# New Features in ibaAnalyzer 6.7.0

Author: Michael Verschaeve

Date: 21 January 2016

# Table of contents

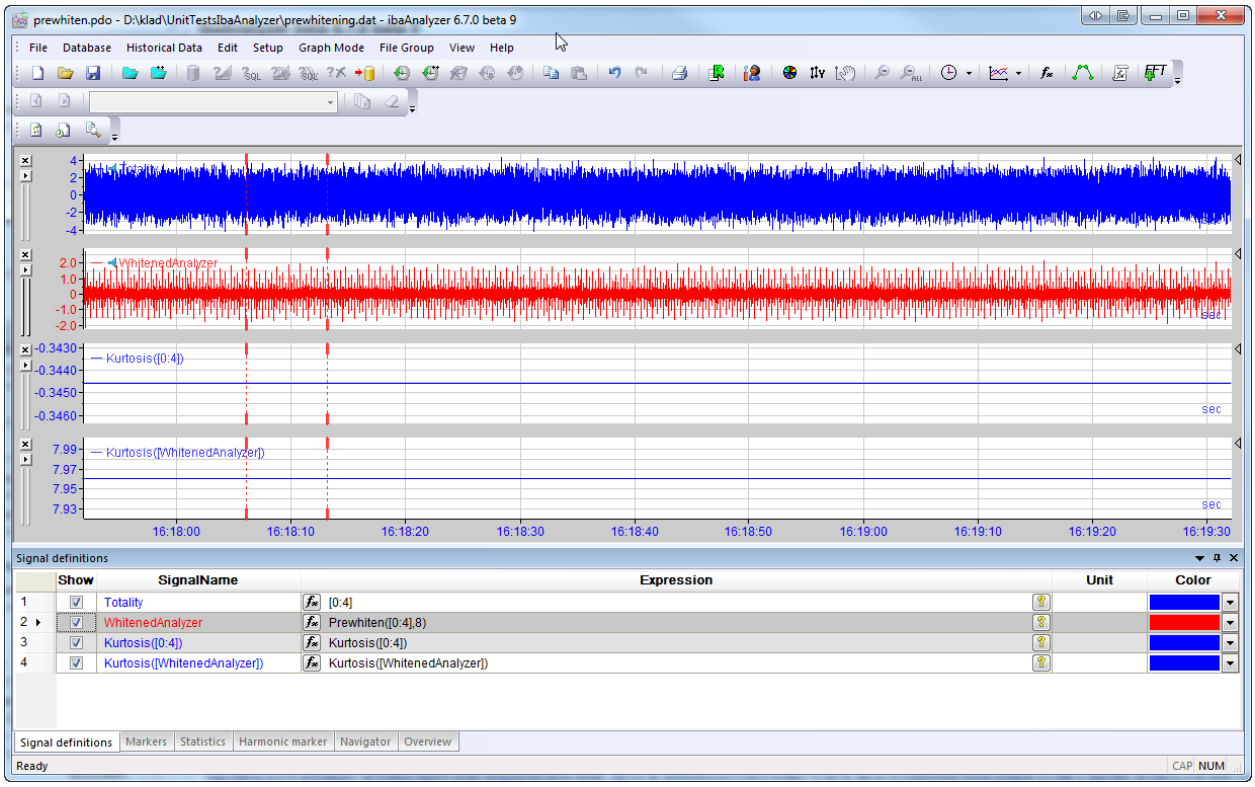
# 1 New functions

## 1.1 Prewhiten function

This function applies a FIR filter with the coefficients determined by the Yule-Walker equations. The filter is a high pass filter, leaving only white noise and impulse components of the signal.

The function takes the following arguments:

- Expression: Signal to filter.
- Order: The order of the FIR filter that is applied.

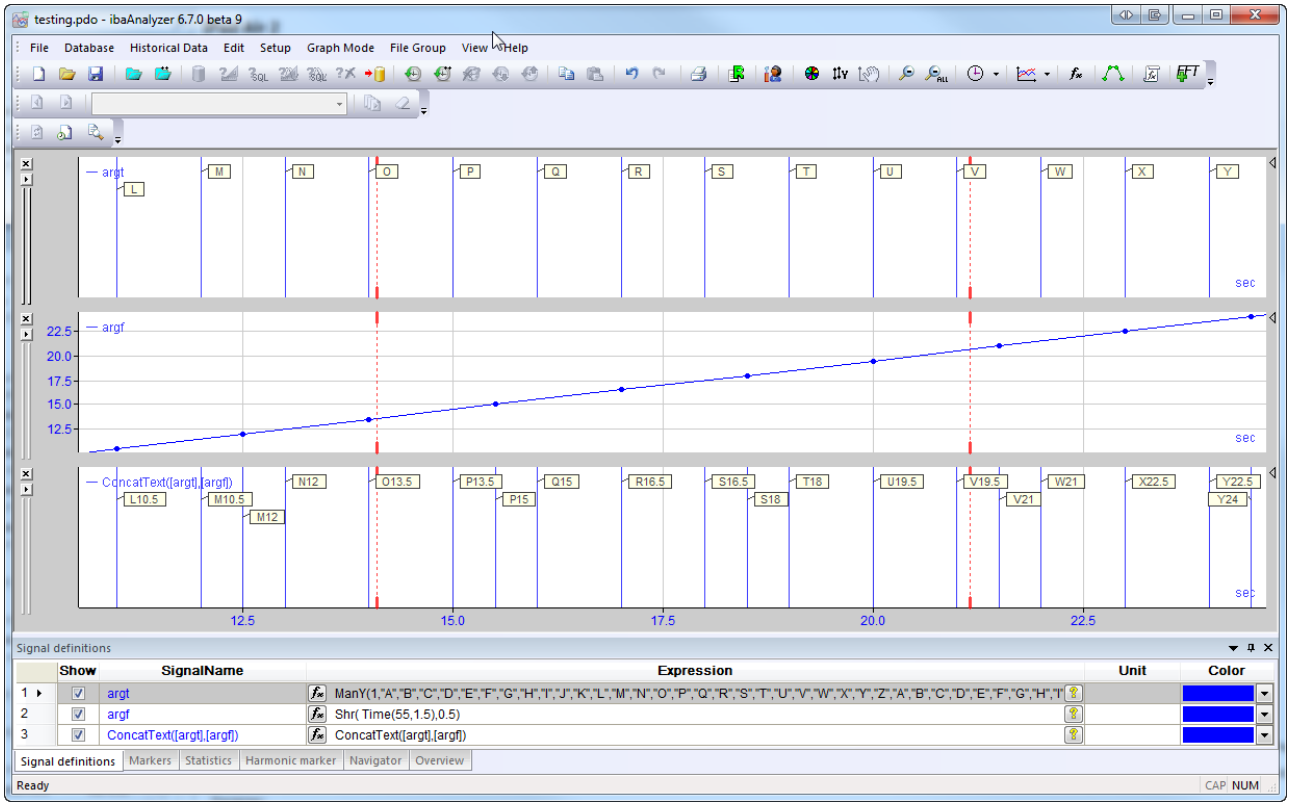



## 1.2 ConcatText function

This function concatenates text channels. The function can take a variable number of arguments so multiple text channels can be concatenated at once.

If the samples of the text channels do not coincide, at each sample of each channel a text will be inserted. For the channels with the missing sample at that timestamp, the sample immediately to the left will be used (if any).

If one of the arguments is a numeric channel rather than a text channel, it will be first converted to a text channel, the result is the same if the *ToText* version were to be applied first to the numeric channel (with default text formatting).




## 1.3 Switch function

This function is similar to the CASE statement in SQL. A *selector* expression is evaluated and compared with one or more *case* expressions that have corresponding *value* expressions. The *value* expression for which the corresponding *case* expression equals the *selector* expression is returned.

Syntax:

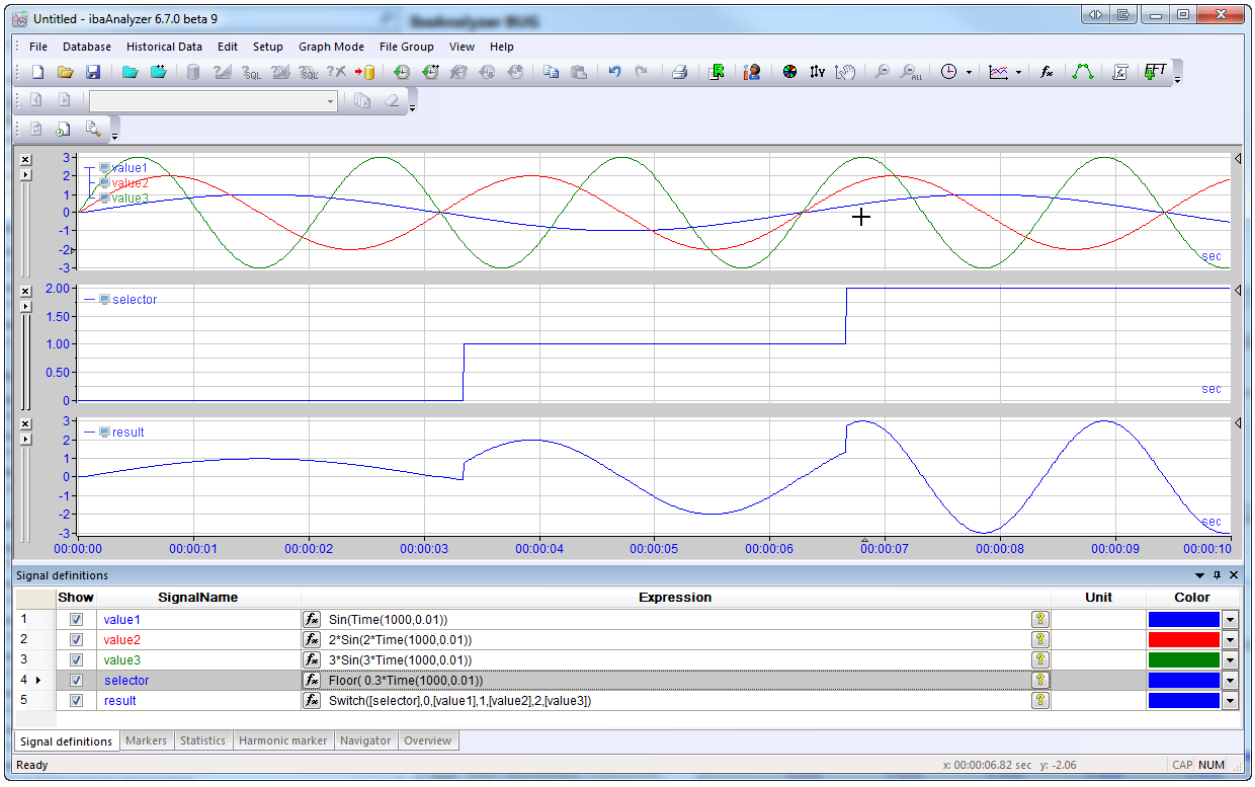
```
Switch(selector expression, case_1 expression, value_1 expression, case_2
expression, value_2 expression, . . . , case_n expression, value_n
expression, default value expression (optional) )
```

A minimum of three arguments needs to be present.
If an even number of arguments are present, the last expression is the *default* expression which will be used if none of the *case* expressions match the *selector* expression.
If an odd number of arguments are present, there is no default expression and if there are samples where none of the *case* expressions match the *selector* expression, a hole will be inserted.
The *case* and *value* expressions need not contain valid data. If a *case* expression is invalid, it will never match the *selector* expression. If a *value* expression is invalid and the corresponding *case* expression matches, holes will be inserted.
If multiple *case* expressions match the *selector* expression, the *case* expression that appears first in the list of arguments has precedence, its corresponding *value* expression will be returned.

The *selector* expression can be a one of the following:

- A numeric constant.
- A text constant.
- A numeric equidistantly sampled channel.
- A numeric non-equidistantly sampled channel.
- A text channel.

In case of the *selector* expression being a numeric constant, only numeric *case* expressions can match. If the *case* expression is not constant itself, the switch function will average the signal before comparing it with the selector.
In case of the *selector* signal being a text constant, only text *case* expressions can match. If the case expression is not constant itself, only the first string in the channel will be compared with the *selector* expression.
In the above two cases, the resulting signal will be a copy of one of the value expressions, or return a blank signal if no cases match or the default signal if no cases match and a default signal is present.
In the other cases, the *case* expressions are expected to be of the same kind as the *selector* expression, if not, any *case* expression that is of a different kind will be considered an invalid expression and hence the corresponding *value* expression will never be considered.
If the *selector* expression is not constant, the resulting signal will not necessarily be a copy of one of the *value* expressions; the resulting signal can contain parts of different *value* expressions, corresponding with where the changing *selector* expression matches different *case* expressions.



# 2 Report generator

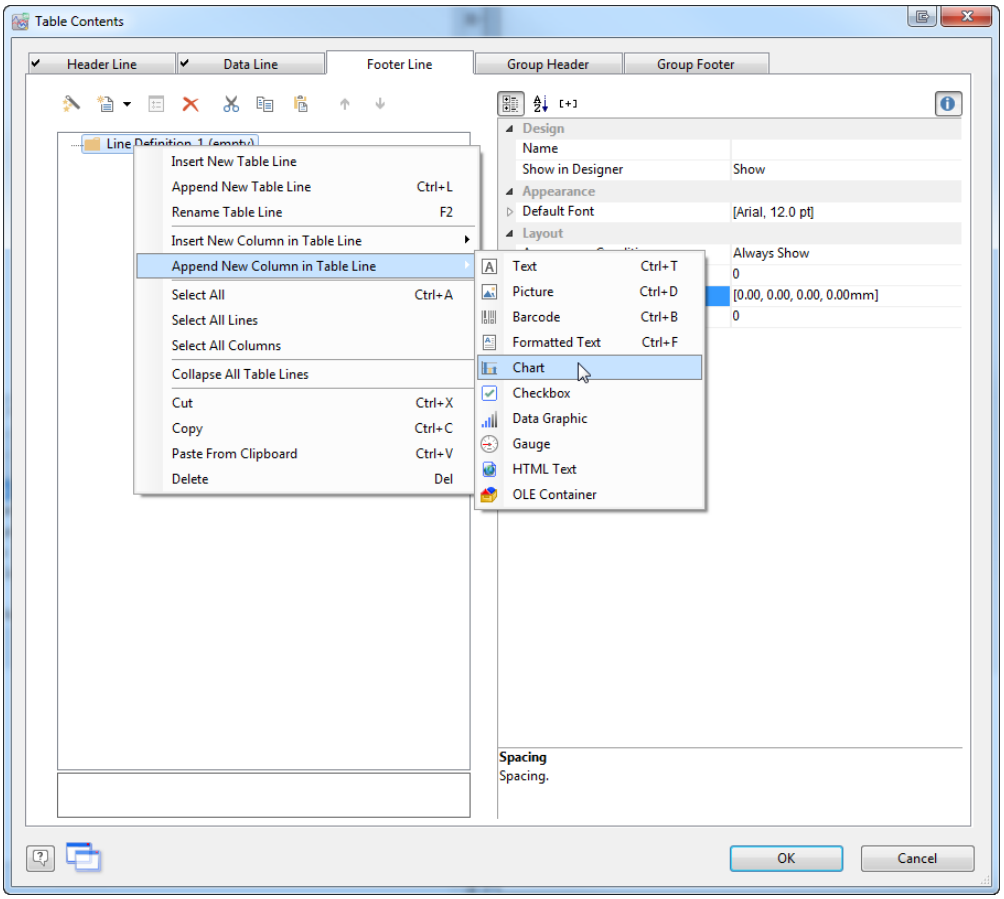
## 2.1 New List and Label version

The Combit List & Label software integrated in the ibaAnalyzer report generator has been updated to version 20 from version 15.

Any report generated in previous versions of ibaAnalyzer (L&L v 15) should be fully compatible with the new version. However, any new report generated in the current version of ibaAnalyzer, will no longer be compatible with previous versions of ibaAnalyzer.

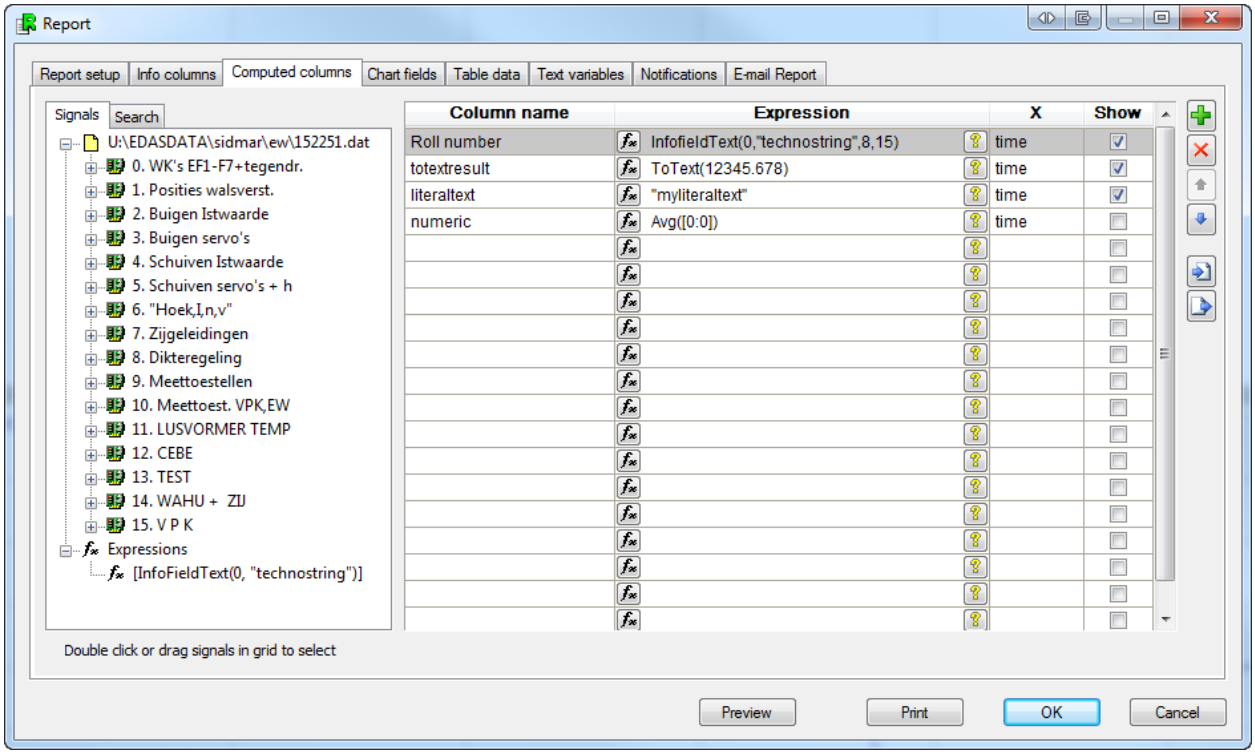
Some major changes are:

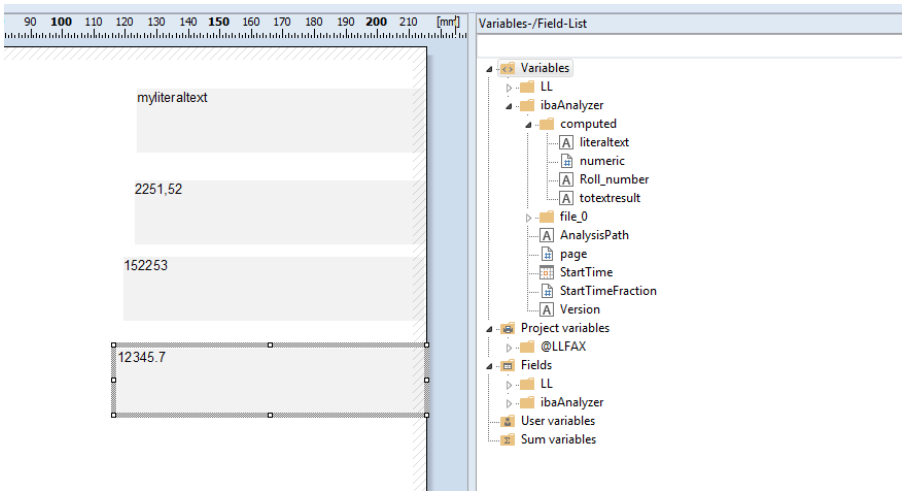
- During installation, the Combit List & Label libraries are no longer copied in to the windows system folder. They are currently copied in the ibaAnalyzer installation folder.
- Charts can no longer be inserted as global objects into the designer. They need to be placed in a header, footer or data line of a table. There are however workarounds:
  - A report created in a previous version of ibaAnalyzer with global chart objects will still have the objects global when opened in the current version.
  - A global chart object can be copy-pasted in a new project from a previous project.



## 2.2 Text computed columns

Besides the “Info columns” and “Text variables” tabs in the Report dialog, it is now also possible to pass texts to the rapport generator through the “Computed columns” tab. This allows for the results of ibaAnalyzer text functions (e.g. *InfoFieldText*, *ToText*, *ConcatText*, …) to be used in reports. If the result is a text channel with multiple texts, only the first text in the channel is passed.



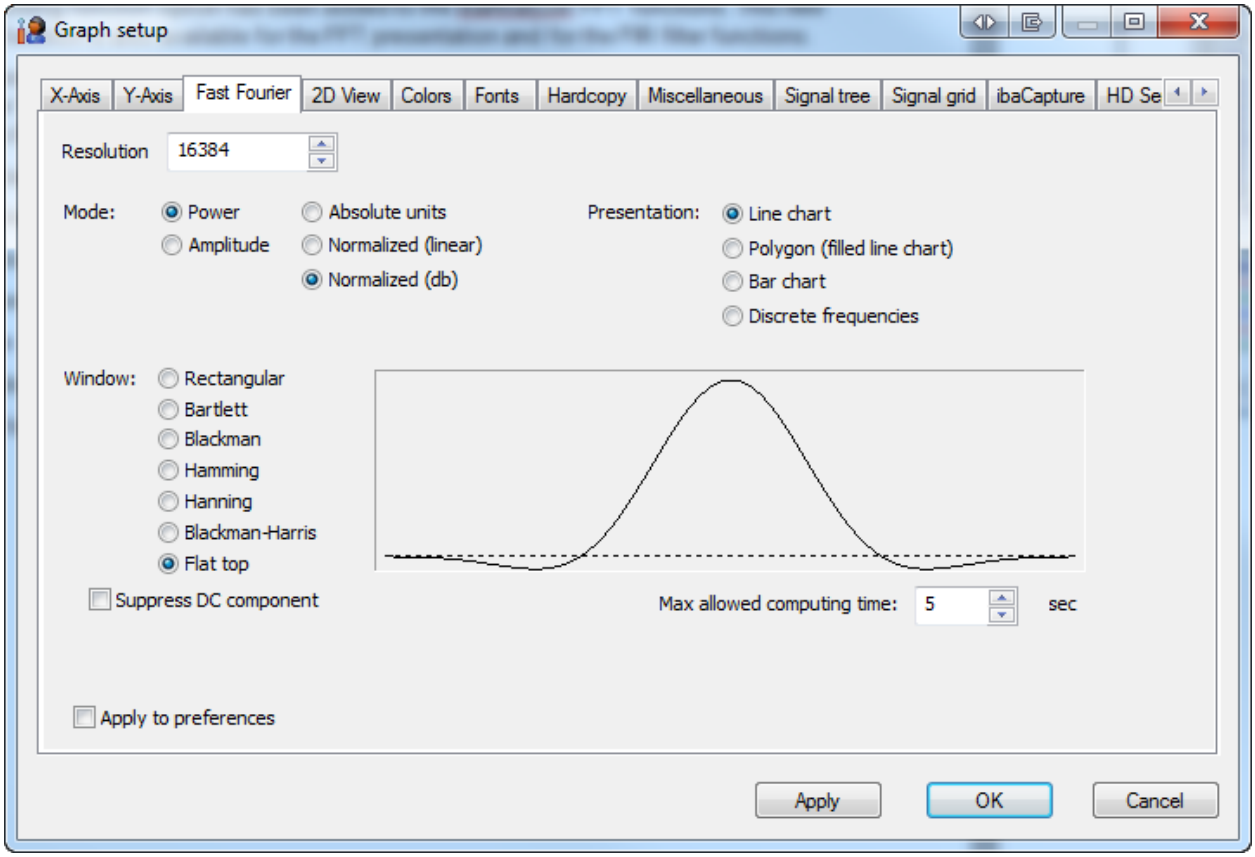
# 3 Miscellaneous features

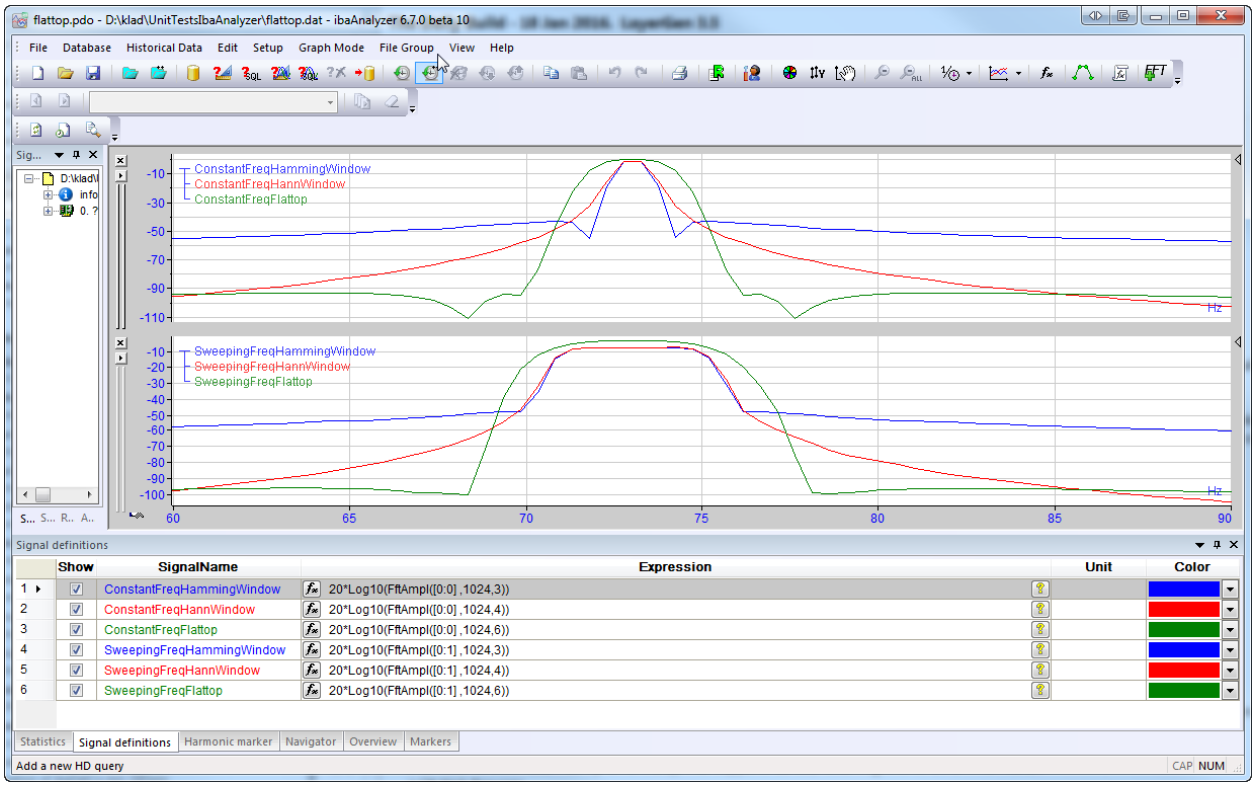
## 3.1 Flat top window

A new windowing function option has been added to the ibaAnalyzer FFT functions. This new windowing function is also available for the FFT presentation and for the FIR filter functions.

The windowing function is specified in an ibaAnalyzer function by specifying the integer 6 as the value for the “Window” parameter. For reference here is the list of current window functions that can be specified for the “Window” parameter:

0. Rectangular
1. Bartlett
2. Blackman
3. Hamming
4. Hann
5. Blackman Harris
6. Flat top



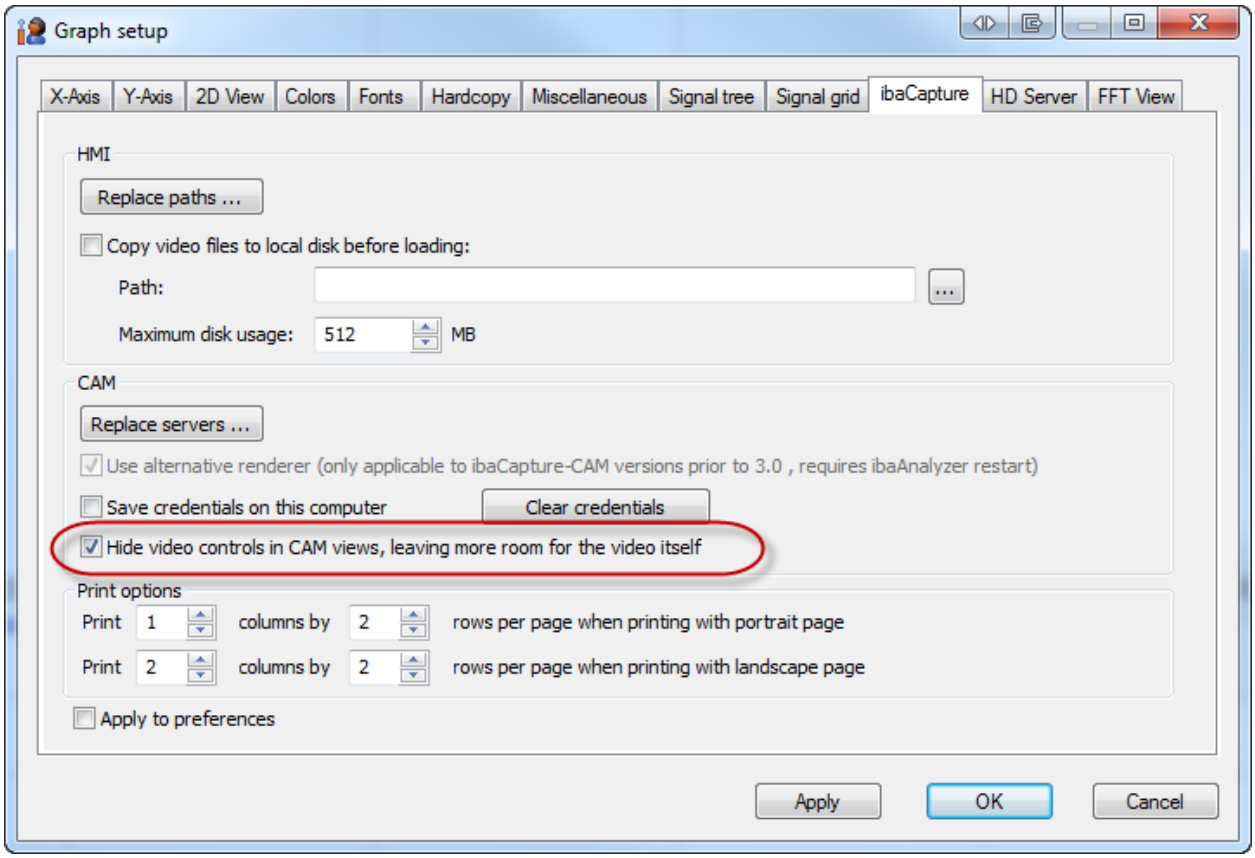


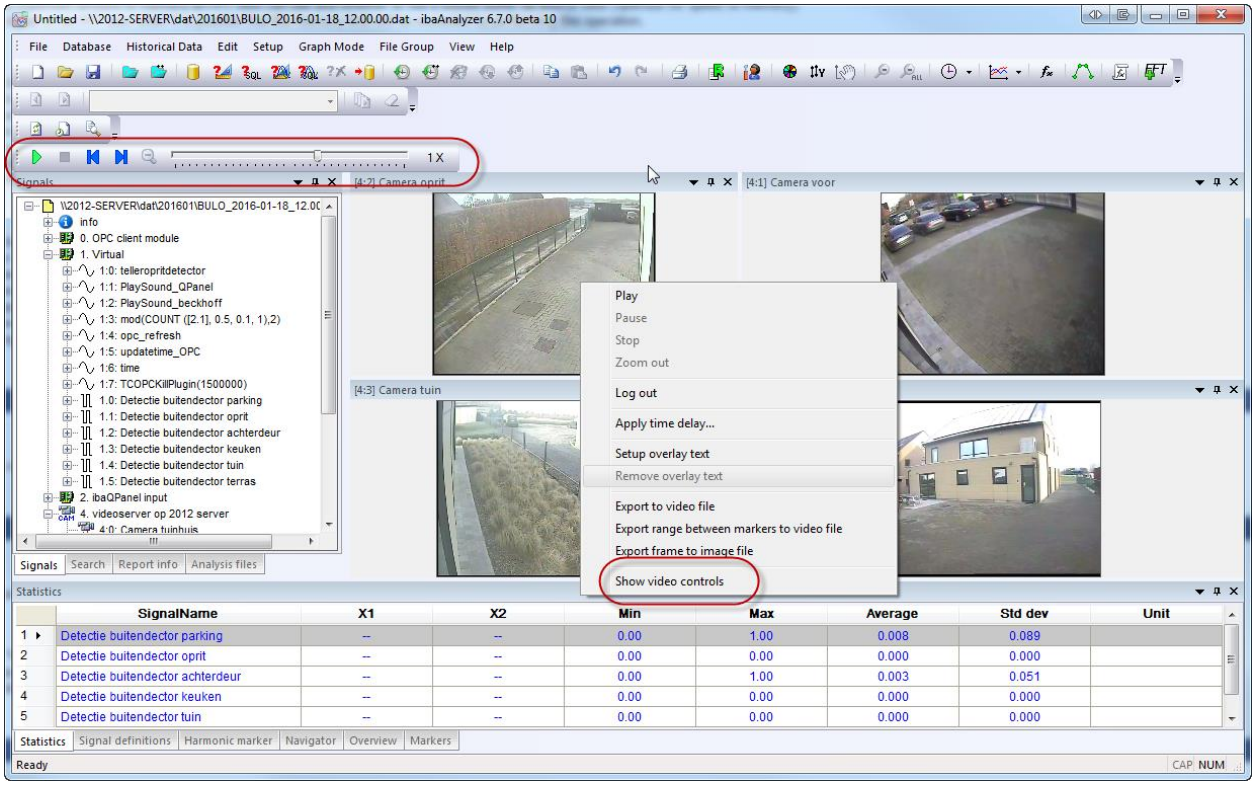

## 3.2 ibaCapture-CAM controls

When multiple ibaCapture-CAM videos are present, the controls to operate them (play/pause, stop, zooming and video replay speed) are repeated for each video. This is redundant and takes a lot of room on the screen. In the current version of ibaAnalyzer, it is possible to hide the controls. The videos can then still be operated from buttons appearing in the toolbar.

One can enable or disable this option from the ibaAnalyzer setup or preferences dialog.

One can also enable or disable this option from the context menu that appears when one right clicks a video.

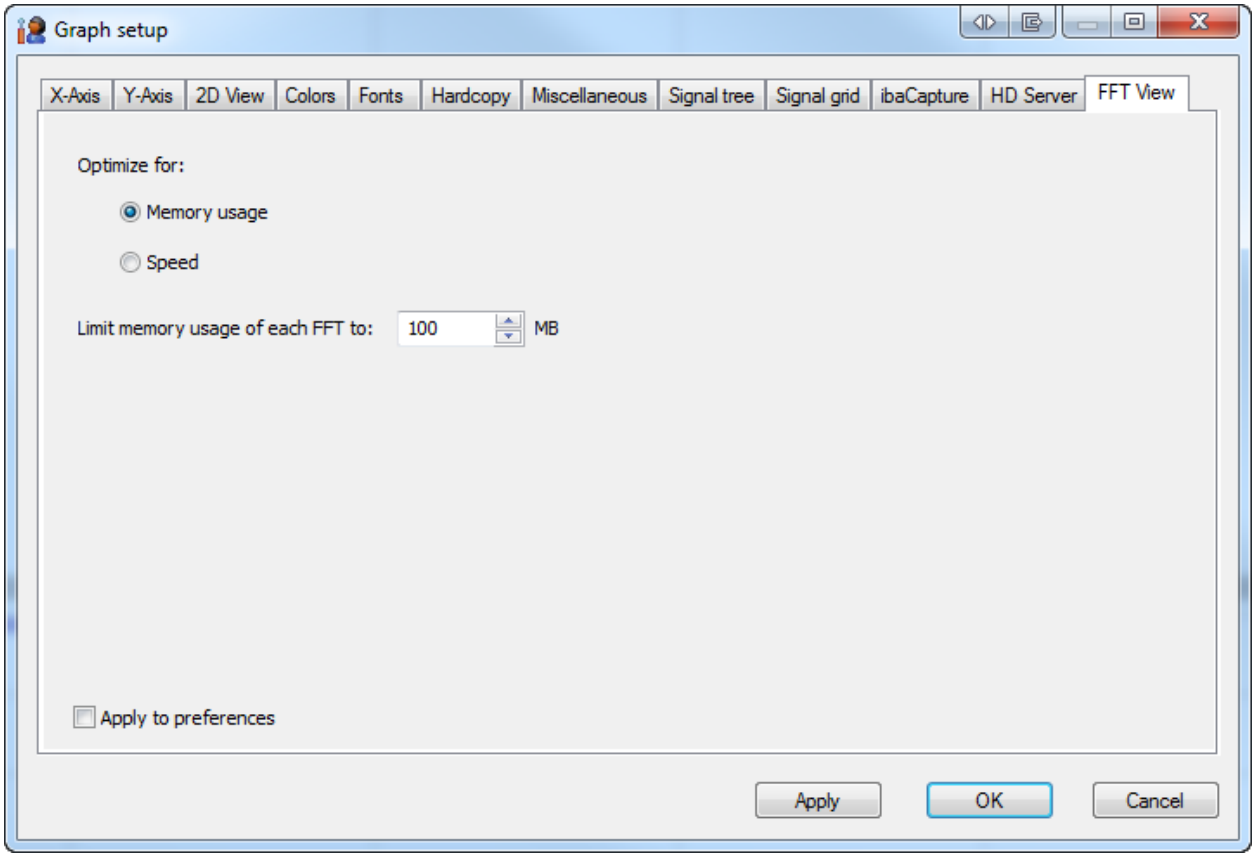

Graph setup
X-Axis
Y-Axis
2D View
Colors
Fonts
Hardcopy
Miscellaneous
Signal tree
Signal grid
ibaCapture
HD Server
FFT View
HMI
Replace paths ...
Copy video files to local disk before loading:
Path:
Maximum disk usage:
512
MB
CAM
Replace servers ...
Use alternative renderer (only applicable to ibaCapture-CAM versions prior to 3.0 , requires ibaAnalyzer restart)
Save credentials on this computer
Clear credentials
Hide video controls in CAM views, leaving more room for the video itself
Print options
Print 1 columns by 2 rows per page when printing with portrait page
Print 2 columns by 2 rows per page when printing with landscape page
Apply to preferences
Apply
OK
Cancel


Untitled - \\2012-SERVER\dat\201601\BULO_2016-01-18_12.00.00.dat - ibaAnalyzer 6.7.0 beta 10
File
Database
Historical Data
Edit
Setup
Graph Mode
File Group
View
Help
[4:1] Camera voor
[4:3] Camera tuin
Play
Pause
Stop
Zoom out
Log out
Apply time delay...
Setup overlay text
Remove overlay text
Export to video file
Export range between markers to video file
Export frame to image file
Show video controls
Signals
Search
Report info
Analysis files
Statistics
SignalName
X1
X2
Min
Max
Average
Std dev
Unit
Detectie buitendector parking
Detectie buitendector oprit
Detectie buitendector achterdeur
Detectie buitendector keuken
Detectie buitendector tuin
Signal definitions
Harmonic marker
Navigator
Overview
Markers
Ready

## 3.3 FFT view memory management

In the ibaAnalyzer setup and preference dialogs, one can specify some options for the FFT Views memory management.

- One can specify if one wishes to have ibaAnalyzer optimize for memory usage or for speed. When selecting the latter, ibaAnalyzer buffers its signals before sending them to the FFT component (requiring additional memory but being significantly faster).
- One can set a limit to the amount of memory each FFT View should use, when one tries to drop more signals on to the FFT view than this memory limit would allow, a warning is given and one has the option to cancel the drop.



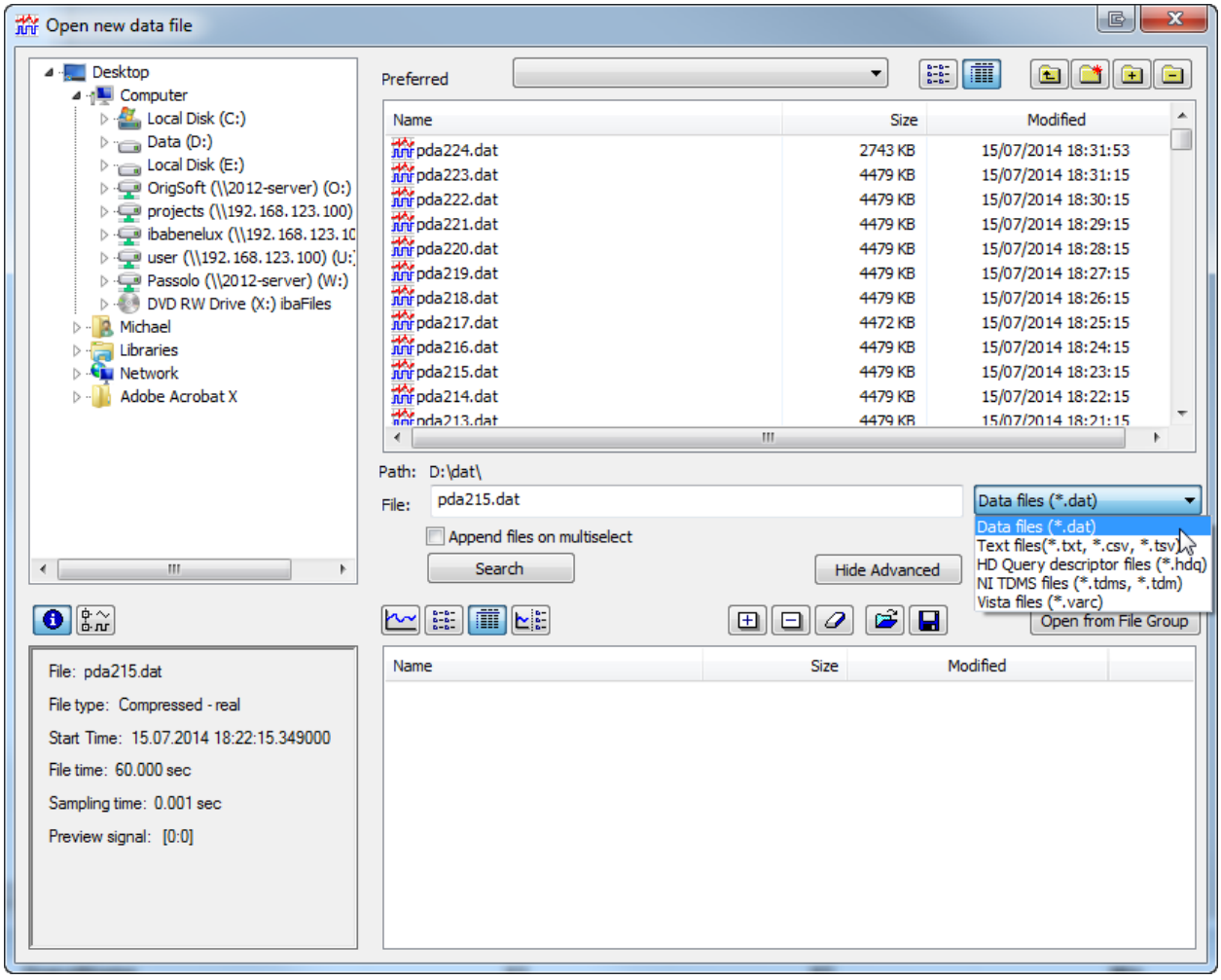
## 3.4 File type selection

In the file open dialog, it is now possible to filter on the type of input file.

Possible options are:

- Dat files, the iba proprietary .dat file format
- Text files, CSV files. Requires a license to read.
- HD Query descriptor files, files describing an HD Query.
- National Instruments TDMS files. Requires a license to read and an additional component to be installed.
- Vista Controls Vlogger files. Requires a license to read and additional components to be installed.

In previous versions of ibaAnalyzer, browsing for a file that requires a license to read caused the license to be immediately reserved. This is still the case in the current version of ibaAnalyzer, however if one does not select the file type of a license protected file format in the filter, one can avoid reserving the license.



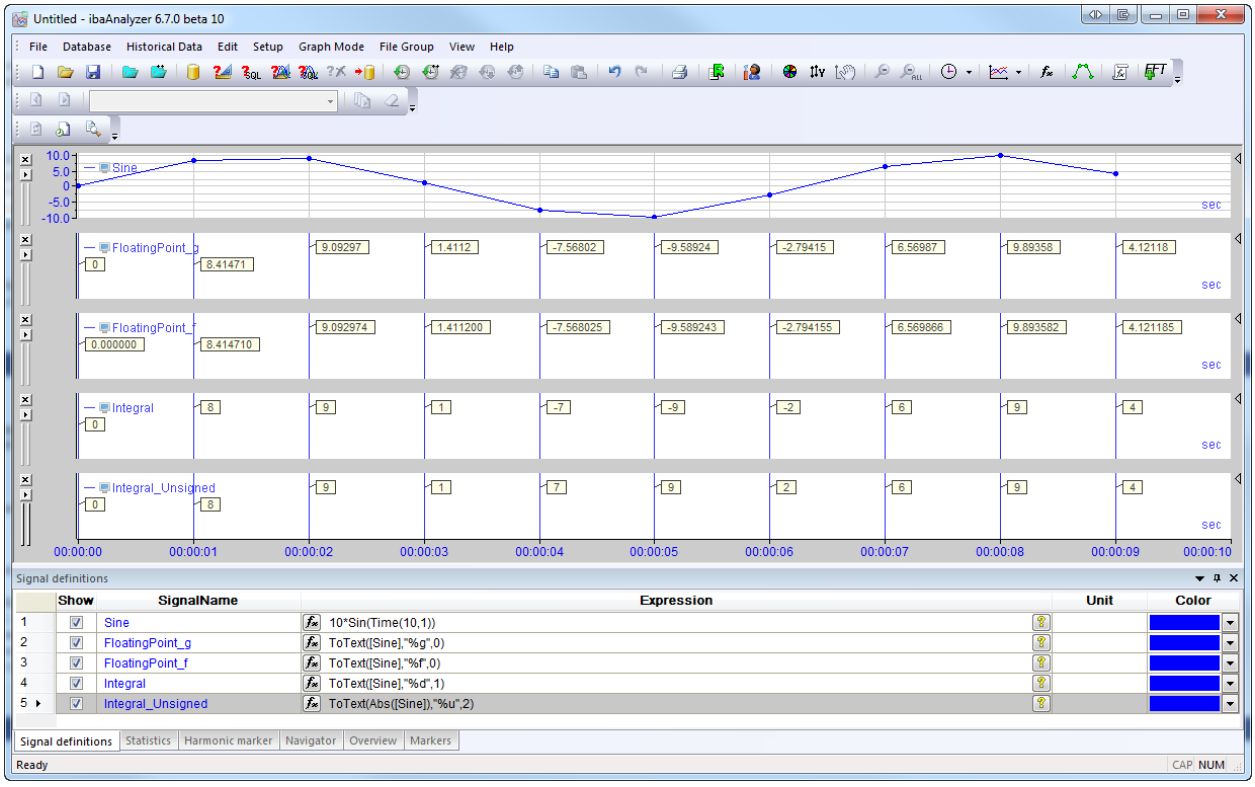
## 3.5 ToText function, additional parameter

An additional third optional parameter can be specified for the ibaAnalyzer *ToText* function which causes the data to be converted to a particular datatype before being formatted and returned as text. This allows for specifying format specifiers for integral types, previously only format specifiers for floating point data types were appropriate.

The possible values the new parameter can take are:

0. This is the default value if the parameter is omitted. It means that the data will not be converted. When used in expressions, all ibaAnalyzer data is floating point (either 32 or 64 bit depending on the source of the data or which ibaAnalyzer expressions are used) hence floating point format specifiers are appropriate.
1. The data will be converted to signed 16 bit integer.
2. The data will be converted to unsigned 16 bit integer.
3. The data will be converted to signed 32 bit integer.
4. The data will be converted to unsigned 32 bit integer.

Please refer to the documentation of the `printf` function of the C programming language to see which format specifiers are appropriate for which datatype.



# 4 Historical Data Server Queries

The functionality to query for data on an HD server has been extended so that rather than querying a fixed time range, one can query ranges that start or stop when one or more specified signal conditions are met.

In previous versions of ibaAnalyzer, the query parameters were persistent by saving a configuration file in the ibaAnalyzer application folder. Currently, multiple queries can be configured and one can still choose to store them in configuration files in the ibaAnalyzer application folder or alternatively one can choose to store the query parameters in the analysis (.pdo file).

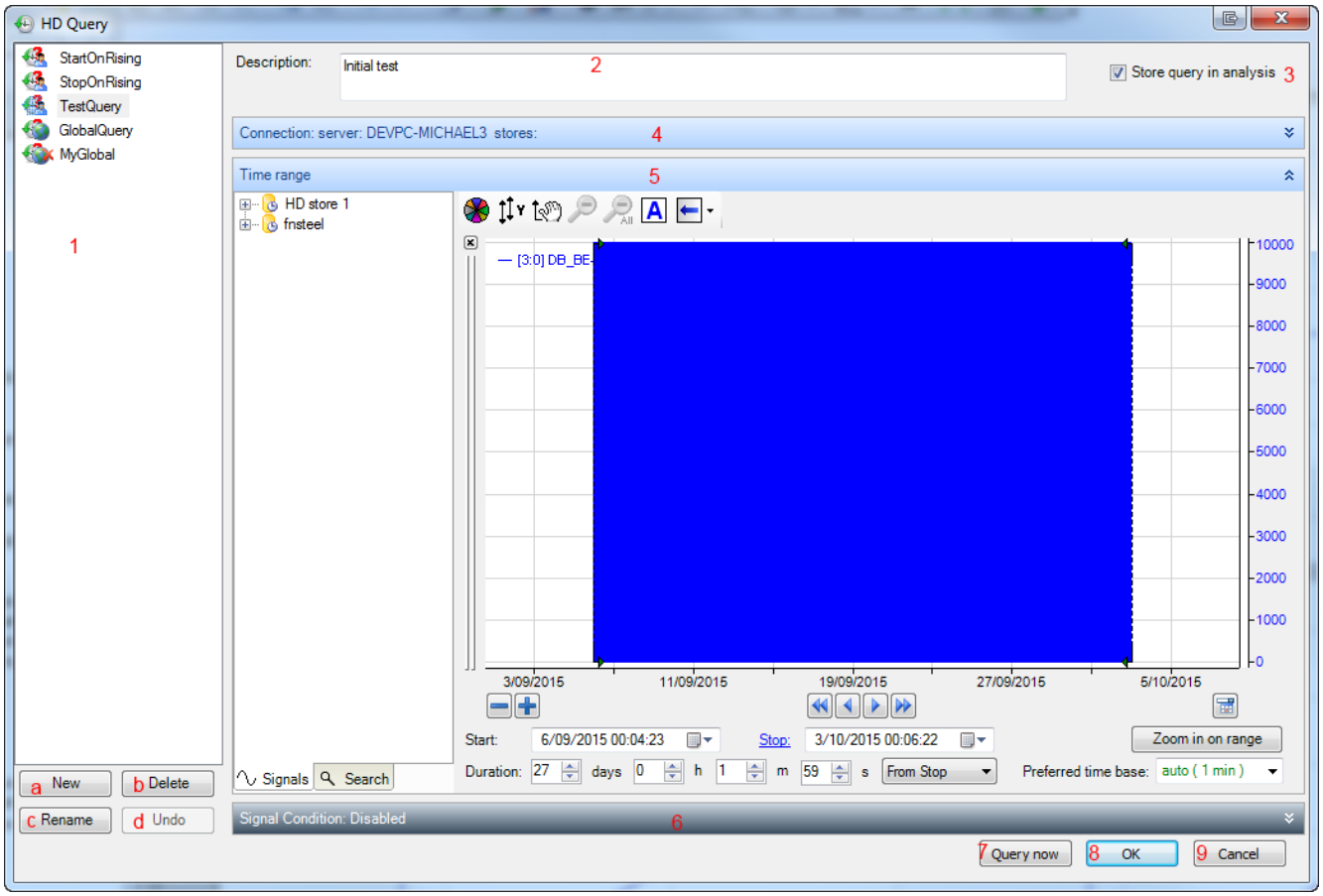
## 4.1 HD Query dialog

The HD Query dialog has been rearranged into the following main components:

1. Query selection tree. Here one can select the current query to configure or execute. Below the tree buttons are present:
   a. New: Add a new query
   b. Delete: Remove the currently selected query.
   c. Rename: Rename the currently selected query.
   d. Undo: Undo any changes in the currently selected query since it was previously saved.
2. Description text box, here one can optionally provide a description for the query.
3. Check box to indicate whether or not the query parameters should be stored in the analysis or not. If not stored in the analysis, the query is “global” which means it’s stored in the ibaAnalyzer application folder in a configuration file. The query is then available over different ibaAnalyzer sessions.
4. The “Connection” section. In this section one can specify the parameters to connect to the HD server. The section can be collapsed or reopened by clicking the title bar. When collapsed, the title bar contains a summary of the connection parameters.
5. The “Time range” section. In this section one can specify a time range over which to perform the query and also preview the data. When not enabling the signal conditions, this is the entire range of data that will be returned when the query is performed. When enabling signal conditions, this is the range where ibaAnalyzer will search for subranges satisfying the start and/or stop conditions.

   The section can be collapsed or reopened by clicking the title bar. When collapsed, the title bar depicts the selected time range (start and stop of the range are displayed).
6. The “Signal condition” section. Here one can enable and specify the signal conditions. We’ll explain this section in more detail later on. The section can be collapsed or reopened by clicking the title bar. When collapsed, the title bar will display “Enabled” if signal conditions are enabled and “Disabled” otherwise. Also the title bar has a gray background color when conditions are disabled.

7. A “Query Now” button. This saves the modified queries, leaves the dialog and performs the currently selected query.
8. An “Ok” button. This saves the modified queries and leaves the dialog but does not perform the query.
9. A “Cancel” button. This leaves the dialog, discarding any changes.

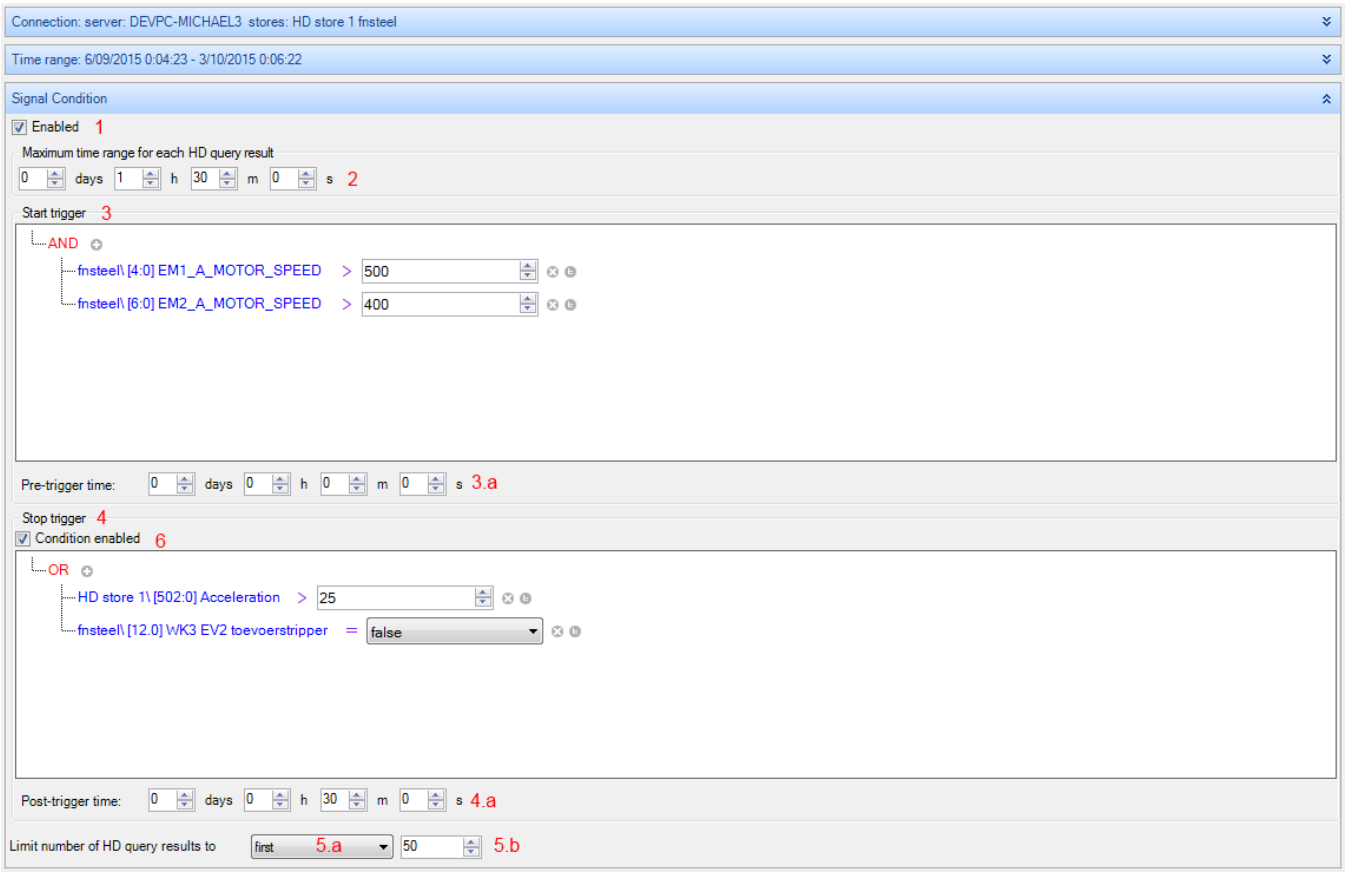


## 4.2 Signal condition query parameters

The Signal Condition section in the query dialog consists out of the following elements:

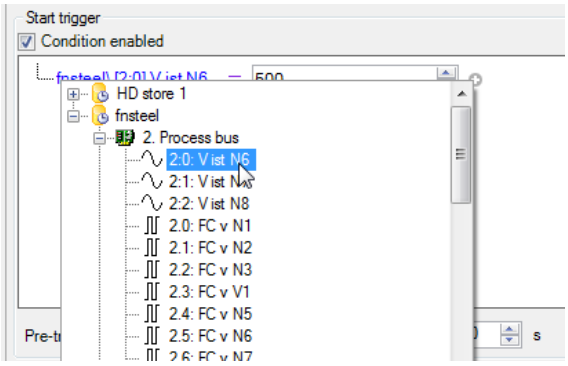
1. An “Enabled” checkbox. This checkbox needs to be checked if one wishes to perform a query based on signal conditions. If not, one single query over the entire selected range is performed (as already possible in previous versions of ibaAnalyzer)
2. A group box where one can specify the maximum time range for each HD query result. The amount of time in a returned HD query result interval will be at most this amount plus any post or pre trigger amounts.
3. A “Start trigger” groupbox. Here one can specify the signal conditions that need to be satisfied when an HD query result starts.
    a. Additionally one can optionally specify a “Pre-trigger” amount of time. Any returned HD Query result interval will start that amount of time before the actual start trigger.
4. A “Stop trigger” groupbox. Here one can specify the signal conditions that need to be satisfied when an HD query result interval ends.
    a. Additionally one can specify a “Post-trigger” amount of time. Any returned HD Query result interval will continue that amount of time after the actual stop trigger.
5. A line near the bottom says “Limit number of HD query results to” followed by:
    a. A dropdown selection box where one can select “first” or “last”.
    b. A spin control where one can set the maximum amount of query results that should be returned.

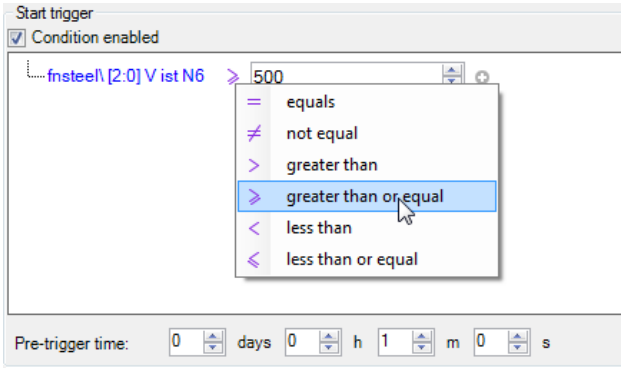
    If “first” is selected in the selection box, ibaAnalyzer will search for results chronologically from the start of the time range specified in the “Time range” section until the maximum amount of results is found or the end of the time range is reached. If “last” is selected, ibaAnalyzer will search in reverse from the end of the time range specified in the “Time range” section until the maximum amount of results is found or the start of the time range is reached.
6. When one has selected to search in chronological order (“first” selected in the selection box) there is a checkbox in the “Stop trigger” group box labeled “Condition enabled”; leaving this checkbox unchecked causes any stop triggers to be ignored, any returned HD result interval will end the amount of time specified in the maximum time group box after the start trigger. Likewise, when one has selected to search in reverse order (“last” selected in the selection box) there is a checkbox in the “Stop trigger” labeled “Condition enabled”; leaving this checkbox unchecked causes any stop triggers to be ignored, any returned HD result interval will start the amount of time specified in the maximum time group box before the stop trigger.



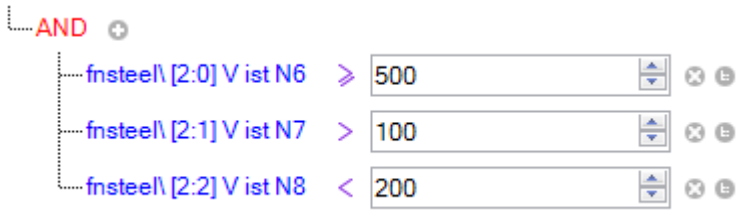

## 4.3 Configuring signal conditions

The “Start trigger” and “Start trigger” group boxes contain controls where one can specify signal conditions. You can specify a condition expression by selecting a signal from a signal tree, selecting a constant level (or TRUE and FALSE for a digital signal) and how the signal should be compared with the level.

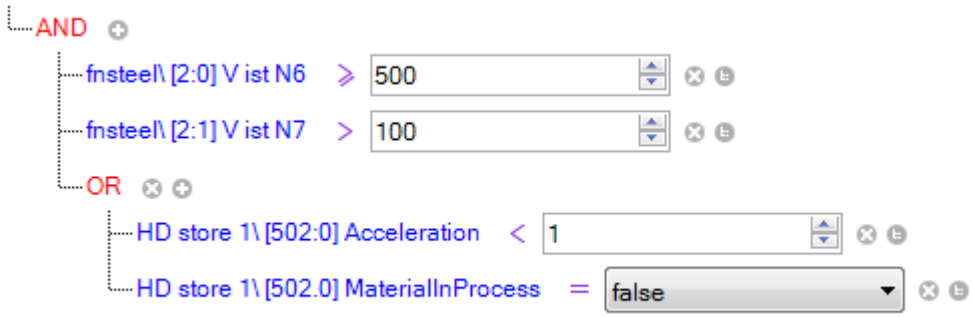



You can combine several conditions in an expression group. At the top of an expression group you can toggle if the condition expressions should be combined by logical OR (only one of the sub conditions needs to be satisfied for the entire expression group to be satisfied) or by logical AND (all sub conditions need to be satisfied for the entire expression group to be satisfied).




Expression groups can also contain conditions that are themselves expression groups. Hence entire hierarchies of expressions are possible.




The condition controls contain the following icons:

- → adds a new condition expression to an expression group or creates an expression group if the condition is not yet part of an expression group (i.e. it is the initial expression).
- → replaces the condition expression by an expression group containing the condition expression itself as first sub condition expression.
- → removes a condition expression or expression group.

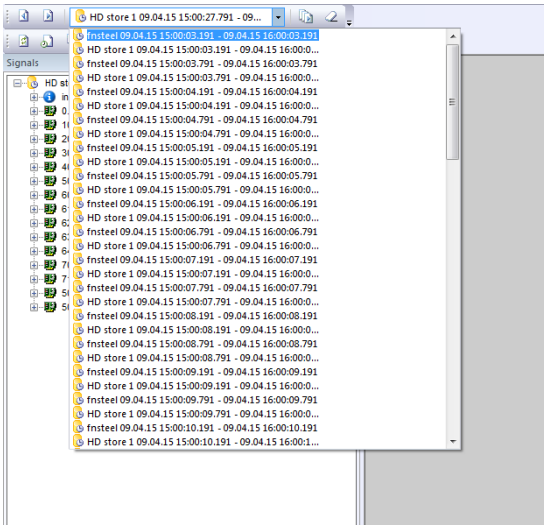
## 4.4 Executing an HD signal condition query

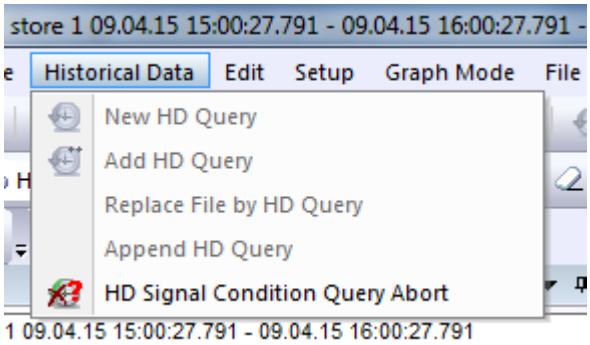
HD Signal condition queries are executed in the same way as the time range queries:

- Using the buttons in the toolbar.
- From the context menu when right clicking in the signal tree
- From the “Historical Data” submenu of the main ibaAnalyzer menu.

The main difference with time range queries is that a variable number of results can be results depending on how many start or stop triggers (when searching in reverse) are found.

All HD query results that are found are placed in the ibaAnalyzer population control (“File Group List”) while the first result that is found is loaded (i.e. the HD Query result is placed in the signal tree and any visualized signals or expressions are updated). The population control is sorted chronologically, which means that while results are found, the list is filled from top to bottom when searching chronologically and from bottom to top when searching in reverse order. The signal condition based HD query can be aborted by clicking the *Abort* button in the toolbar or selecting “HD Signal Condition Query Abort” from the “Historical Data” submenu of the main ibaAnalyzer menu.






Abort button

## 4.5 Returned results

Some notes on the results returned by an HD Signal Condition query:

- For performance, ibaAnalyzer relies on the fact that HD data is organized in hierarchical levels of coarser and finer sampled data. It evaluates the conditions first on the coarsest level and skips the parts of the time range where it concludes that the conditions cannot be satisfied. If a data block on the coarsest level is not yet completed (i.e. when the HD server is still compiling it), this can mean that more recent results can be missed. Of course, if data is missing on the finest level because of the HD server cleanup, this also means results can be missed.
- The preferred time base that can be set in the time range section only applies to time range queries and is currently ignored for HD Signal Condition queries. The results returned by an HD Signal Condition are sampled at the level that would have been selected if “auto” was chosen for the preferred time base.
- When the start or stop signal conditions are satisfied for multiple consecutive samples within the search interval (specified by the “Time range” section), only the first such sample is considered for a start or stop trigger.
- When searching chronologically, ibaAnalyzer first searches for a start trigger and then searches for a matching stop trigger in the time range specified by the “Maximum time range” in the Signal Condition section starting from the start trigger. This implies that the HD Query result can lie partially outside of the selected time range specified in the Time Range section.
- When searching in reverse order, ibaAnalyzer first searcher for the stop trigger and then searches (also in reverse) for the matching start trigger in the time range specified by the “Maximum time range” in the Signal Condition section ending with the stop trigger. This again implies that the HD Query result can lie partially outside of the selected time range specified in the Time Range section.
- When searching chronologically and no matching stop trigger is found or the stop condition is not enabled, the post-trigger interval is ignored. The resulting interval will end after the amount of time specified by the “Maximum time range” parameter past the start trigger.
- When searching in reverse order and no matching start trigger is found or the start condition is not enabled, the pre-trigger interval is ignored. The resulting interval will start before the amount of time specified by the “Maximum time range” parameter in front of the stop trigger.
- HD Signal Condition query results can partially overlap. When searching chronologically, a start trigger can lie in the interval of a previously found result; however a stop trigger must lie after the interval of any previously found results. When searching in reverse, a stop trigger can lie in in the interval of a previously found result; however a start trigger must lie in front of the interval of any previously found results.
- If in the “Connection section” multiple HD stores are selected, ibaAnalyzer will return an HD query result for each store. This implies that the number of returned queries can exceed the number of HD query results specified by the limit in the “Signal condition” section (at most the number of stores times the specified limit).

- HD Signal condition start and stop triggers are depicted in ibaAnalyzer the same way as start and stop trigger of PDA data files (green arrows).