New Features in ibaAnalyzer 6.9.0

Author: Michael Verschaeve

Date: 20 December 2016

Table of contents

1 Intervals

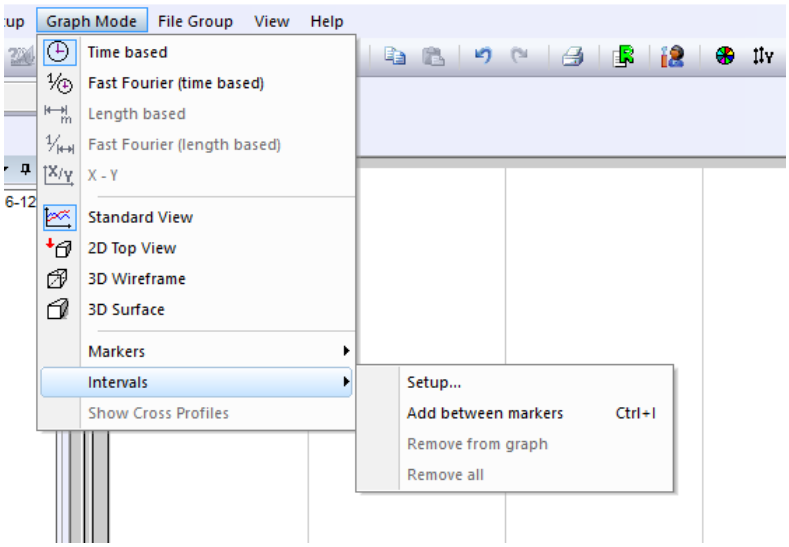
It was already possible in previous versions of ibaAnalyzer to annotate graphs with static markers. It was also possible to measure distances between events with the dynamic markers.

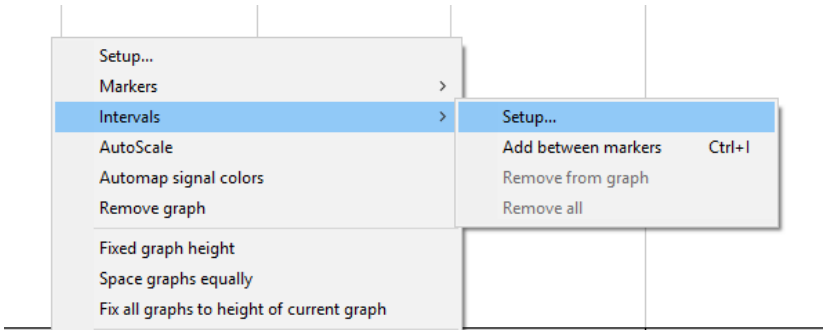
In the current version of ibaAnalyzer it is possible to combine these functionalities with the *Intervals*. These Intervals consist out of vertical lines on selected positions, a horizontal line with arrows (unless there is no room to draw the arrows) and a label that displays the distance between the two positions, optionally also a name for the interval is displayed in the label.

The intervals are associated with a displayed channel, moving the channel to another graph will also move the associated intervals.

1.1 The interval menu options

The intervals can be inserted and configured through menu options. These options are available in the ibaAnalyzer main menu under the *Graph Mode* submenu and in the context menu that pops up if you right-click on a graph. These menu options are themselves available under yet another submenu labeled *"Intervals"*.





The menu options are:

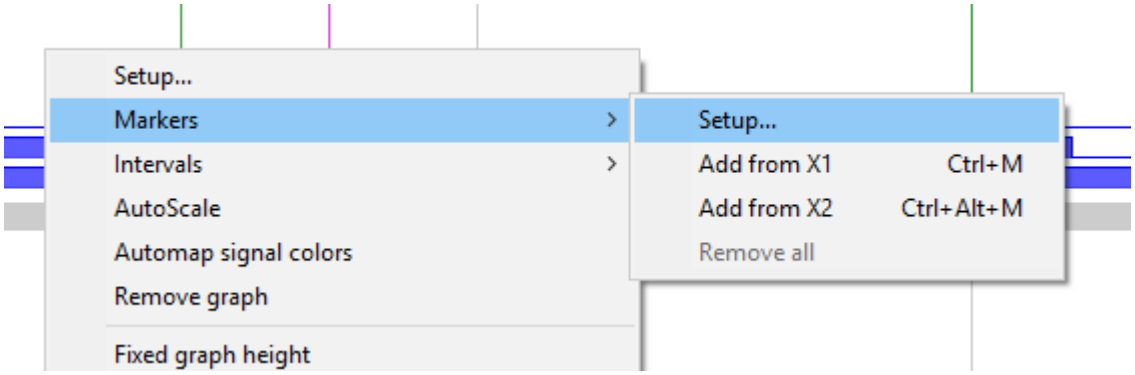
- *Setup…*: This opens a dialog where the intervals associated with the selected channel can be configured.
- *Add between markers:* This adds an unnamed interval between the dynamic X1 and X2 marker, if the X2 marker is in front of the X1 marker, the displayed distance will be negative. If the keyboard shortcuts would insert an interval that is identical to an already existing interval, then the new interval will not be inserted. Instead the old interval will be deleted if it was visible, and will be made visible if it was invisible.
- *Remove from graph*: This deletes all intervals associated with the channels present in the selected graph. This option is disabled (grayed) if no intervals are present on any channels in the graph.
- *Remove all*: Removes all intervals from all channels. This option is disabled (grayed) if no intervals are present.

1.2 The marker menu options

To make the behavior with the intervals consistent, the marker functionality has also been moved to a submenu in the context and main menu labeled “Markers…”

The menu options are:

- *Setup…*: This opens the marker visibility dialog where the markers can be configured.
- *Add from X1* : Adds an unnamed marker at the same position as the dynamic X1 marker. If an identical marker was already present, its visibility will be toggled (i.e. if it was visible it will be hidden, if it was hidden it will be made visible).
- *Add from X2* : Adds an unnamed marker at the same position as the dynamic X2 marker. If an identical marker was already present, its visibility will be toggled (i.e. if it was visible it will be hidden, if it was hidden it will be made visible).
- *Remove all*: Removes all markers. This option is disabled (grayed) if no markers are present.

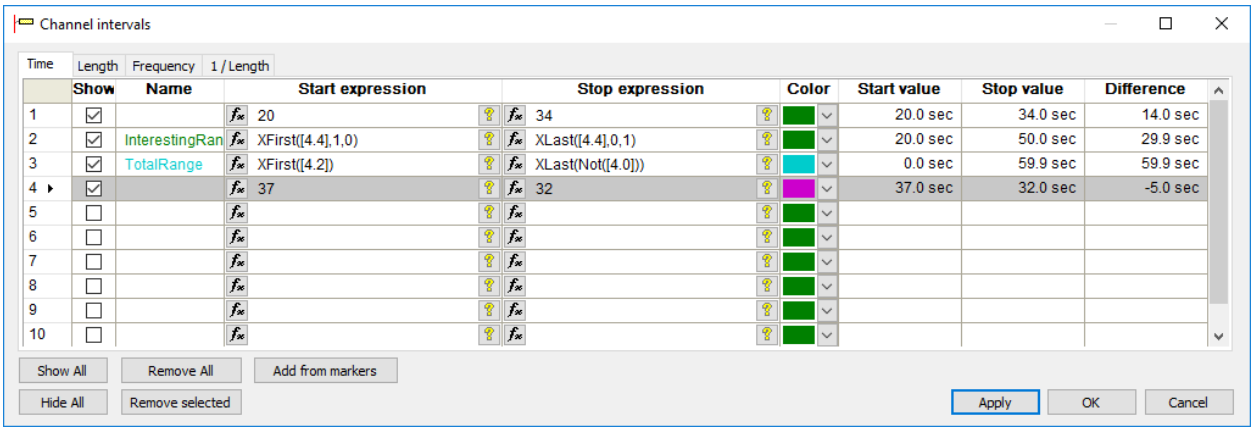


1.3 The interval dialog

This dialog allows one to configure the intervals associated with the currently selected channel.

One can specify for each domain (time, length, frequency, inverse length) intervals; a tab page is provided for each. Note that contrary to static markers, intervals are currently not supported for XY-views. The dialog consists out of the following elements:

- A grid where one can specify the intervals, containing the following elements:
  - A row header, numbering the intervals.
  - A column labeled *"Show"*, where one can specify if the interval is visible or not. This allows one to disable an interval without losing any specified start or stop expression.
  - A column labeled *"Name"*; here one can optionally specify a name for the interval. The name will be prefixed to the displayed distance for the interval. Leave this column blank if this is not desired.
  - A column labeled *"Start expression",* specify here an expression (a constant is also an expression) identifying the start of the interval.
  - A column labeled *"Stop expression"*, specify here an expression (a constant is also an expression) identifying the end of the interval. If this expression evaluates to a value smaller than the value that *"Start expression"* evaluates to, the displayed distance will be negative.
  - A color selector column where one can select the color of the displayed interval. The colors are selected out of 16 predefined colors (which can be adjusted in the *Colors* graph settings). By default the third predefined color is selected (usually green), which contrasts nicely with the default colors of markers (second predefined color, usually red) and signals (first predefined color, usually blue).
  - A column labeled *"Start value"*. This column cannot be edited. It will display the value to which *"Start expression"* evaluates. For newly defined intervals, the interval needs to be inserted first by pressing the *"Apply"* button before this column will display any value. If the expression cannot be evaluated, a question mark will be displayed.
  - A column labeled *"Stop value"*. This column is identical to the previous column except it displays the value "Stop expression" evaluates to.
  - A column labeled *"Difference"*. The difference between the previous two values. This is the value displayed in the interval when drawn on the graph. Again, for newly defined intervals, the *"Apply"* button must first be clicked.



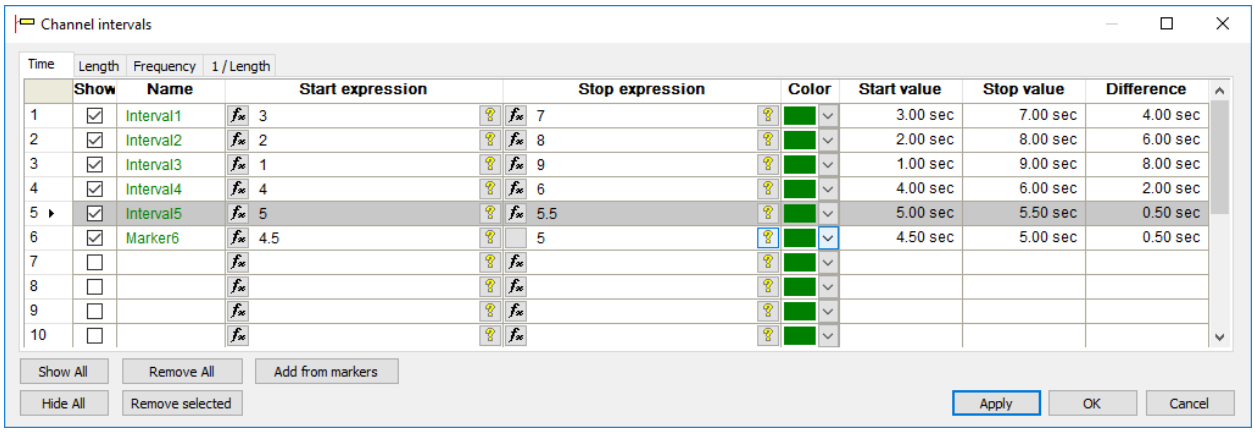
- A button labeled *"Show all"*. This sets the visibility column to checked for all intervals in the current domain tab.
- A button labeled *"Hide all".* This sets the visibility column to unchecked for all intervals in the current domain tab.
- A button labeled *"Remove all"*. This removes all intervals and hence clears the grid for the current domain tab.
- A button labeled *"Remove selected"*. This removes all selected intervals from the grid. To select an interval, select its row by clicking the row header, multiple entries can be selected by pressing *Shift* (for a range) or *Ctrl* (add additional entries to the selection).
- A button labeled *"Add from markers"*. Fills in a new row with start and stop expression taken from the positions of the X1 and X2 dynamic markers.
- A button labeled *"Apply"*. This applies any changes without leaving the dialog.
- A button labeled *"OK"*. This applies any changes and leaves the dialog.
- A button labeled *"Cancel"*. This discards any changes since the last *"Apply"* and leaves the dialog. If no *"Apply"* was clicked, all changes since entering the dialog are discarded.

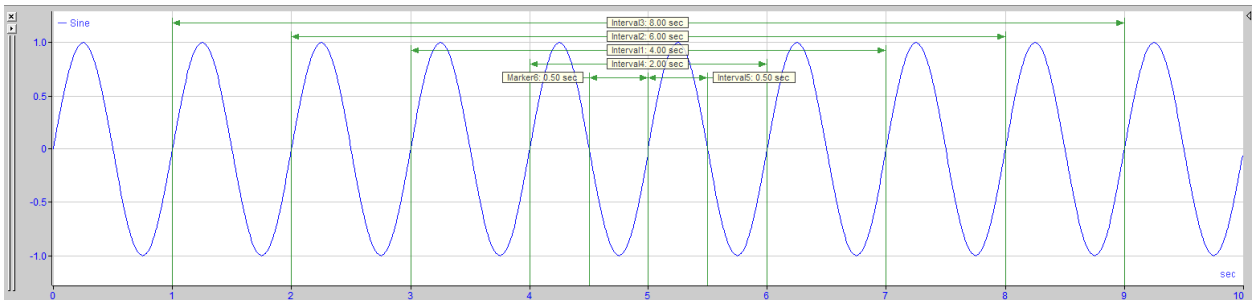
Important note, for intervals specified for the frequency and inverse length domain, the expressions for start and stop are also evaluated in the frequency and length domain. This is contrary to the marker expressions that are evaluated instead in the time and length domain respectively. If necessary, one can use the *ConvertBase* ibaAnalyzer function to force evaluation in the desired domain.

1.4 Drawing order

The intervals are drawn in the order that they are specified in the interval dialog, however if one interval entirely encompasses the other, it takes precedence.

ibaAnalyzer attempts to draw the interval as close as possible to the top of the graph, If there is no room to draw the horizontal line and the label (because of the legend or previously drawn intervals) a lower position is attempted. If the bottom of the graph is reached without a suitable location being found to draw the interval, this interval is not drawn; ibaAnalyzer will skip it and attempt to draw subsequent intervals or other elements.





The order in which elements on the graphs are drawn on top of the signals are:

1. The dynamic markers
2. The harmonic markers, including sidebands or harmonics if so specified.
3. The legend
4. Intervals
5. Static Markers
6. Text labels of text channels.

1.5 Shortcuts

The following keyboards are present to quickly insert markers or intervals.

- *Ctrl*-M: Insert an unnamed static marker at the X1 marker position. The marker will be visible in the selected graph, and invisible in all other graphs.
- *Ctrl*-*Alt*-M: Insert an unnamed static marker at the X2 marker position. The marker will be visible in the selected graph, and invisible in all other graphs.
- *Ctrl*-I: Insert an unnamed interval spanning from the X1 marker position to the X2 marker position.

If the keyboard shortcuts would insert a marker that is identical to an already existing marker, the new marker will not be inserted. Instead the visibility of the old marker will be toggled (i.e. if it was visible it will be hidden, if hidden it will be made visible).

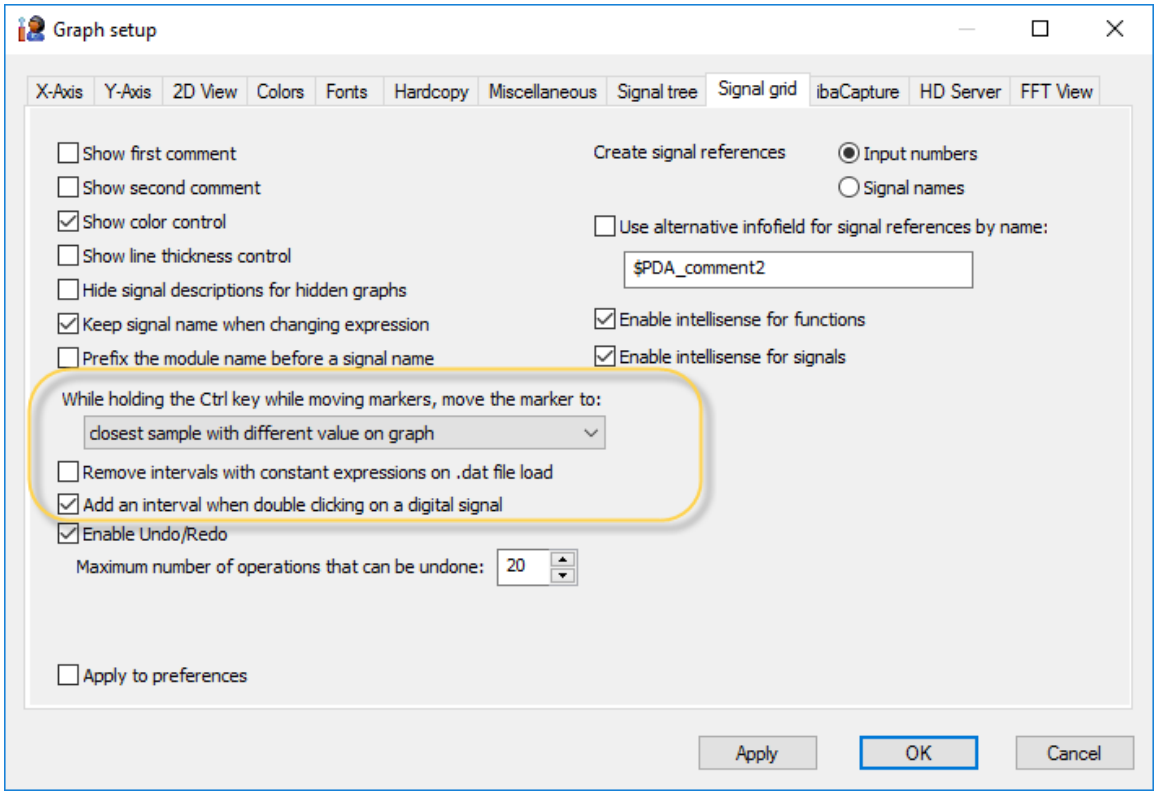
If the keyboard shortcuts would insert an interval that is identical to an already existing interval, then the new interval will not be inserted. Instead the old interval will be deleted if it was visible, and will be made visible if it was invisible.

Additionally, double clicking on a digital signal will insert an unnamed interval.

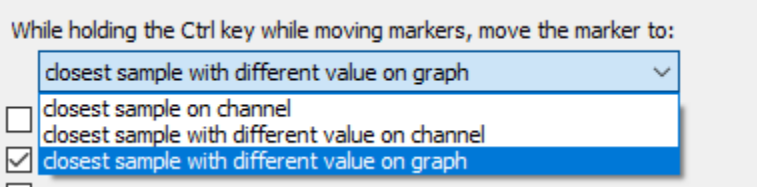
If the digital signal was TRUE at the location that was clicked, this interval will span from the rising edge in front of the location that was clicked until the falling edge after the location that was clicked. If the digital signal was FALSE at the location that was clicked, this interval will span from the falling edge in front of the location that was clicked until the rising edge after the location that was clicked. If the interval inserted this way would be identical to an already existing interval, the new interval will not be inserted. Instead the old interval will be deleted if it was visible, and will be made visible if it was invisible.

1.6 Preferences

A couple of preferences that influence how intervals and markers operate are present in the signal grid tab of the preferences and graph setup dialogs:


Graph setup
X-Axis
Y-Axis
2D View
Colors
Fonts
Hardcopy
Miscellaneous
Signal tree
Signal grid
ibaCapture
HD Server
FFT View
Show first comment
Show second comment
Show color control
Show line thickness control
Hide signal descriptions for hidden graphs
Keep signal name when changing expression
Prefix the module name before a signal name
Create signal references
Input numbers
Signal names
Use alternative infofield for signal references by name:
$PDA_comment2
Enable intellisense for functions
Enable intellisense for signals
While holding the Ctrl key while moving markers, move the marker to:
closest sample with different value on graph
Remove intervals with constant expressions on .dat file load
Add an interval when double clicking on a digital signal
Enable Undo/Redo
Maximum number of operations that can be undone:
20
Apply to preferences
Apply
OK
Cancel

- In previous versions of ibaAnalyzer, holding the *Ctrl* key while moving markers (either by dragging them or pressing the arrow keys), caused the markers to move from sample to sample on the selected signal. This behavior is still possible, however two new options are available:
  - *Move to the closest sample with different value on graph:* This option will consider for every signal in the graph the closest sample that has a different value than the sample at the current marker position for that signal. From these candidates the closest sample to the current marker position is selected to move the marker to.
  - *Move to the closest sample with on channel*: This option will only consider the currently selected signal. It will select the closest sample that has a different value than the sample at the current marker position for that signal.



  The default behavior is "*Move to the closest sample with different value on graph*".

  Note that for digital signals, moving the markers with *Ctrl* and one of the above options enabled, corresponds to moving the markers to the rising and falling edges of the signal.

  This option is stored in the analysis and is available in the preferences to set the default for new analyses.

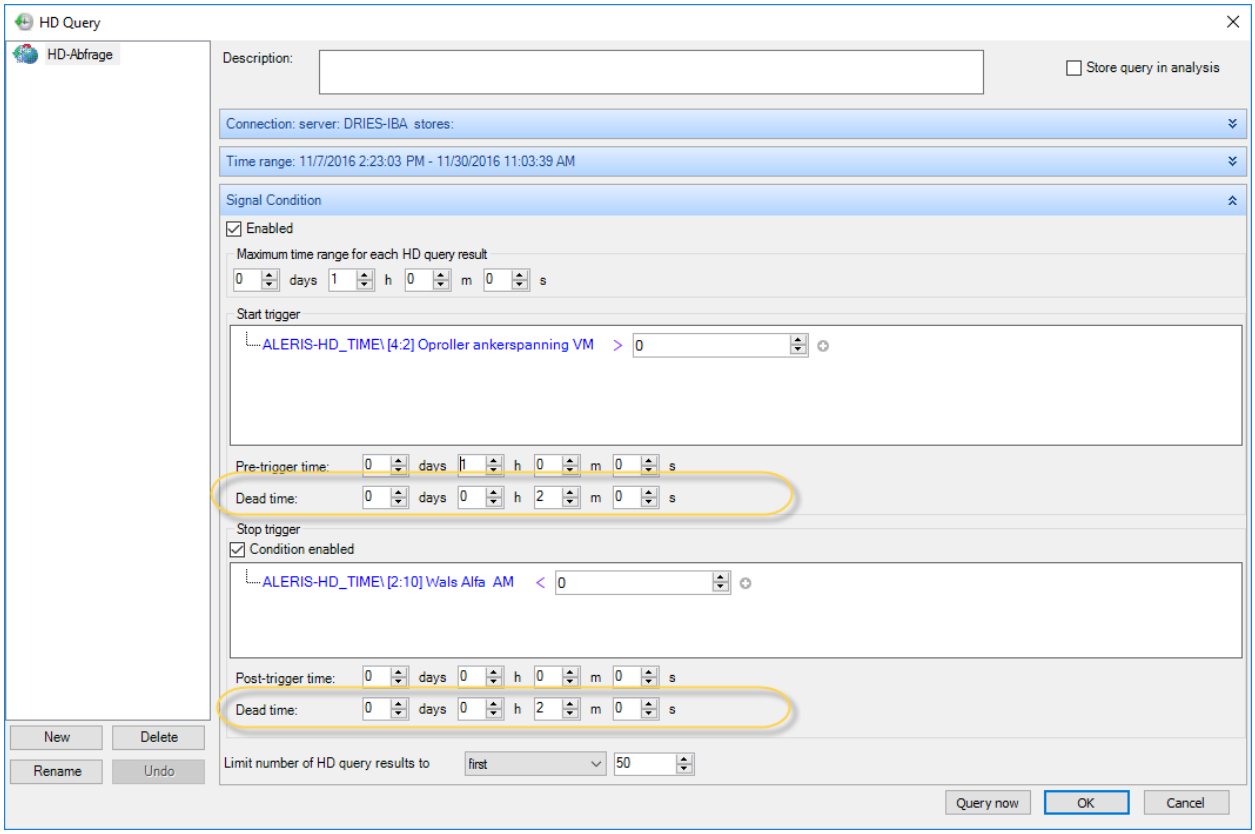
- *Remove intervals with constant expressions on .dat file load*: This removes intervals where the start and stop expressions are pure constants when a new `.dat` file is loaded.
  Intervals inserted by pressing the *"Insert from markers"* button in the dialog or by using the keyboard shortcut or by double clicking on a digital signal are such intervals with constant expressions. Intervals inserted this way are likely no longer relevant if a new `.dat` file is loaded, hence enabling this option causes them to be removed automatically. Intervals for which the start or stop expressions contain a function or mathematical operation are considered to be dependent on the loaded data and are hence retained. By default this option is enabled. This option is stored in the analysis and is available in the preferences to set the default for new analyses.
- *Add an interval by double clicking on a digital signal.* You can uncheck this if you do not want that an interval is added when double clicking on a digital signal by accident. By default this option is enabled. This is a global preferences option stored in the ibaAnalyzer registry settings for each user; it is not stored in the analysis.

2 HD Queries

2.1 Dead times on conditions

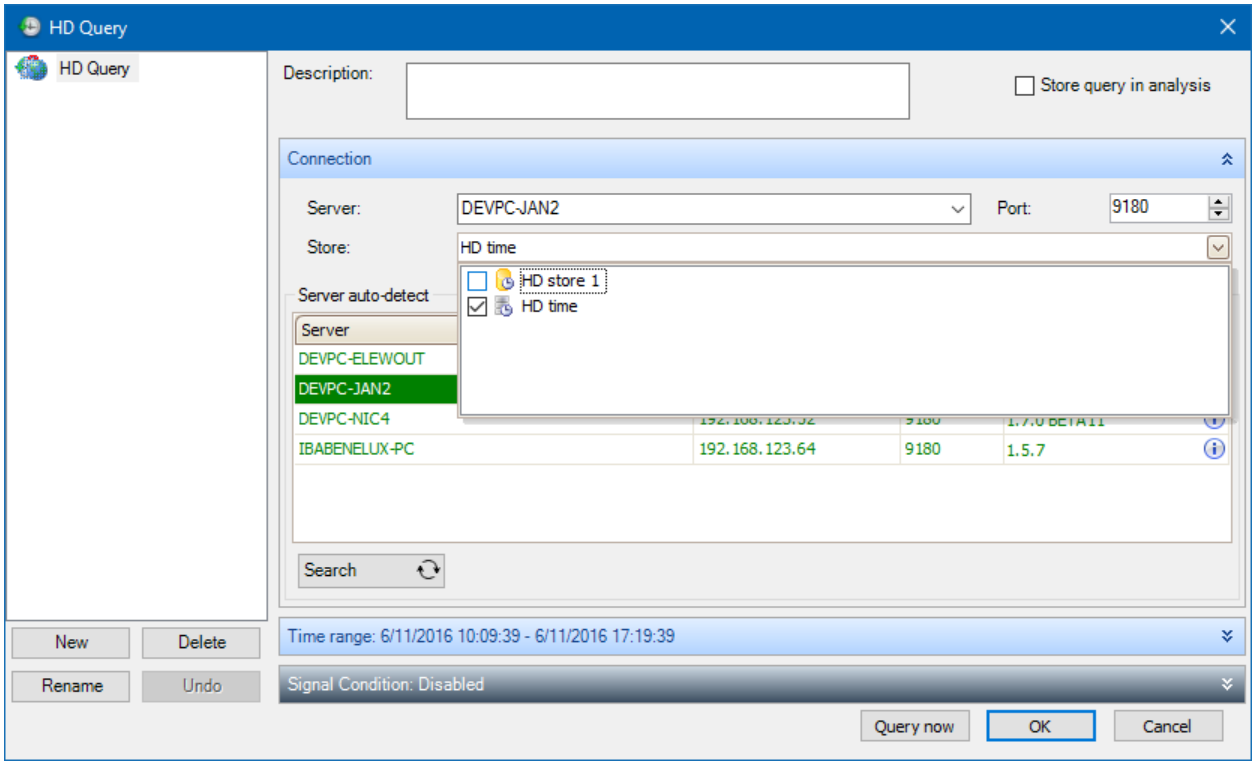
Because of noise on the signals, it can happen that a signal used in a start or stop condition satisfies the specified threshold, than shortly does not satisfy the threshold any longer, followed by a longer interval that the threshold is satisfied. This causes the condition to be satisfied two (or more) times while the second occurrence should not actually be considered a different occurrence than the initial one. When such query is executed and this happens in the start condition, spurious results are returned in the population tree. When this happens in the stop condition, a result can be terminated to early by the stop condition of a previous result.

To address this, dead times are introduced on the conditions. A start condition will not be considered satisfied while still in the specified dead time after the start time of a previous result. Likewise a stop condition will not be considered satisfied while still in the specified dead time after the stop of a previous result.

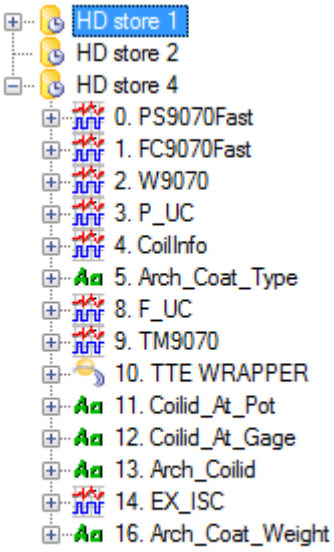


2.2 Various changes

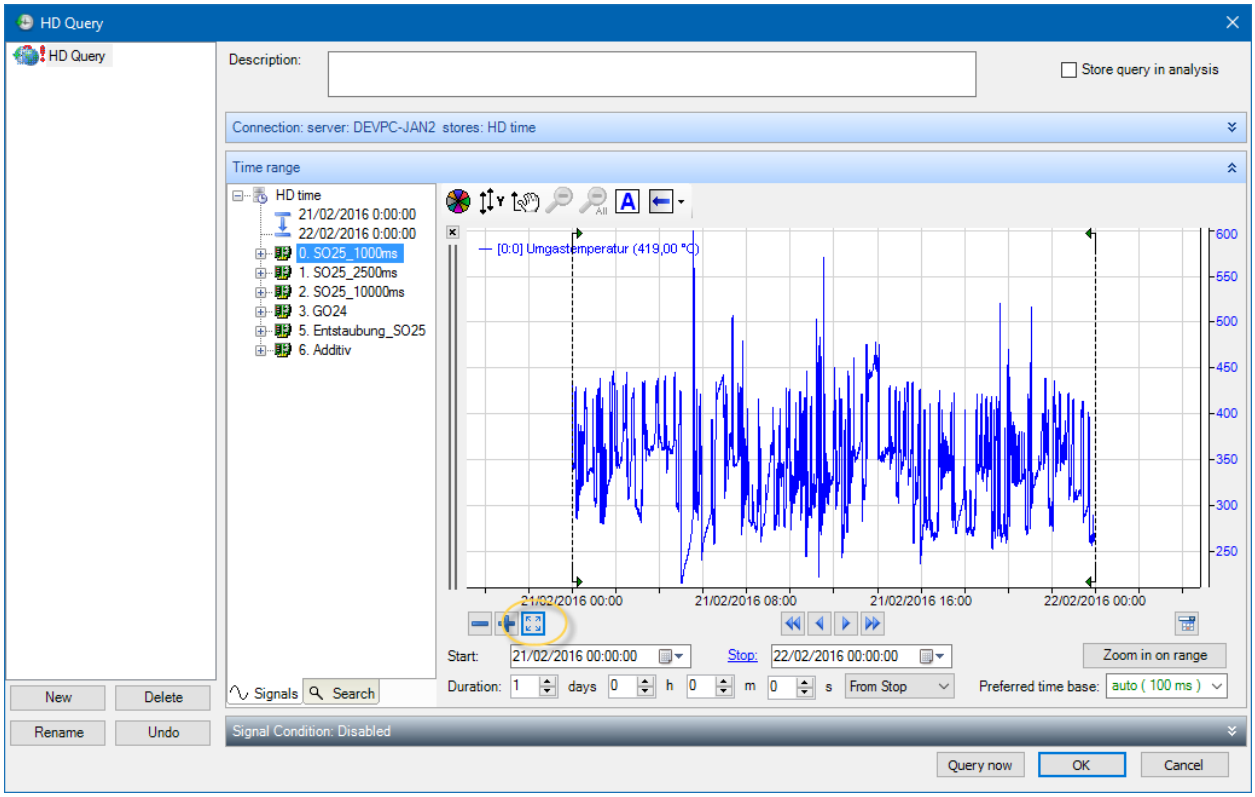
Mounted HD stores are displayed in the HD query dialog store drop-down list



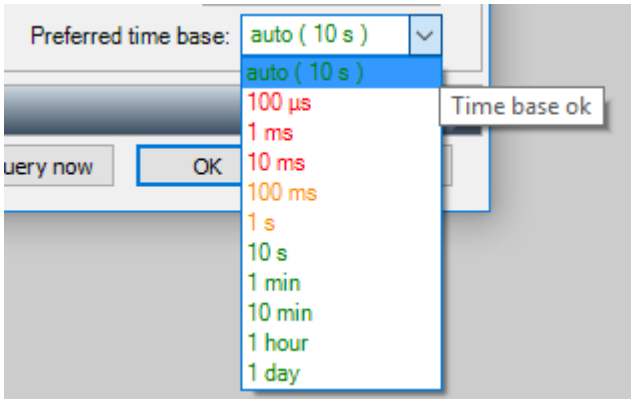
The ibaAnalyzer signaltree icons for an HD query result now correspond with the icons used in the HD dialog and ibaPDA.


HD store 1
HD store 2
HD store 4
0. PS9070Fast
1. FC9070Fast
2. W9070
3. P_UC
4. CoilInfo
5. Arch_Coat_Type
8. F_UC
9. TM9070
10. TTE WRAPPER
11. Coilid_At_Pot
12. Coilid_At_Gage
13. Arch_Coilid
14. EX_ISC
16. Arch_Coat_Weight

Auto zoom button on HD query dialog zooms out the X-axis to the raw data range of the displayed signals and auto-scales Y-axes



More options for ibaHD preferred time have been added.



3 Function improvements

3.1 XFirst and XLast functions

New parameters have been added to the *XFirst* function so that rather than the first X-axis coordinate for which an event occurs; also subsequent reoccurrences can be reported. Similarly the *XLast* function has been extended in the same way, but rather than searching from the start of the signal, the events are searched from the end of the signal in reverse order.

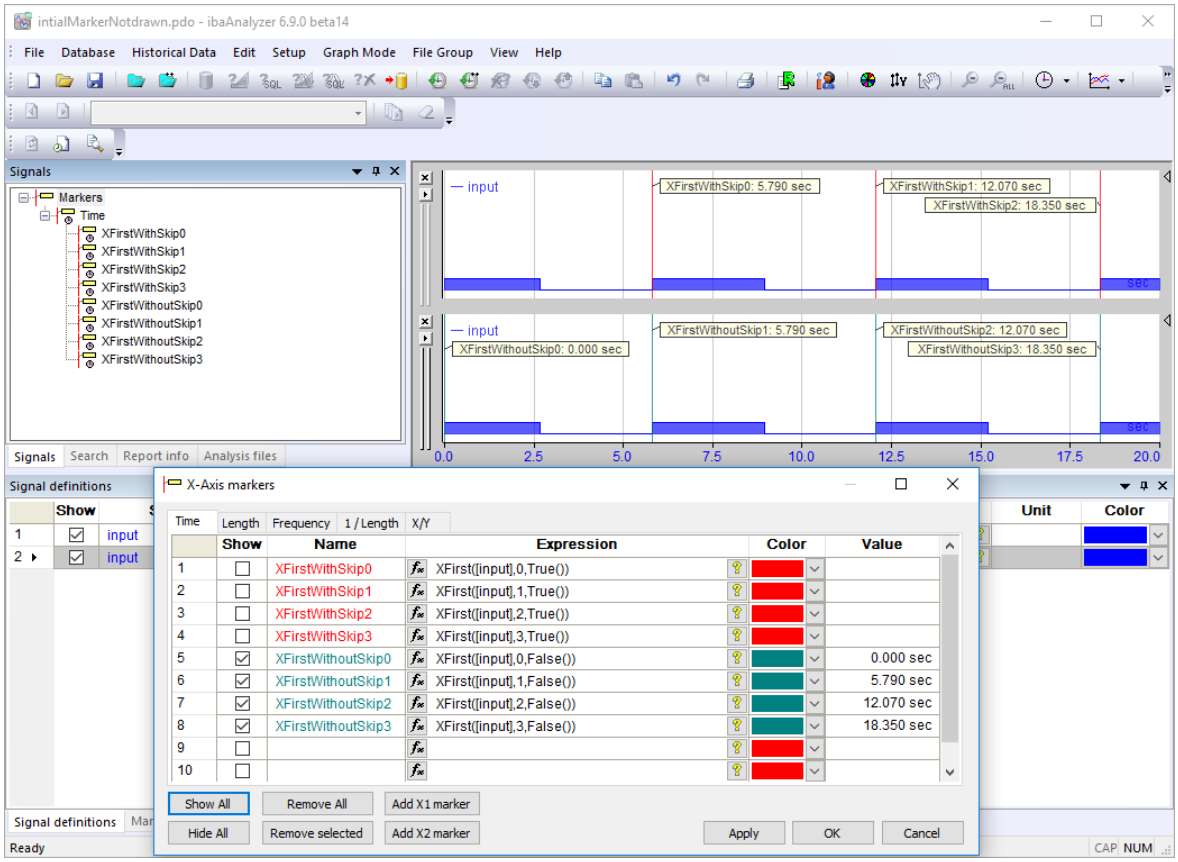
The *XFirst* function now takes the following parameters:

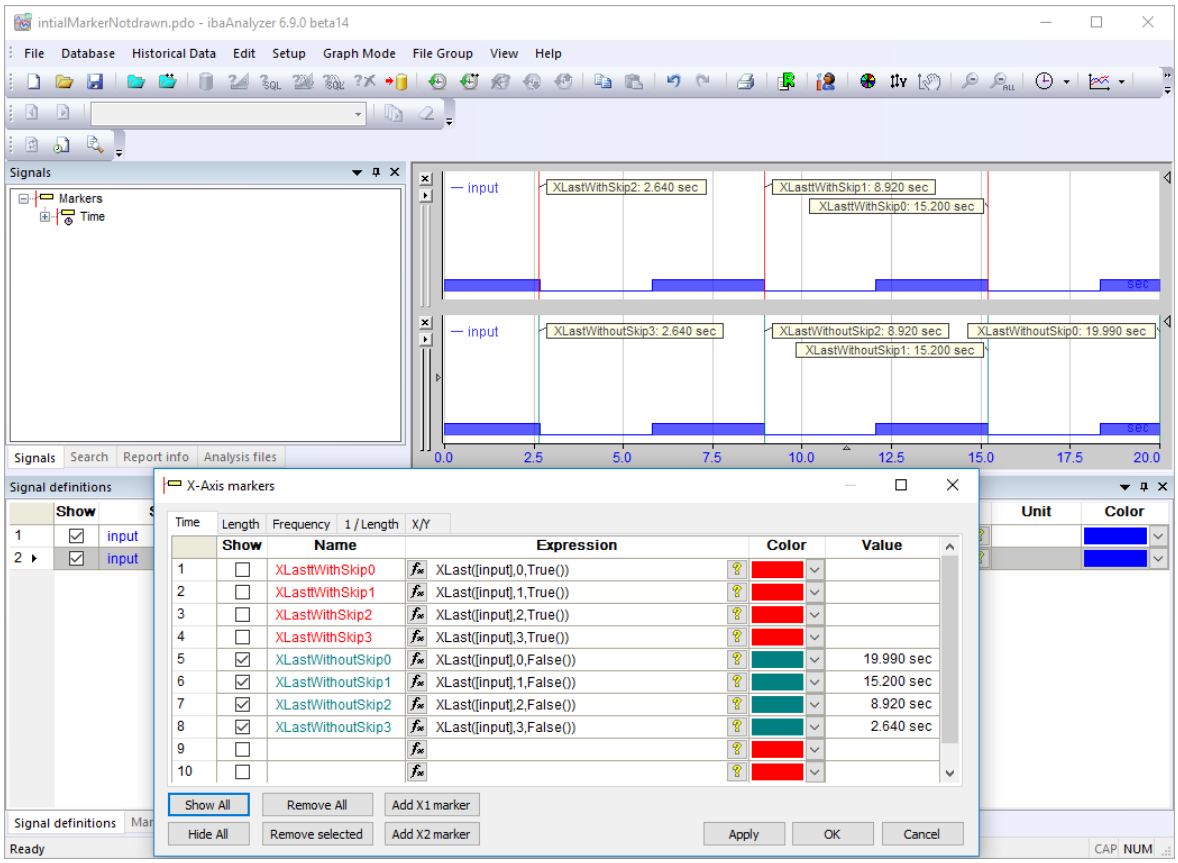
- Expression: A logical expression (i.e. it evaluates to either TRUE or FALSE) for which the first or alternatively a subsequent occurrence of the condition being TRUE (rising edge) needs to be searched.
- Skip: Optionally, the number of rising edges in Expression that need to be skipped. By default this parameter is zero and no rising edges are skipped.
- SkipInitialEdge: Determines whether or not the signal being initially TRUE is also considered as a rising edge. This parameter has no effect if the signal is not initially TRUE. If this parameter is TRUE and Expression is initially TRUE, the initial part of expression is skipped and only subsequent rising edges in Expression are considered. If this parameter is FALSE and Expression is initially TRUE, the initial sample will be reported if Skip is zero or the number of rising edges after the initial signal that will be skipped will be one less than *Skip*. This parameter is optional, if omitted, it is FALSE by default.

The *XLast* function now takes the following parameters:

- Expression: A logical expression (i.e. it evaluates to either TRUE or FALSE) for which the last or alternatively a previous occurrence of the condition being TRUE (falling edge) needs to be searched.
- Skip: Optionally, the number of falling edges in Expression that need to be skipped. By default this parameter is zero and no falling edges are skipped.
- SkipFinalEdge: Determines whether or not the signal being TRUE for the last sample is also considered as a falling edge. This parameter has no effect if the signal is not TRUE for the last sample. If this parameter is TRUE and Expression is TRUE for the last sample, the last part of expression is skipped and only preceding falling edges in Expression are considered. If this parameter is FALSE and Expression is TRUE for the last sample, this sample will be reported if *Skip* is zero or the number of falling edges before the last signal that will be skipped will be one less than *Skip*. This parameter is optional, if omitted, it is FALSE by default.

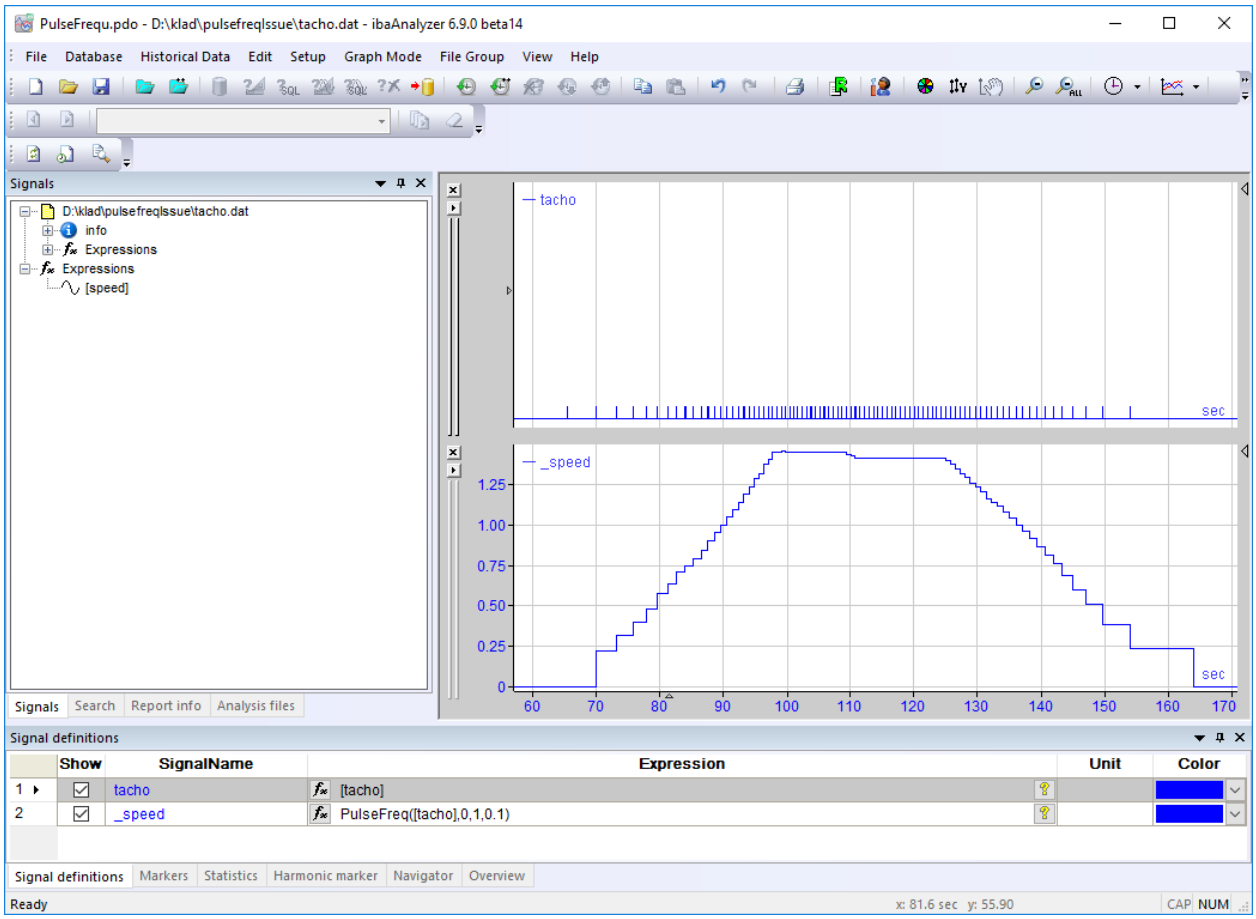
We envision one important usage of the *XFirst* and *XLast* functions (with the new additional parameters) will be to have Markers and Intervals that adapt to the loaded data.




intialMarkerNotdrawn.pdo - ibaAnalyzer 6.9.0 beta14
File Database Historical Data Edit Setup Graph Mode File Group View Help
Signals
Markers
Time
input
XLastWithSkip2: 2.640 sec
XLasttWithSkip1: 8.920 sec
XLasttWithSkip0: 15.200 sec
input
XLastWithoutSkip3: 2.640 sec
XLastWithoutSkip2: 8.920 sec
XLastWithoutSkip0: 19.990 sec
XLastWithoutSkip1: 15.200 sec
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5 20.0
Signals Search Report info Analysis files
Signal definitions
Show
Unit
Color
1 input
2 input
X-Axis markers
Time Length Frequency 1 / Length X/Y
Show Name Expression Color Value
1 XLasttWithSkip0 XLast([input],0,True())
2 XLasttWithSkip1 XLast([input],1,True())
3 XLastWithSkip2 XLast([input],2,True())
4 XLastWithSkip3 XLast([input],3,True())
5 XLastWithoutSkip0 XLast([input],0,False()) 19.990 sec
6 XLastWithoutSkip1 XLast([input],1,False()) 15.200 sec
7 XLastWithoutSkip2 XLast([input],2,False()) 8.920 sec
8 XLastWithoutSkip3 XLast([input],3,False()) 2.640 sec
9
10
Show All
Remove All
Add X1 marker
Hide All
Remove selected
Add X2 marker
Apply
OK
Cancel
Signal definitions
Ready
CAP NUM

3.2 PulseFreq function

In previous versions of ibaAnalyzer, the *PulseFreq* function would report 0 Hz (i.e. stationary state) when the input signal stayed constant for 1000 consecutive samples. In the current version of ibaAnalyzer, rather than using this hardcoded number of samples, a minimum frequency can be provided as last optional parameter. From this frequency (along with the signal sample rate) the maximum number of consecutive samples before *PulseFreq* reports 0 Hz can be calculated. This parameter can be omitted, in which case the minimum frequency of 0.05 Hz is used.



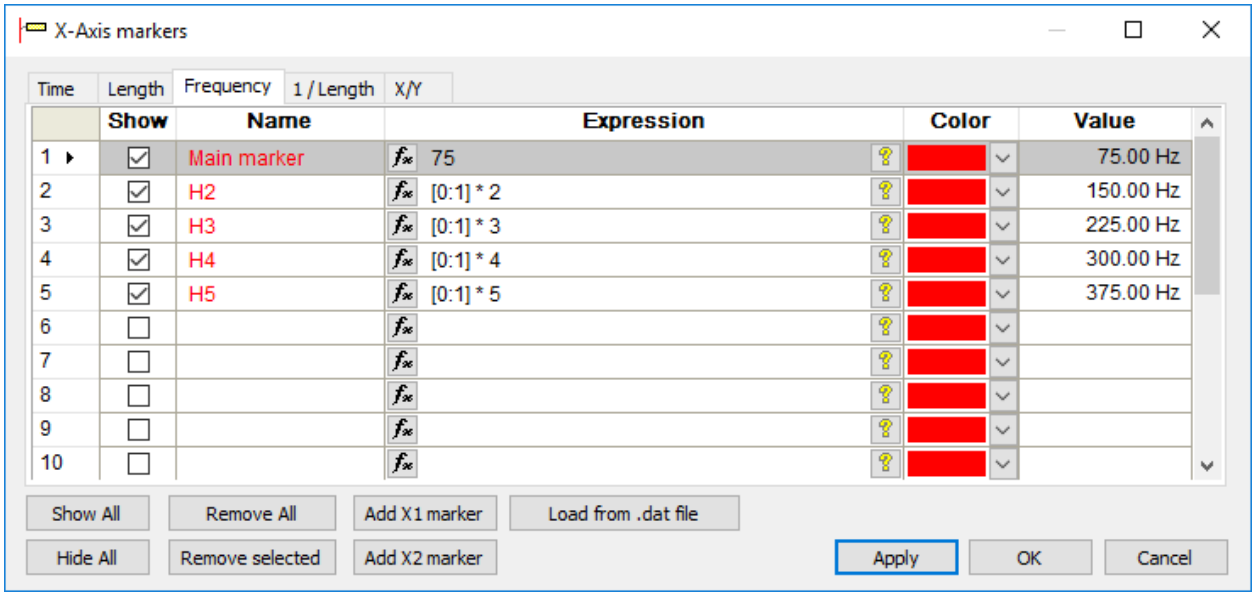
4 Miscellaneous feature

4.1 Inserting markers from InSpectra

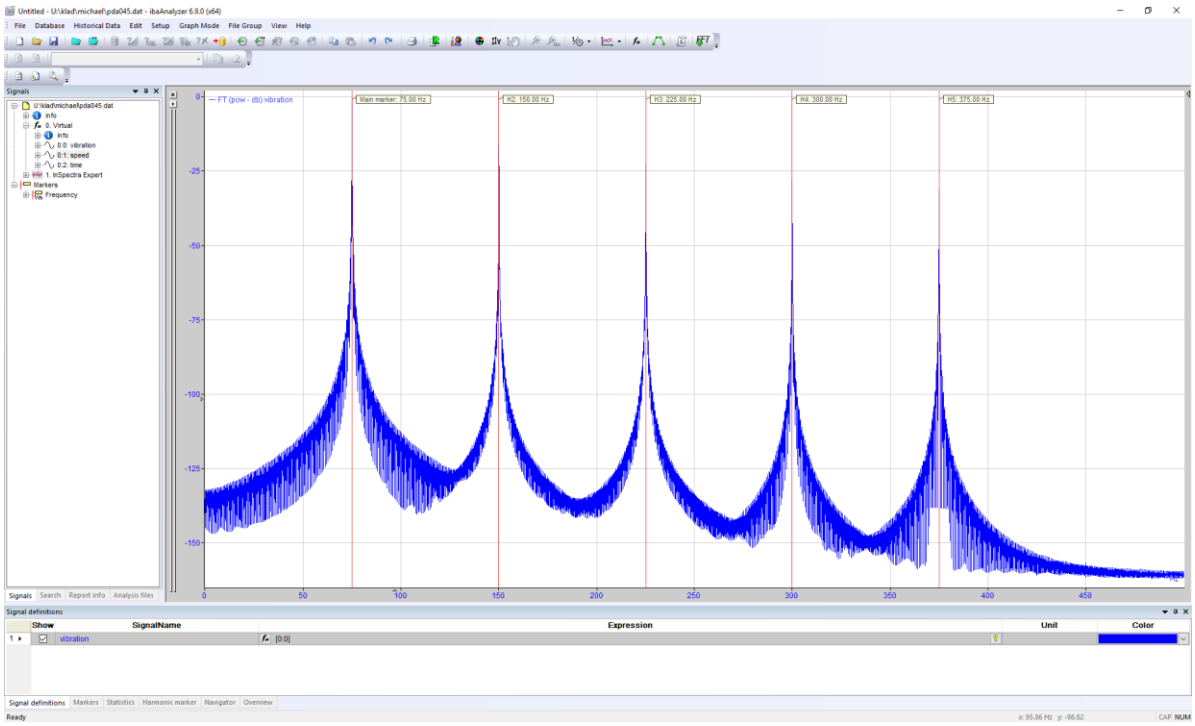
From version 6.37.0 on, ibaPDA will write the Inspectra marker expressions as hidden infofields in the .dat files it generates. These markers can be imported as ibaAnalyzer static markers.

Important note, the signals used in the inspectra marker expressions must also be written to the .dat file in order for the ibaAnalyzer marker expressions to evaluate properly.

The markers can be imported by clicking the button labeled “Load from .dat file”. This button is not present if there are no Inspectra marker expressions present in the .dat file.



By default the visibility column is not checked, one can check individually the markers one wishes to display.


Signals
Signal definitions
Show
SignalName
Expression
Unit
Color
vibration