New Features in ibaAnalyzer 6.4.0

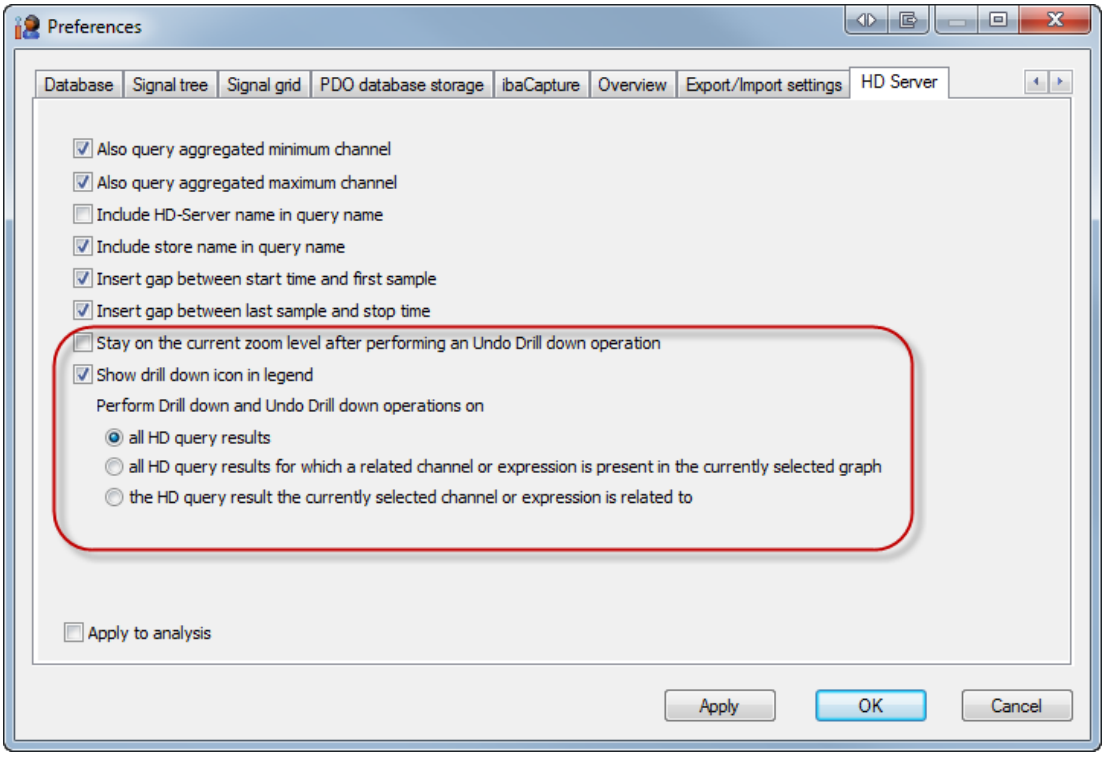
Author: Michael Verschaeve

Date: 4 November 2013

Table of contents

1 New HD query options

The HD Server options tab (available both in the preferences as in the analysis setup) has been extended:



The following new options are available;

- ❑ *"Stay on current zoom level after performing an Undo Drill down operation"*.
- ❑ *"Show drill down icon in legend"*
- ❑ *"Perform Drill down and Undo Drill down operations on …*

1.1 The option "*Stay on the current zoom level …"*

The "*Drill down*" operation is basically a shortcut for starting a new HD query with parameters adapted to the current zoom level. The "*Undo Drill down*" operation is a shortcut for returning to the previous HD query before the "*Drill down*" operation was performed. Performing an HD query is in its turn analogous to opening a `.dat` file. In ibaAnalyzer when a `.dat` file is opened, a *"Zoomout all"* operation is performed in order to visualize all available data. Hence when executing an "Undo Drill down" operation, the "*Zoomout all"* operation was done as well.

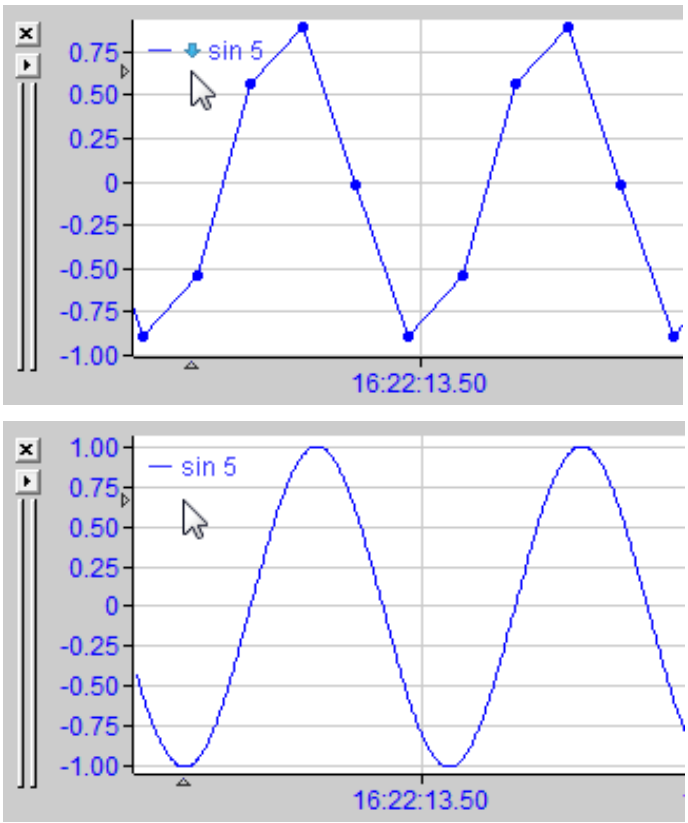
This behavior might not be desirable; hence in the current version of ibaAnalyzer you have the option to disable the zoomout by checking this option. This option is disabled by default.

1.2 The option "*Show drill icon in legend*"

Optionally one can set that ibaAnalyzer shows an icon in the channel legend when the data can still be "drilled down", i.e. when there is finer sampled level of data available on the HD server. The icon is not shown when one is on the *raw* data, i.e. when the sampling level is equal to the acquisition level and hence is the finest level available.

The icon can be clicked in order to perform a drill down operation. The icon is grayed out when there is finer sampled data available but a drill down operation is not possible, which is currently only the case when the user has not zoomed in. When clicking the icon, before performing the drill down operation, the channel and graph are selected before the drill down operation is performed as the selected channel can influence the drill down operation (see paragraph 0).

This option is enabled by default.



1.3 The option "*Perform Drill down and Undo Drill down operations on …"*

In previous versions of ibaAnalyzer, "*Drill down*" and "*Undo Drill down*" operations were performed on any HD query for which there was a signal in the currently selected graph or for which an expression in the currently selected graph had an argument that was dependent on the HD query.

In the current version of ibaAnalyzer, the behavior of "*Drill down*" and "*Undo Drill down*" can be fine-tuned by selecting on which HD queries the actions need to be performed.

- "All HD query results", simply perform the actions on all open HD queries.
- "All HD queries for which a related channel or expression is present in the currently selected graph". This allows one to drill down certain HD queries while leaving other queries unaltered. This was the behavior of previous versions of ibaAnalyzer.
- "The HD query the currently selected channel or expression is related to". This allows one to drill down a specific HD query while leaving all other queries unaltered.

The default behavior is "All HD query results", hence different from previous versions of ibaAnalyzer.

2 New and adapted functions in ibaAnalyzer

2.1 VectorToSignal function

This function was implemented in version 6.1.1. To recapitulate, it takes the following arguments:

- Vector: The vector to take the samples from
- XBase: The sample rate of the returned signal.

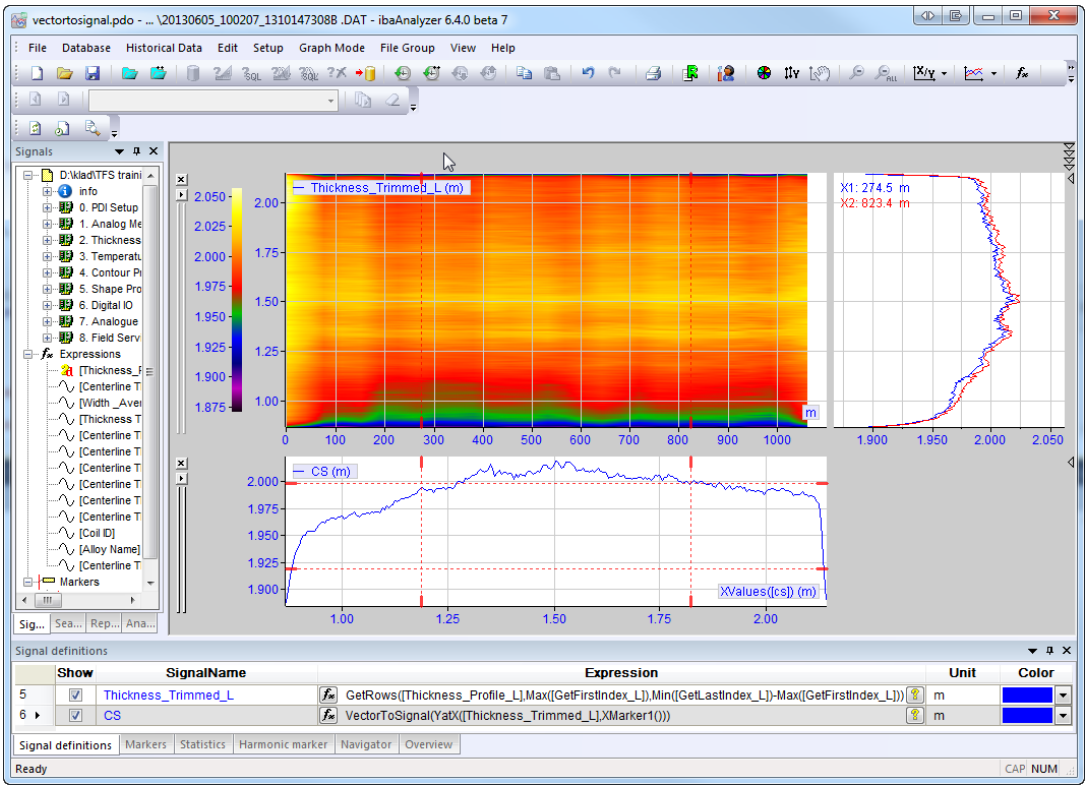
This function creates a (1 dimensional) signal from the elements of a vector. Each sample of the resulting signal corresponds with an element in the vector. The elements in the vector are expected to be constant signals, if they are signals that vary over time, they will be averaged first before being used as samples in the resulting signal.

In version 6.4.0 of ibaAnalyzer, this function has been altered so that the second argument is optional. When the second parameter is omitted, the zone widths and Z-offset of the vector is used. The resulting signal can be non-equidistant.

The function was originally implemented to be used in the “Table data” tab of the report generator, however with the recent changes, the function is especially useful to generate cross profiles of thickness plots. For example, given a thickness profile **`[Thickness_Profile]`**, the following expression

`VectorToSignal(YatX([Thickness_Profile],XMarker1())`

gives a cross profile at the location of the **X1** marker.



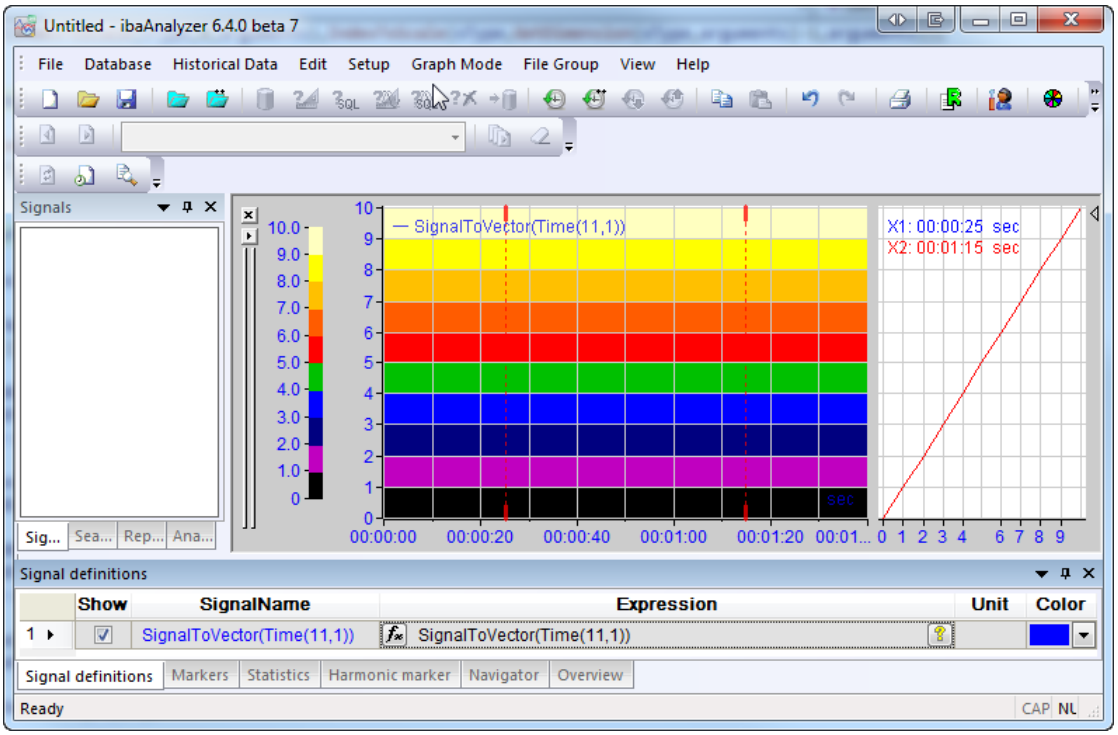
Note

For plotting the result of a *VectorToSignal* function, it is useful to create a secondary expression that uses the *XValues* function on the result of the *VectorToSignal* function. This signal can then be used to create an X-axis with an XY plot. This way the X range of the *VectorToSignal* plot is independent of the original signals.

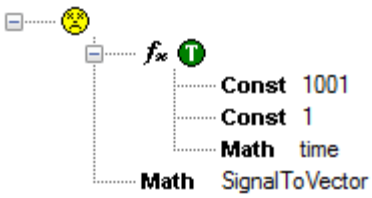
2.2 SignalToVector function

Analogous to *VectorToSignal*, *SignalToVector* was implemented to create a vector from a signal. A vector of constants is created, each element of the vector corresponding to a sample in the input signal. The Z-offset and zone widths of the resulting vector are determined by the X-offset and sample rate of the input signal.

The function takes only one argument, namely the signal to create a vector from.



The input signal must contain 1000 or less samples, if not, the expression will not be evaluated and be marked as *too complex* (when pressing the diagnose button next to the expression, a smiley with crossed out eyes will be depicted).



The function was created to facilitate providing vectors to the *chart fields* tab in the report dialog (see paragraph 4.1).

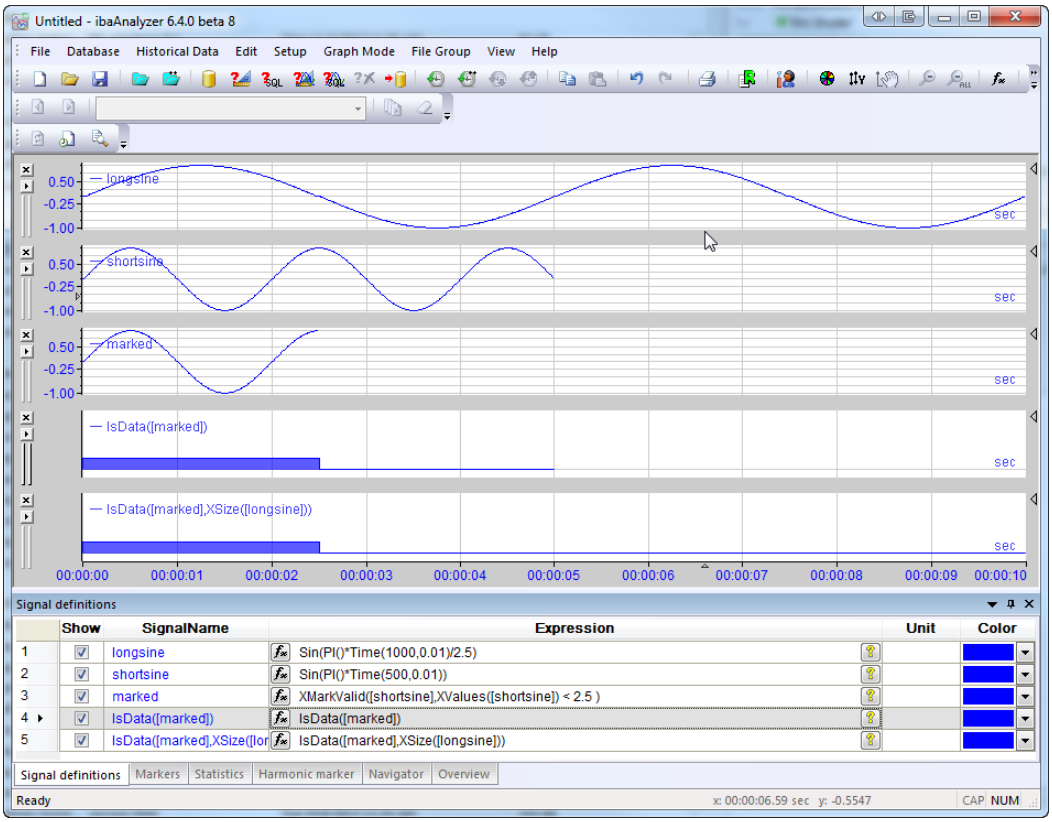
Note

Contrary to the VectorToSignal function, this function has no optional arguments to set the zone widths and Z-offset. For this we recommend the **`SetZoneWidths`** function.

2.3 IsData function

The IsData function has been adapted so that the required length in X-axis units of the returned signal can be specified. If this optional parameter is omitted, the function behaves as it used to, i.e. the returned length is the same as the length of the input signal (including invalid samples).

With the optional parameter, the returned signal can be shortened or lengthened to match the size of another signal.



This modification was implemented in version 6.3.1 of ibaAnalyzer but has not been documented until now.

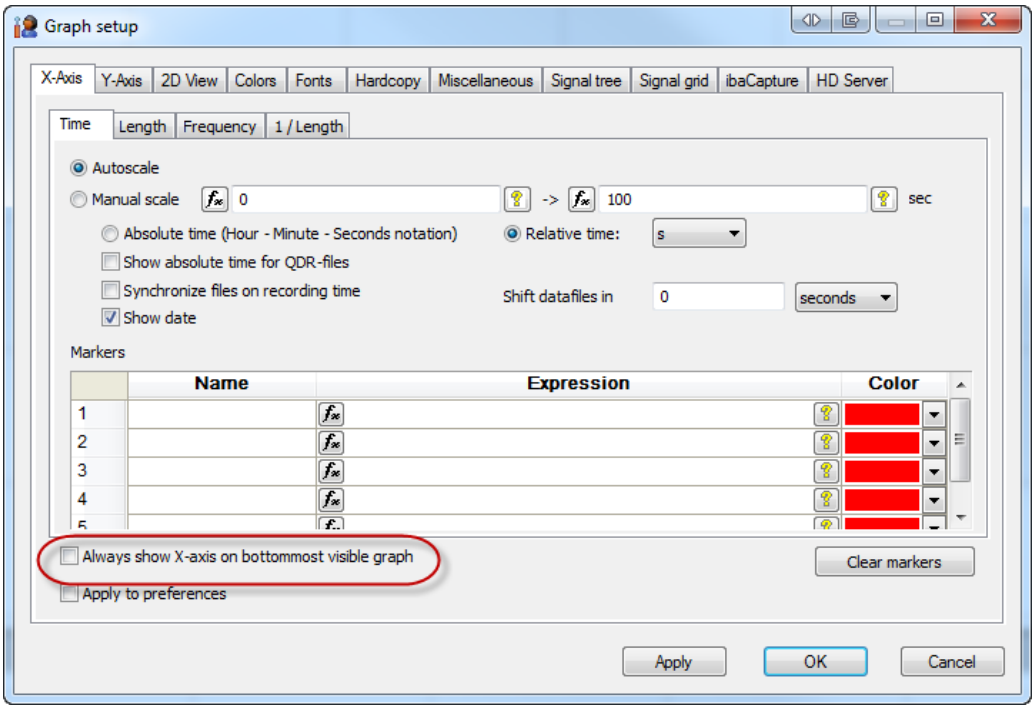
3 New visualization features

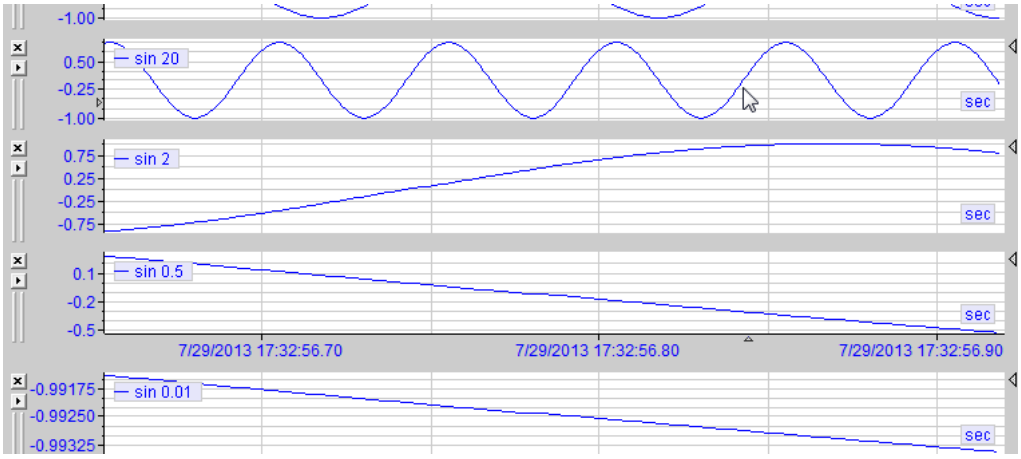
3.1 Always visible X-axes

By default, ibaAnalyzer depicts the X-axis below a graph only for the last graph that has that X-axis before a graph with a different X-axis follows. This means that the X-axis might not be visible for the currently scrolled portion of the recorder view.

In the current version of ibaAnalyzer, one can chose to have the X-axis to be also depicted on the last graph that is entirely visible for the currently scrolled portion of the recorder view.

The setting is available in the *Preferences* and *Graph Setup* under the X-axis tab, all sub tabs (Time, Length, Frequency, 1/Length) have the option, however it is common for all X-axes, altering the option in one subtab will also alter it in the other tab. By default the option is disabled.

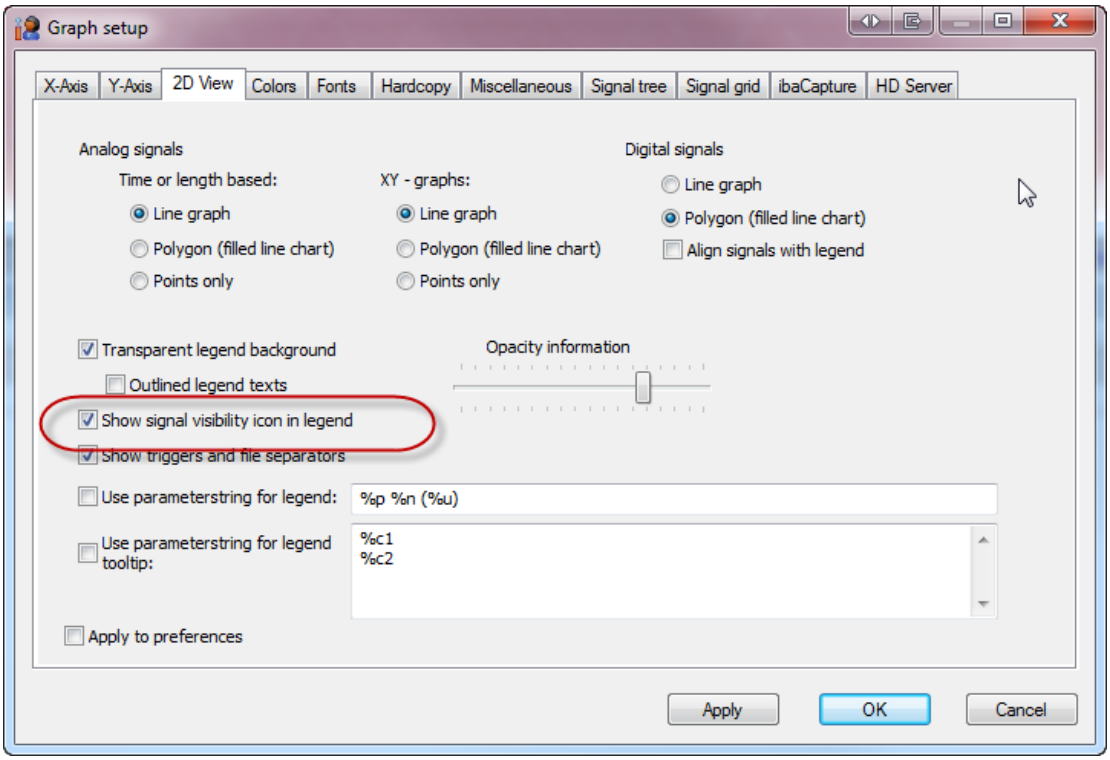


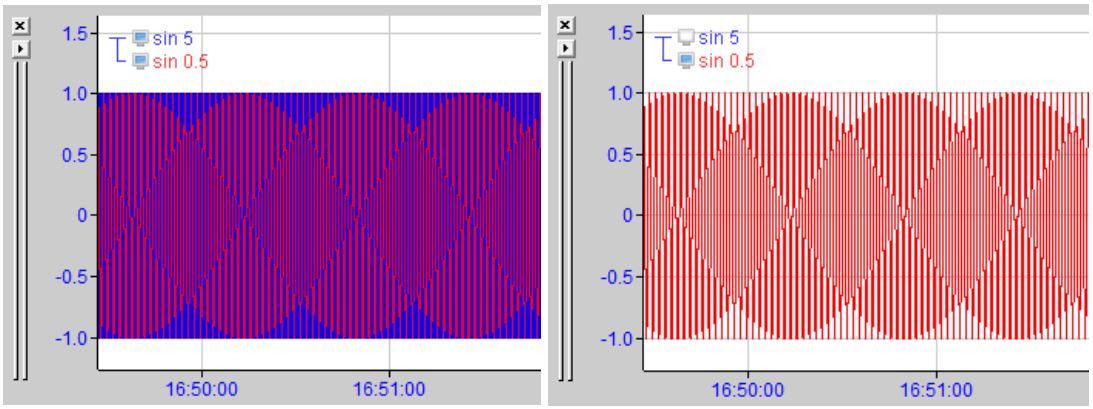


3.2 Show-hide symbol in legend

Similar as in ibaPDA (Version 6.29.0) one can chose to have a little icon present in the signal legend where one quickly can toggle whether or not a signal is shown.

The setting is available in the Preferences and Graph Setup under the 2D View tab.



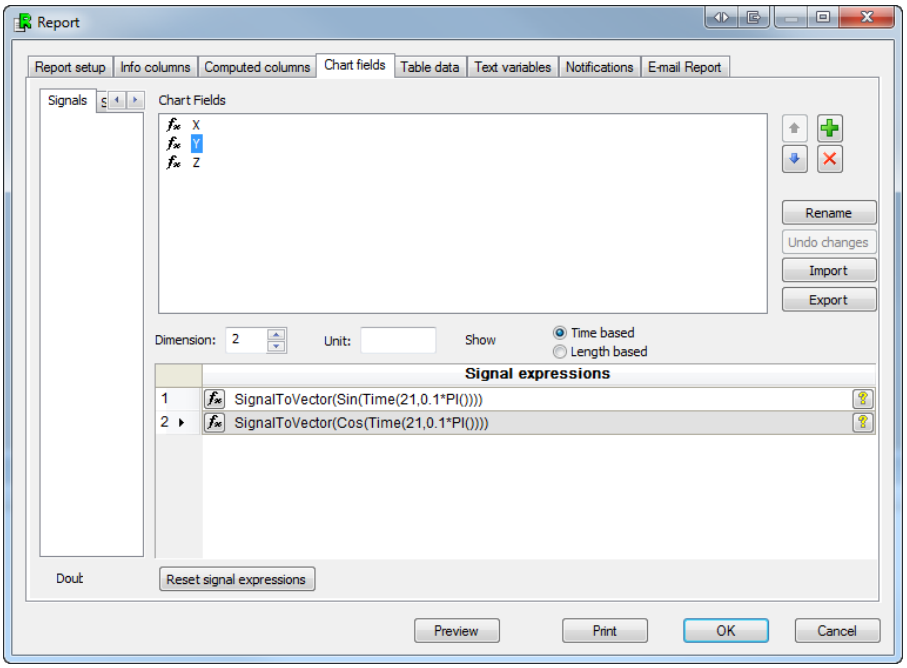


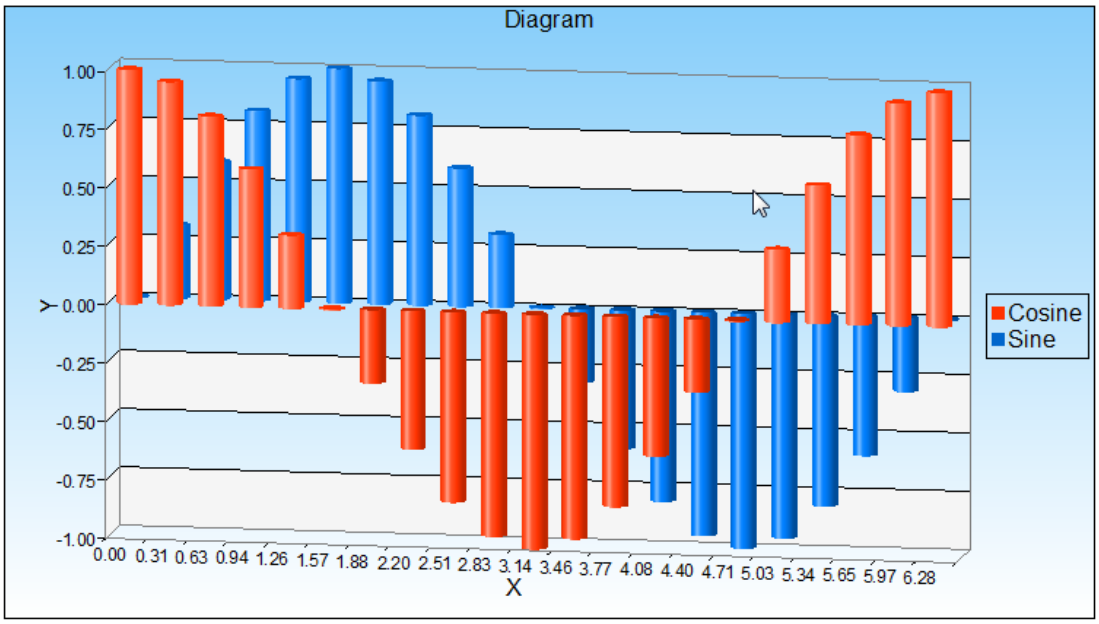
Toggling the visibility of a signal with this option does not affect the “Show” state of a signal as specified in the signal grid. The two ways of hiding signals are quite different since hiding a signal by clicking the checkbox in the “show” column will also hide the legend, affect auto scaling of the graph and possibly hide the graph entirely, while clicking the icon does not have any other effect than hiding or showing the signal.

4 Miscellaneous new features

4.1 Multidimensional expressions in chart fields

In previous versions of ibaAnalyzer, the *Chart fields* tab in the *Report* dialog expected a list of expressions, one expression for each value of the chart. This was cumbersome, so this has been altered so the *Chart fields* tab can take a multidimensional expression as one of the elements. The rows of the vector are then used for the chart values. You can also specify multiple vectors in which case the rows of all vectors get appended so a larger vector results, this is identical as how one can create logicals from multiple vectors in the logicals dialog.



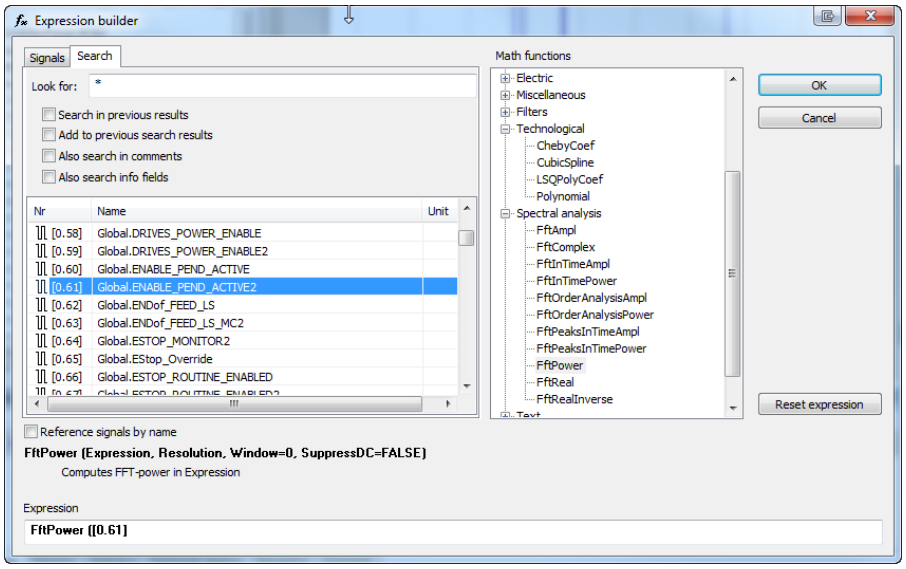


4.2 Search tab added to signal trees in several dialogs

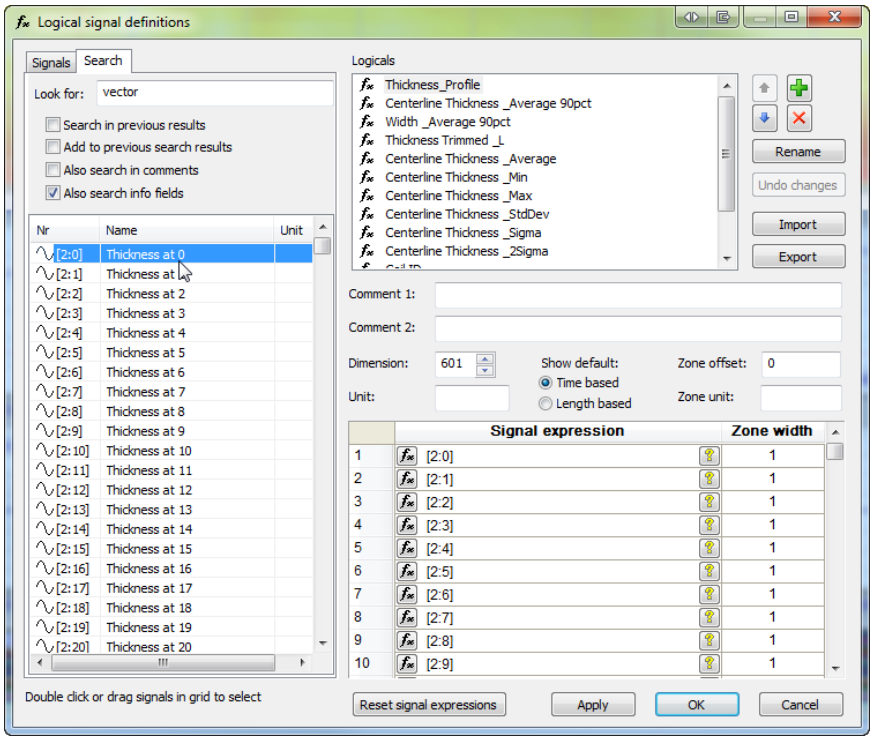
In all dialogs where a signal tree is present where signals can be selected to use in expressions, a search tab has been added as an extra tab to the signal tree. This tab is identical to the search tab present in the main ibaAnalyzer view and allows one to search for signals to use in that dialog. Contrary to the main ibaAnalyzer view, the tab cannot be untabbed.

The affected dialogs are:

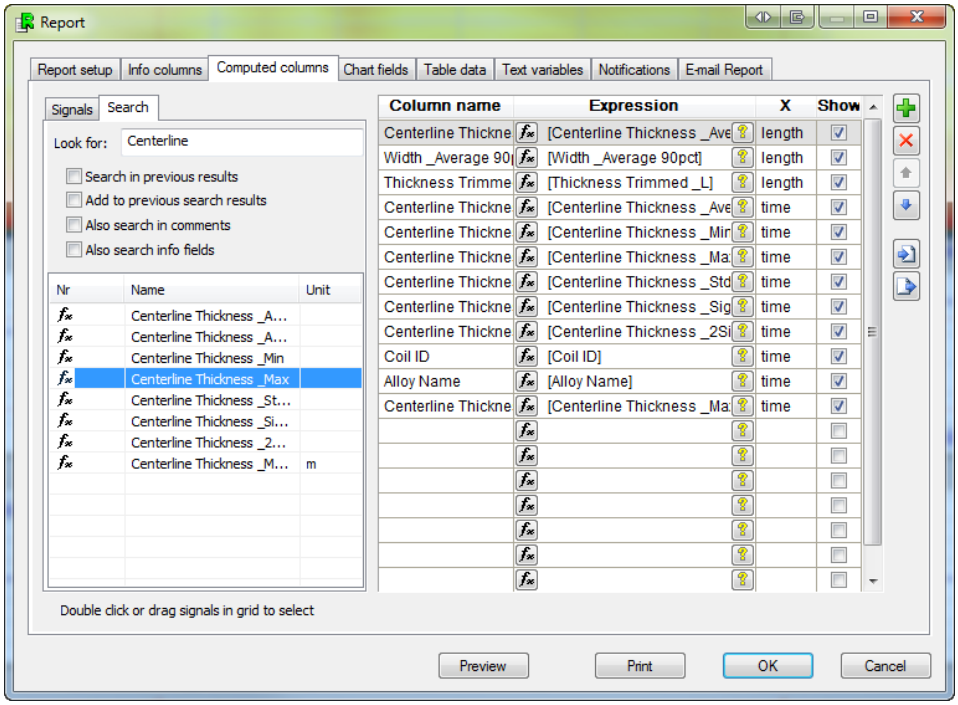
- *Expression builder*



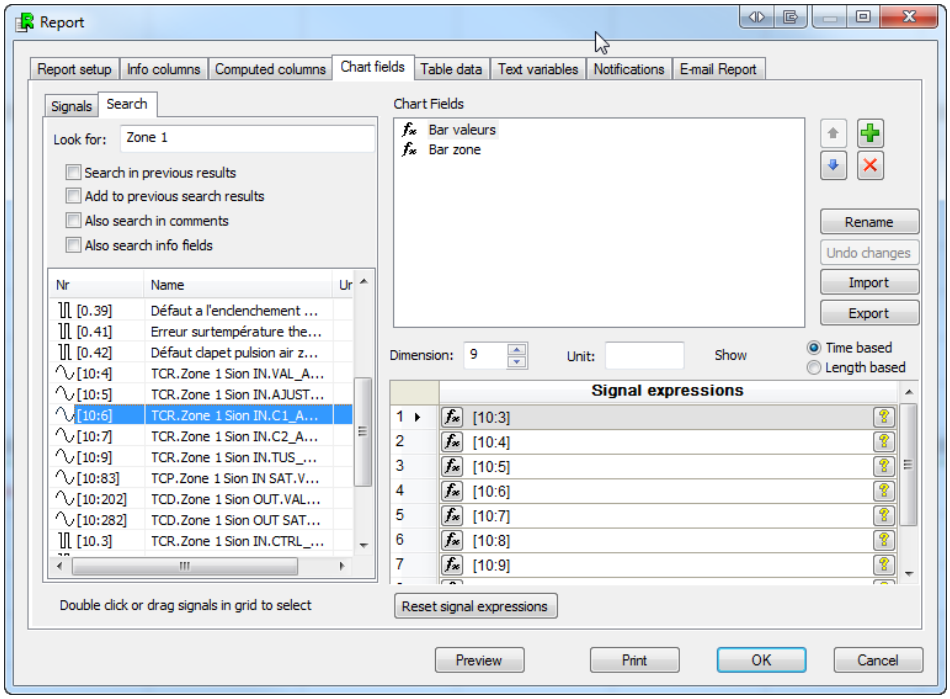
- *Logical signal definitions*



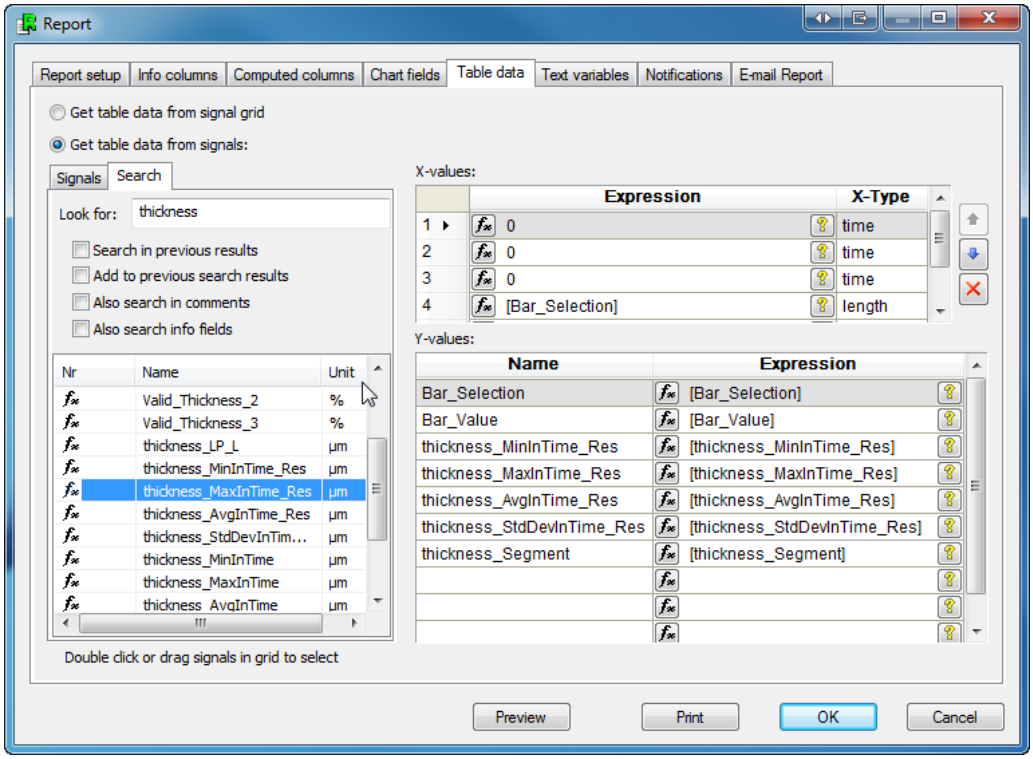
- *Computed columns* tab, both in the report and extract dialog



- *Chart fields* tab in the report generator



- *Table data* tab in the report generator



4.3 Buttons added to grids

In the *computed columns* and *info columns* tabs in both the *Report* and *Data Extractor* dialogs, 6 new buttons have been added

-  Adds an empty line to the grid at the current row, the current row and all subsequent rows are moved one row down.
-  Removes the current row in the grid. All subsequent rows are moved one row up.
-  Swaps the current row in the grid with the row above it. This button can be clicked repeatedly to move the current row several rows upwards.
-  Swaps the current row in the grid with the row below it. This button can be clicked repeatedly to move the current row several rows downwards.
-  Exports the entire grid to a text file.
-  Imports the entire grid from a text file saved with the export button. The entire content of the grid is replaced with the content of the text file.