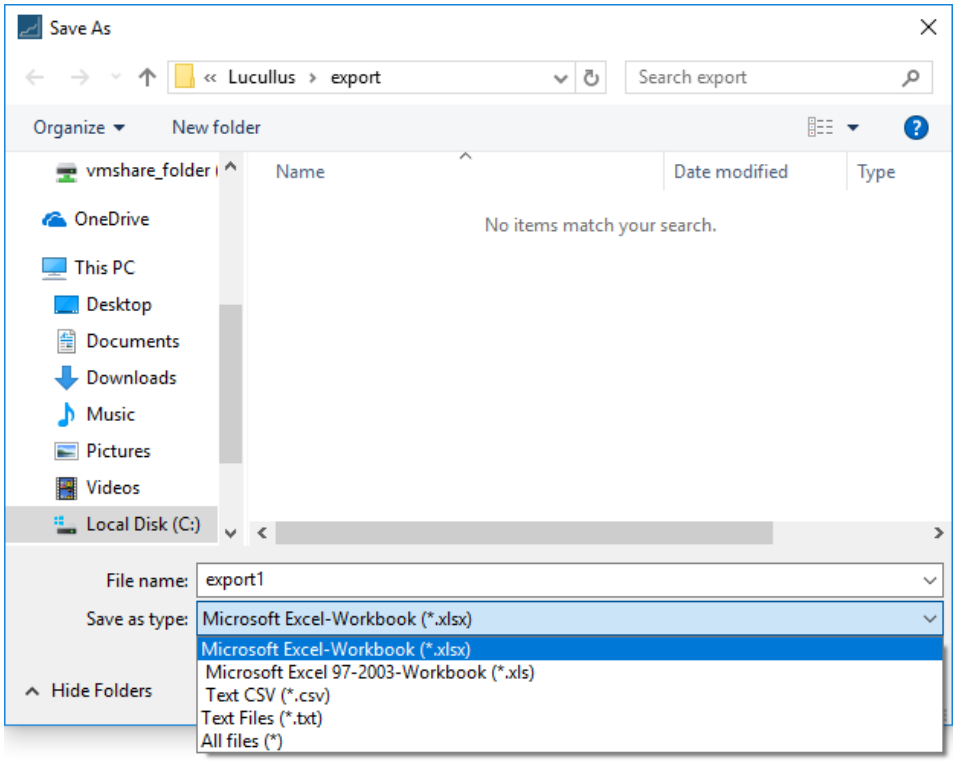


Here one can save a single process which is automatically saved as a single file. If you save all processes you can choose to have the data for all processes in a single file or one file per process.

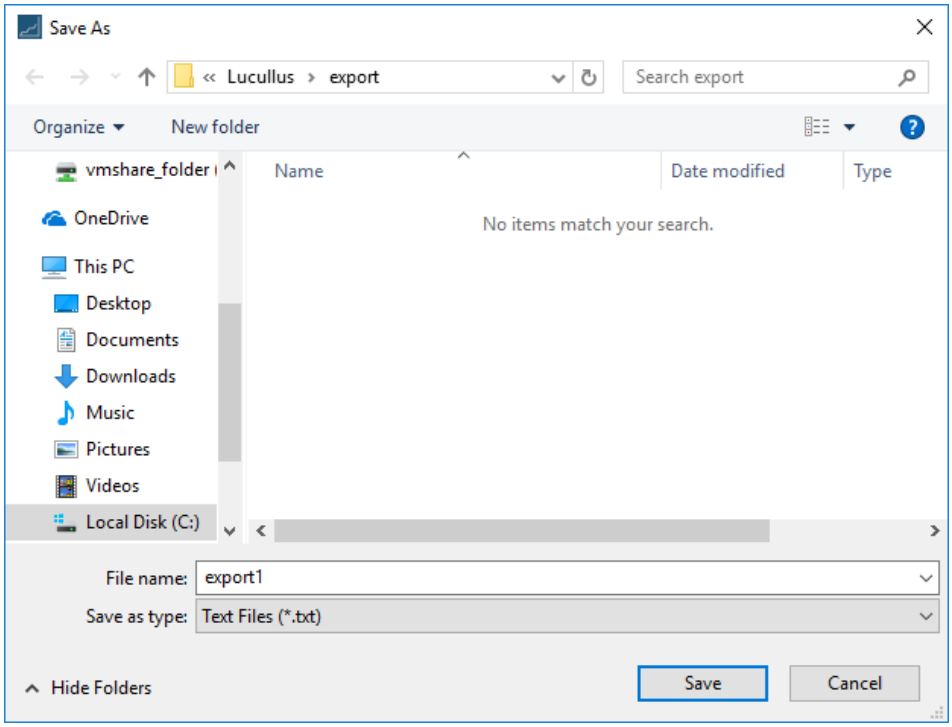
For the current example, we choose **Single Process**. Then click **Browse** to specify the target location **and** the file name:



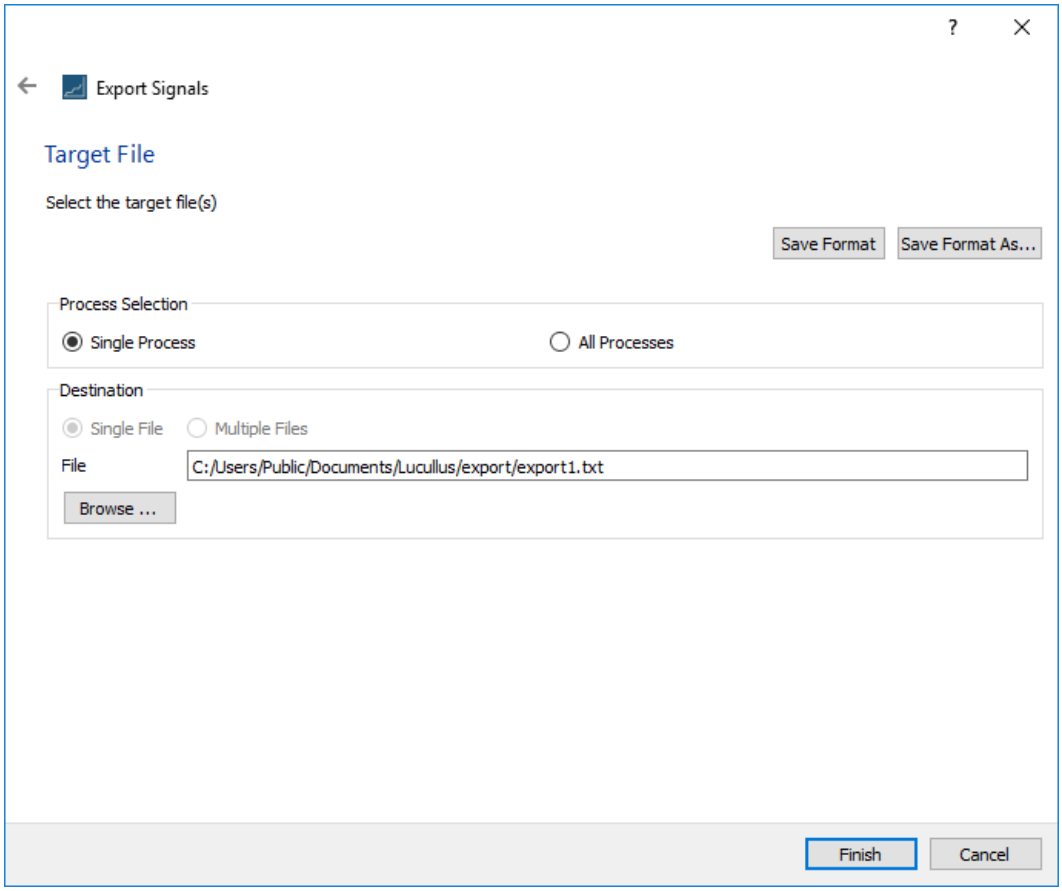
You can select the format of the output:



The file will be saved in C:\lucullus\export, under file name export1 and you can choose the format as XLS, CSV, or TXT:



Click on **Save**. The next window is shown:



Click on **Finish** and the export is created. The file extension you type determines the output format.

An exported .TXT file *can* look as follows:

```
Time [h]        m_temp
0.000805556     19.6
0.00302778      19.6
0.00608333      19.6
0.00858333      19.6
0.0113611       19.6
0.0141389       19.6
0.0169167       19.6
0.0196944       19.6
0.0224722       19.6
0.02525         19.6
0.0280278       19.6
0.0308056       19.6
0.0335833       19.6
0.0366389       19.6
0.0391389       19.6
0.0416389       29.5
0.0444167       29.5
0.0471944       29.5
0.0499722       29.5
0.05275         19.5
0.0555278       9.5
0.0583056       1.6
0.0610833       1.6
0.0641389       1.6
0.0669167       49.5
0.0694167       49.5
```

An exported .csv file *can* look as follows:

```
Time [h],m_temp
0.000805556,19.6
0.00302778,19.6
0.00608333,19.6
0.00858333,19.6
0.0113611,19.6
0.0141389,19.6
0.0169167,19.6
0.0196944,19.6
0.0224722,19.6
0.02525,19.6
0.0280278,19.6
0.0308056,19.6
0.0335833,19.6
0.0366389,19.6
0.0391389,19.6
0.0416389,29.5
0.0444167,29.5
0.0471944,29.5
0.0499722,29.5
0.05275,19.5
0.0555278,9.5
0.0583056,1.6
0.0610833,1.6
0.0641389,1.6
0.0669167,49.5
0.0694167,49.5
```

Excel format is also possible. If one wishes to include additional information, this will be part of the header – but the precise layout may differ depending on the version of Lucullus.

Exporting from the current run

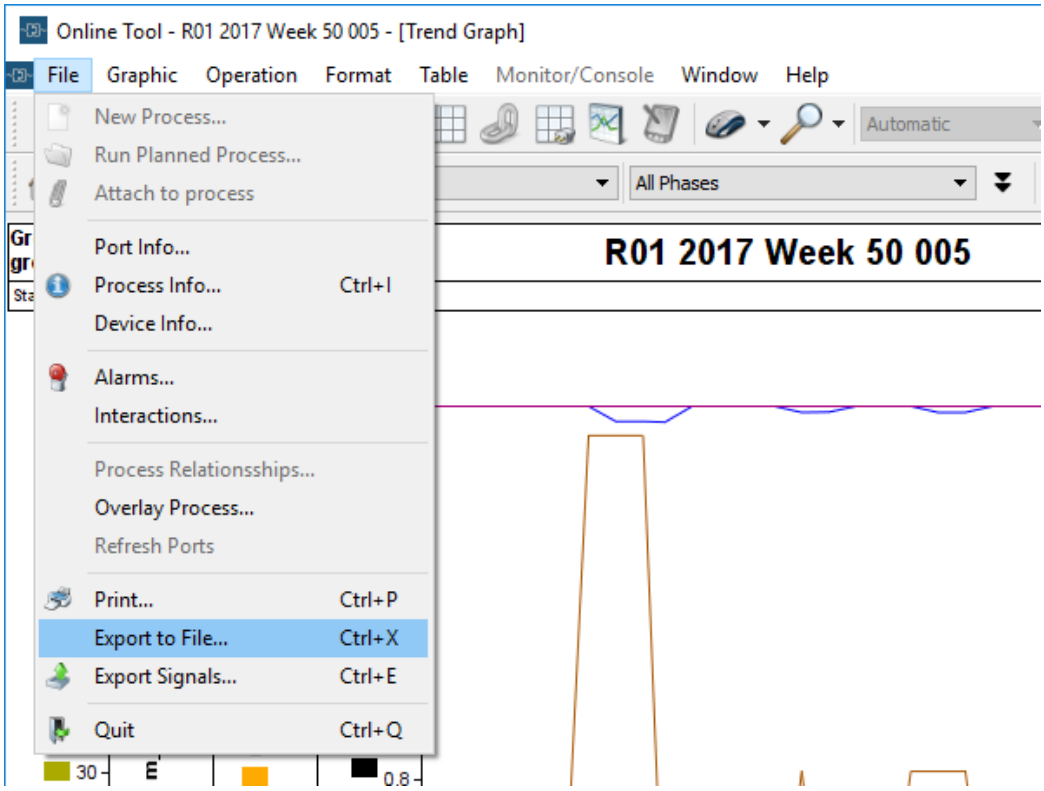
Exporting data from the current run is almost identical to exporting data from a prior run. The only difference is that the data is already "loaded", so you can immediately proceed to export the desired signals. The procedure is as follows:

1. Navigate to **File → Export signals**
2. Select the desired signal
3. Select the phases (select **all** if in doubt)
4. Select the mode for time intervals and interpolation options
5. Provide target folder and file name
6. Complete the export

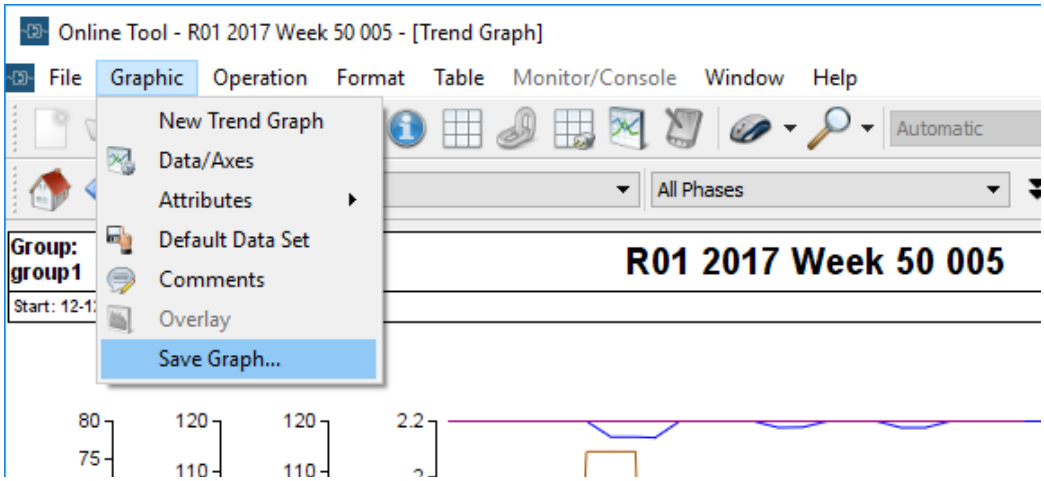
Steps 2-6 here are identical to the steps 3-7 of the previous section.

3.8 SAVING GRAPHS

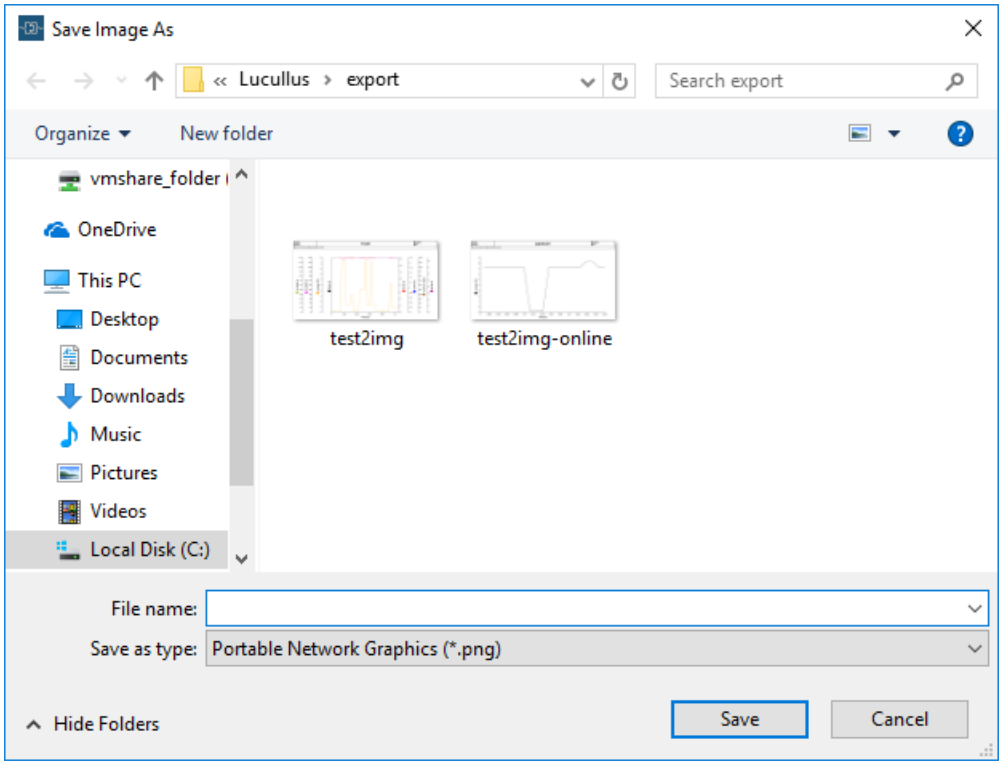
You can save an image of the trend of the currently running process or of a loaded data set; the procedure is the same in either case. For a running process, you can save a screen dump via the following menu selection:



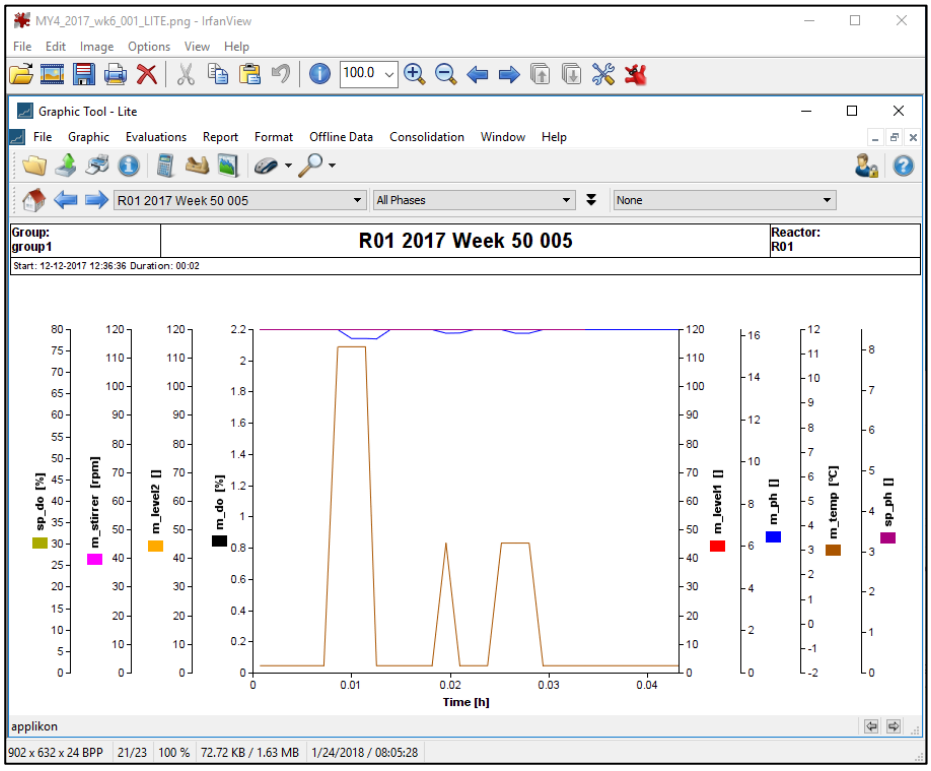
Or the following menu selection:



Either of these will popup the same file save dialog:



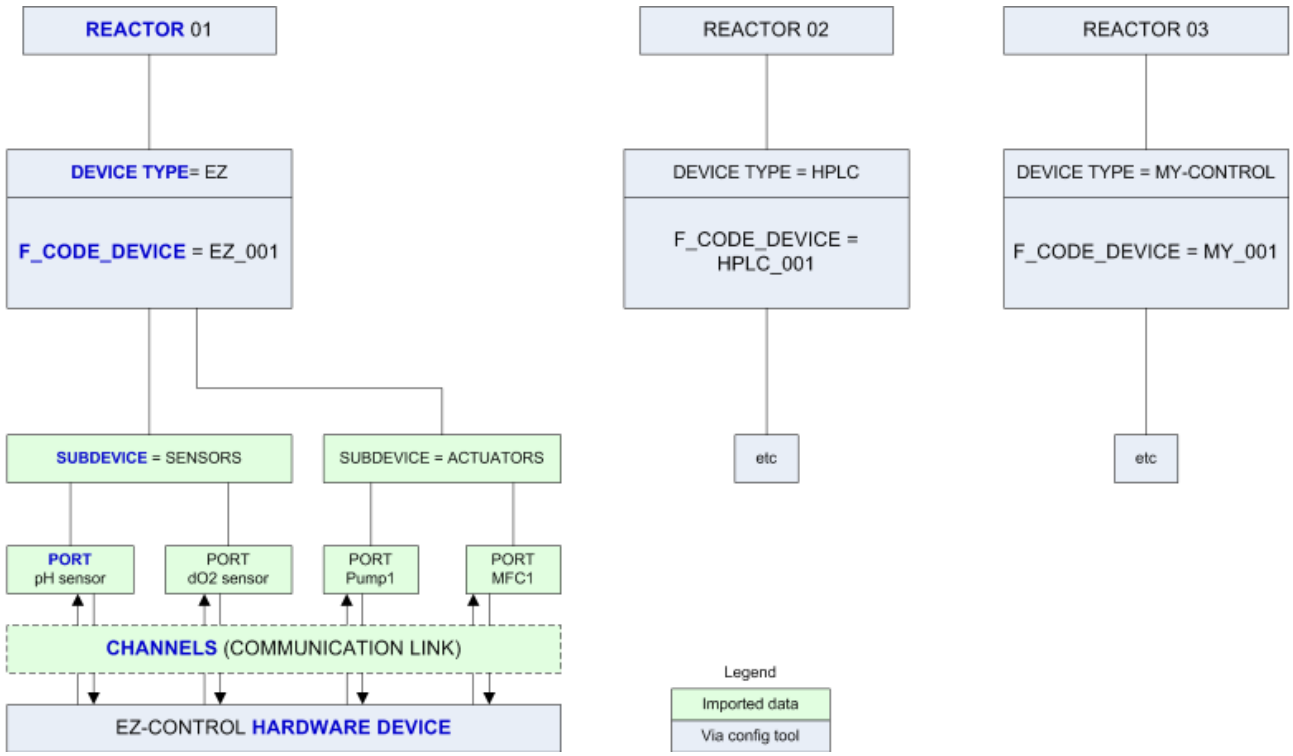
Whereby the default is to save the image in PNG format. Here is the saved image is displayed using the IRFAN image viewer:



4 APPENDIX: BASIC CONCEPTS

Not only does Lucullus support an extensive set of functionalities, but it can also control broad installations with many different cell culturing systems and auxiliary systems, all running in tandem. These components can be clustered and linked in many different ways and this flexibility has given rise to clustering or grouping abstractions that allow Lucullus to accommodate many different configurations.

The following diagram gives an abstract representation of a system from the point of view of how it is configured during the initial configuration process. The terms to be defined are given in blue.



The above diagram shows data objects representing actual hardware in the real world, such as a *bioreactor*. Other data objects such as *subdevices* do not correspond to an actual piece of hardware but represent a cluster of other logically related data objects.

With the aid of the above diagram, the main concepts are summarized as follows:

Component	Description
Bioreactor	A data object representing a bioreactor. Does not embody any functionality, just the attributes (e.g. volume, name, etc.) of an actual bioreactor or similar device. Some real world examples are: Reactor L01 - Glass reactor L01 Reactor L02 - Glass reactor L02 Reactor M01 - Steel reactor M01 Reactor MB01 - Microbiological reactor MB01 Shaker A01 - Shaker A01
Device Type	Refers to the *category* of a device, not any instance of a device. It holds the general "characteristics" or attributes of a particular kind of device. Examples: Applikon EZ Applikon MyControl Braun GA Feed Control system
Device	The designation for a particular device *instance* as a data object. The instance need not refer to a physical instrument; the device instance can also be a so-called *logical system device* implemented in software, such as an exponential ramp. Examples: Applikon EZ01 Applikon EZ02 Applikon EZ03 Applikon MY01 Applikon MY02 Braun GA01 FEEDCTRL01
Hardware device	The designation for a particular device instance as a hardware device. This refers to the actual *thing* - a physical instrument with nuts and bolts and electronics. The driver used to control and read from that instrument is also defined here. A hardware device will be coupled to its device (instance). So if one wishes to couple a particular ***ez***-Control physical device to instance *Applikon EZ02*, then one has to also name the new hardware device *Applikon EZ02*. The equality of the names is used by the channel import procedure to associate the right Device with the right Hardware Device.
Channel	A data object that defines the communication link between a hardware device and a higher level system, such as a SCADA system. One can consider this from the standpoint of defining a *channel* that opens a path to a source (and recipient) of data; it opens a path for input and/or output. During configuration, one defines a device and gives it a name and one then chooses a hardware device and gives it the same name. Using the same name creates a logical association but still does not define how the two will communicate. This is done by defining a **channel**. One can define the communication manually using the various Lucullus configuration screens, but it is more convenient to enter them into an Excel spreadsheet in a specific column format and then import the spreadsheet. An import is often done because there may be hundreds of channels and ports. Ports are defined next.
Port	A data object representing a single data entity (input or output). For instance, pH, Temperature, Stirrer output.
Subdevice	An abstract data object representing a cluster of ports. It is a useful *organizing or grouping* construct, as the following examples illustrate:

	Take for instance a reactor REACTOR01 which will be controlled by Device EZ_001. The subdevices of EZ_001 are: ACTUATOR LIMITS HL_acid HL_alkali HL_air LL_acid LL_alkali … ACTUATOR OUTPUTS Output_acid Output_alkali Output_air Output_CO2 … BALANCES Balance_01 Balance_02 CONTROL PARAMETERS ph_cycletime ph_deadzone ph_I ph_P … … and many more. As one can see, there is justification to clustering ports in the way shown above. ACTUATOR_LIMITS, for example, clusters the ports corresponding to the minimum and maximum values of the system actuator outputs in use.
F_code_device	The device driver software component for controlling and receiving input from Applikon controllers. This component will drive: ***in***-Control ***my***-Control ***ez***-Control