IbaAnalyzer 5.13.0 new functionality description

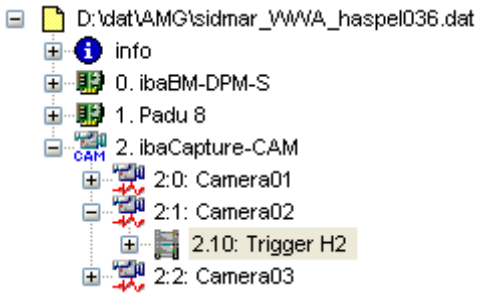
ibaCapture-CAM video playback

General

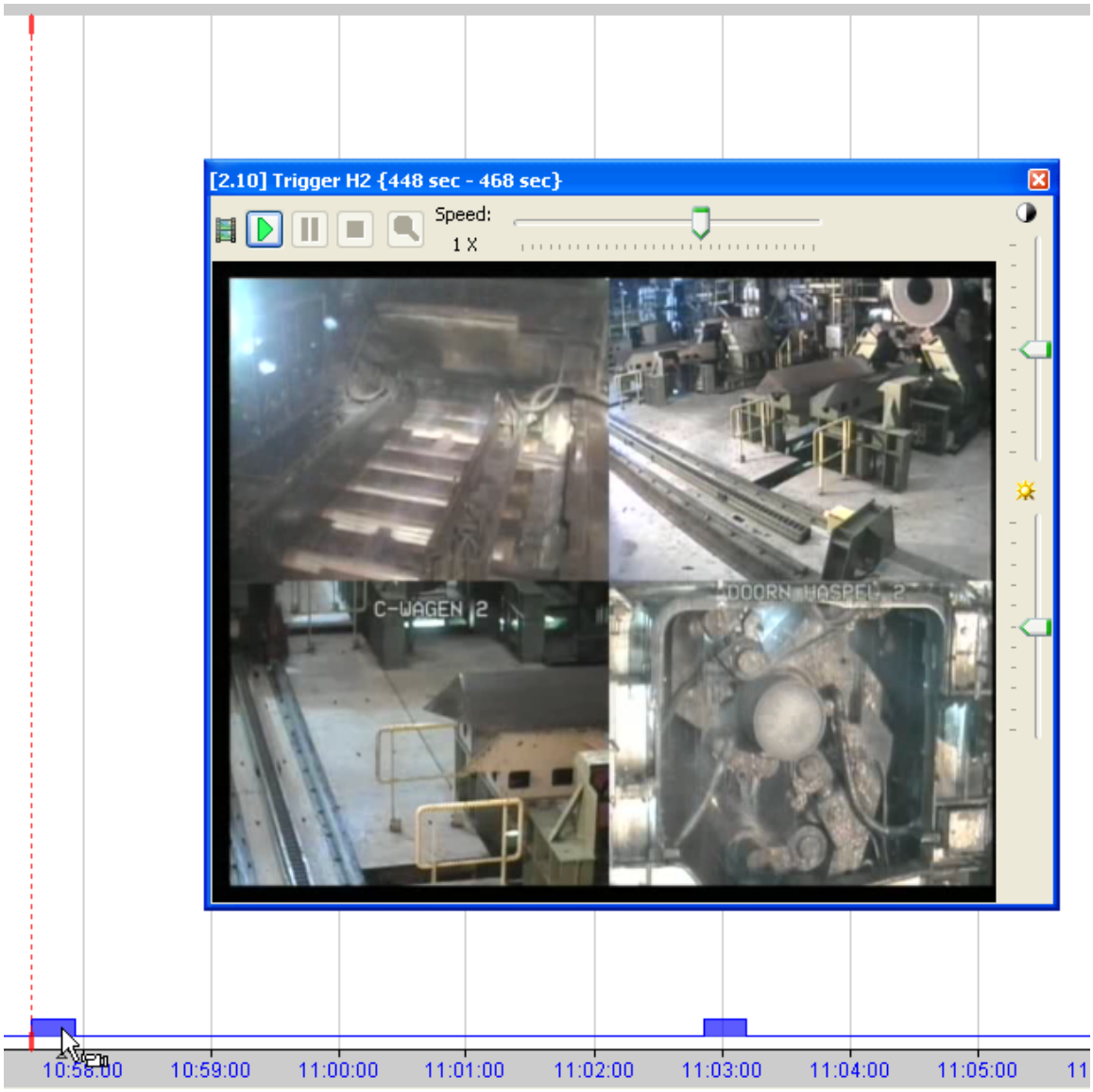
Videos recorded with the ibaCapture-CAM system can be loaded from the server and replayed in ibaAnalyzer by opening the corresponding .dat file and selecting a video synchronization signal or by selecting a trigger signal and double clicking on where the trigger signal is TRUE. While playing the video, the marker position of the X1 marker is continuously updated to correspond with the pictured video frame. Conversely the video position can be set by dragging the X1 marker.

Opening videos

A .dat file recorded with PDA can contain one or more *video-modules.* Each video module corresponds to one ibaCapture-CAM server.



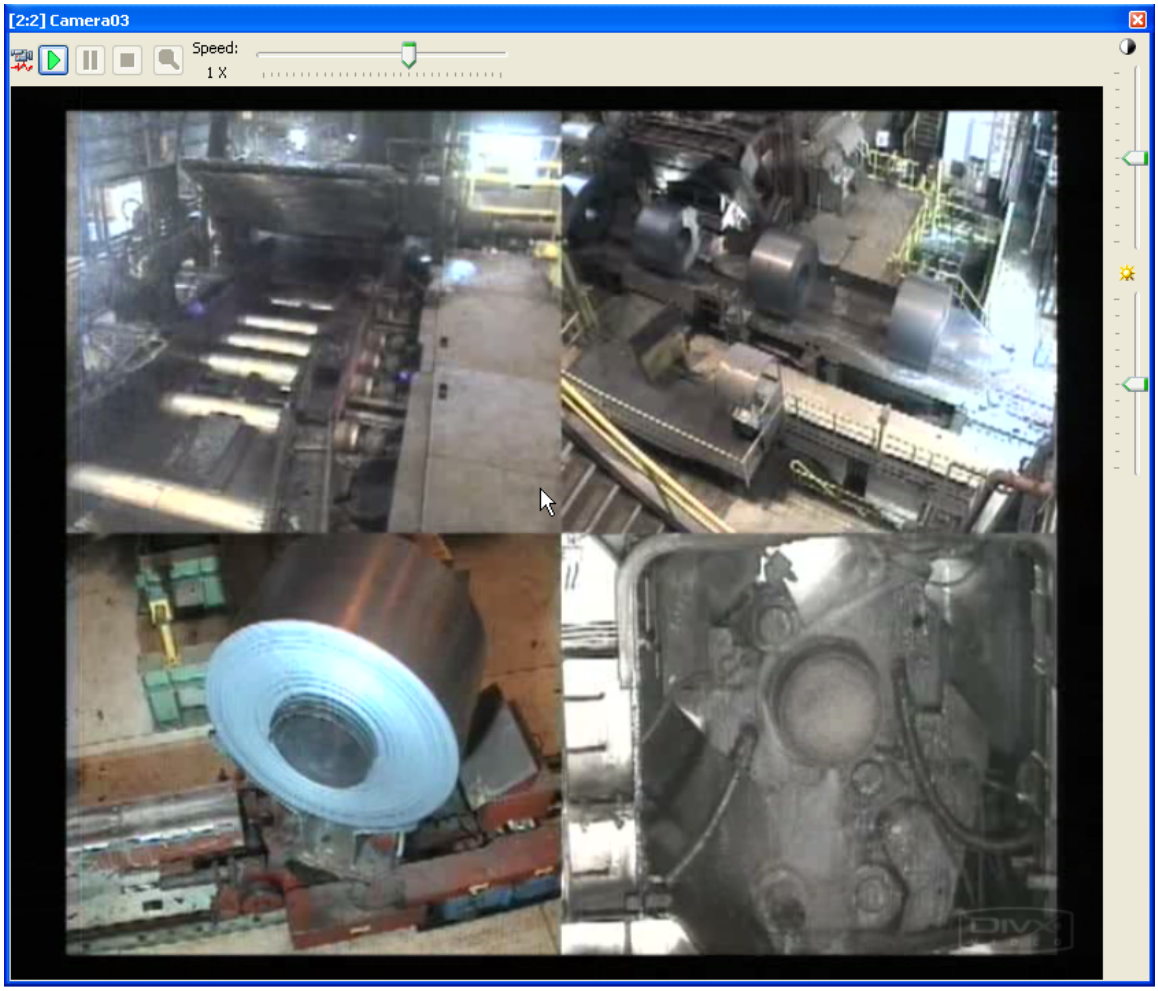
Under each video module there will be one or more *video synchronization* signals corresponding with each camera connected to the video server. These signals can be opened by double clicking them or dragging them in the recorder view to display the video. The video can then be played in a newly opened dockable window provided that the corresponding part of the unprotected buffer on the videoserver is not yet overwritten. There can be several *trigger signals* under each synchronization signal. Opening one of these signals depicts a digital signal that is TRUE for each time interval where the video trigger occurred. The corresponding video fragment was marked on the server as protected (i.e. not so fast overwritten as the unprotected buffer). Double clicking on the depicted trigger signal where it is TRUE will also open a dockable video window where you can play the protected video fragment. If you move the X1 marker out of the interval where the video trigger signal was TRUE, an appropriate error message will be displayed in the video window until you move the X1 marker again in an interval where the trigger signal is TRUE.



Playing videos

At the top left of the video window is an icon indicating whether the video is the entire video (the icon looks like a camera then) or a triggered video fragment (the icon looks like a piece of videotape).
Next to this icon is the *play* button; pressing this button causes the video to play. When the markers are visible, the X1 time marker will indicate the time position the video is on. Any other video windows will play their video simultaneously with the active video window provided the recording times overlap.
When the video is playing you can press the *pause* or the *stop* button next to the play button. The pause button causes the video to pause on the currently depicted frame.

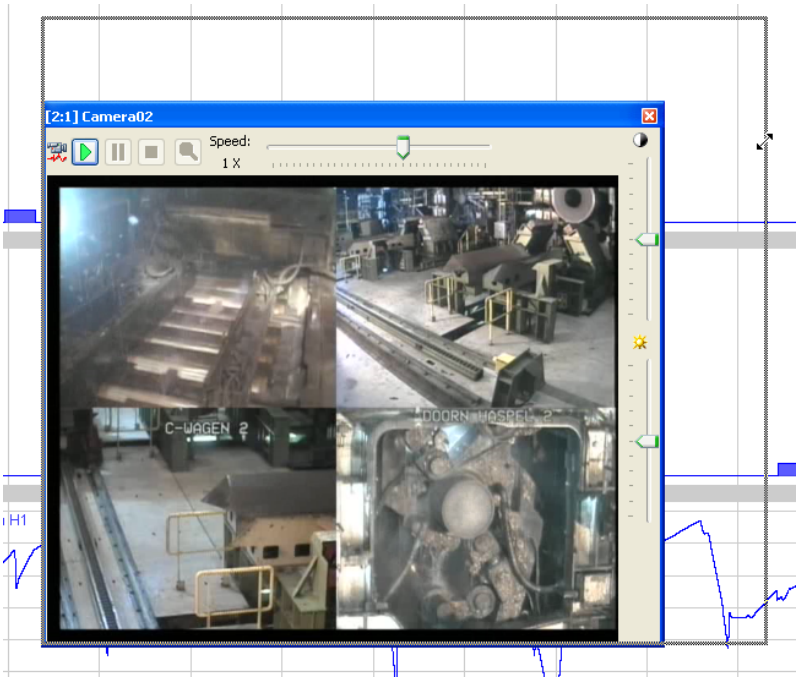
The stop button pauses the video and positions it to the first video frame, which is at the *.dat* file start time if the video is the entire video or is at the start of the trigger if the video is a triggered video fragment.



Next to the buttons is a slider bar where you can set the speed at which the video is played, you can set this to negative speeds to cause the video to play in reverse. After pressing the play button, this slider gets the focus so you can control the video speed by pressing the left or right arrow key. Pressing spacebar while this slider still has the focus causes the video to pause.
To the right of the video window are two slider bars where you can control the contrast and brightness of the video.

Resizing and docking the video window

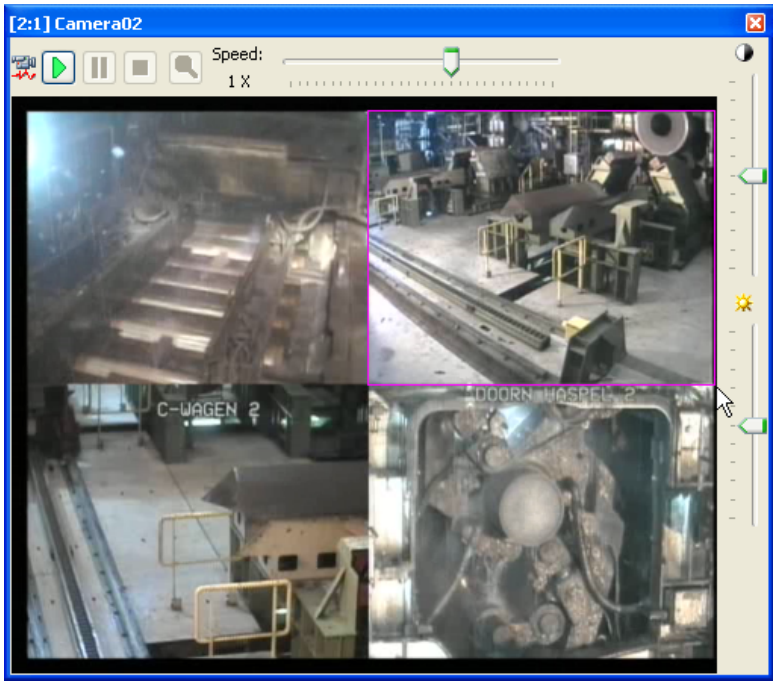
While a video window is floating you can resize the window by dragging at its edges. However, the window cannot be resized to sizes with a different length to width ratio than the original video.



The video windows can be docked to the sides of the graph window, mingling with the signal tree window or the grids window. When docked, the video window can be resized by dragging the docking window edges but the video itself will keep the length to width ratio of the original video. The video will be either centered horizontally, taking the entire available height of the docked video window or will be centered vertically, taking the entire available width of the docked video window, whichever maximizes the video area.

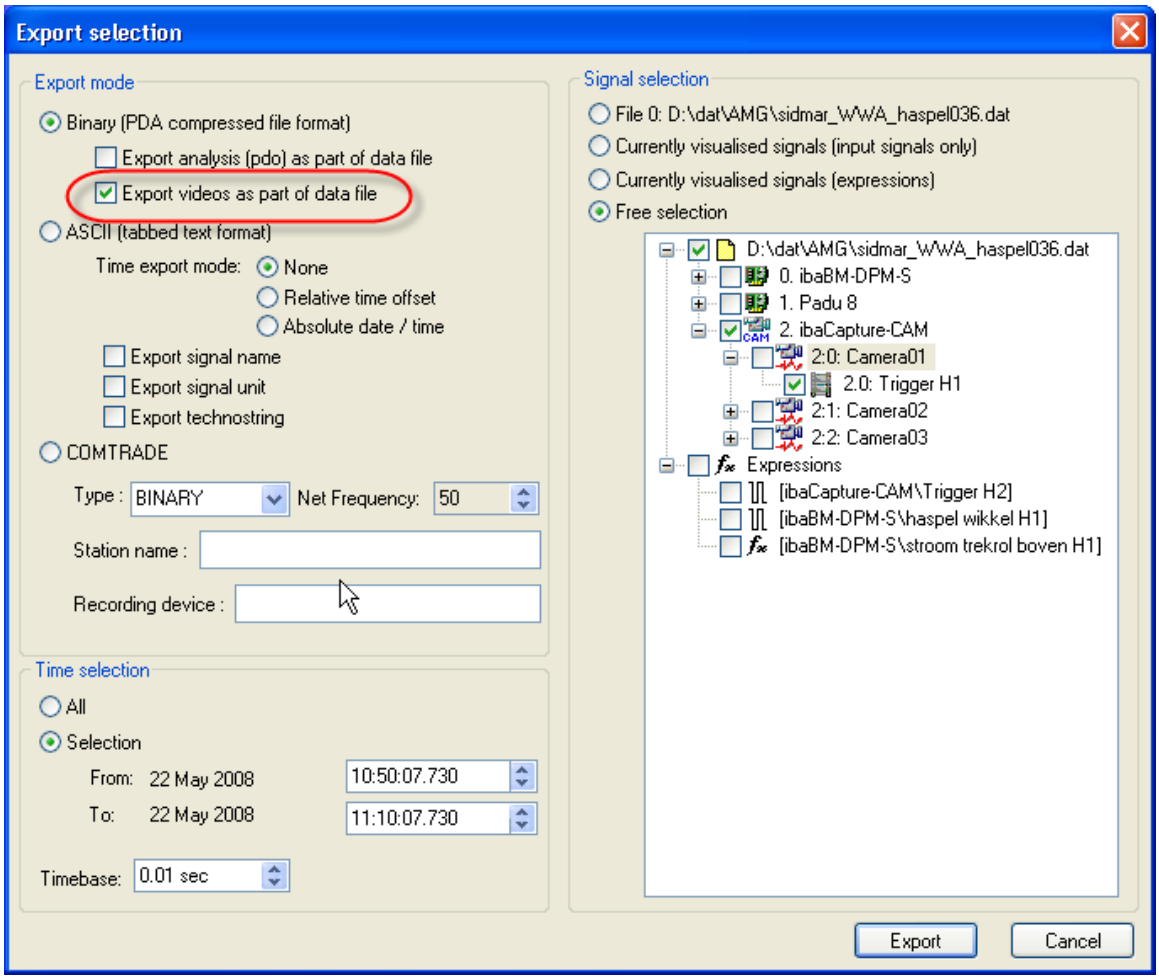
Zooming

You can zoom in by dragging a zoom rectangle over the video. You can zoom out to the previous zoom (ultimately to the entire video again) by pressing the zoom out button next to the play, pause and stop buttons.



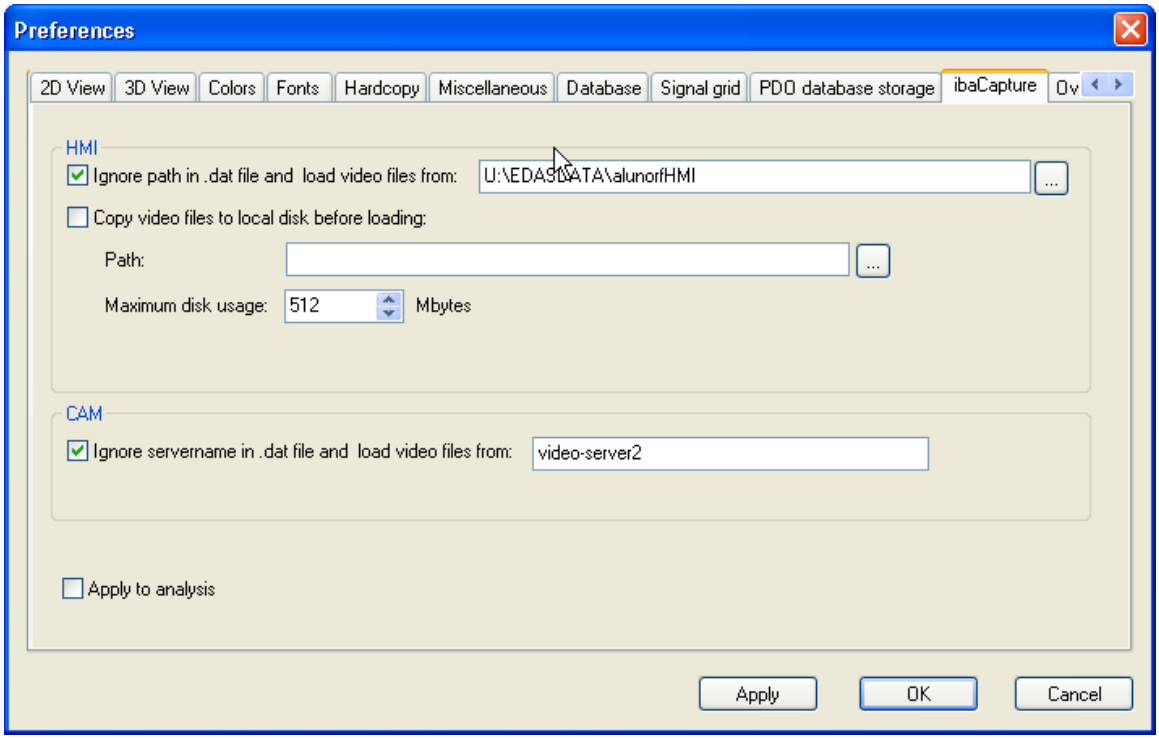
Exporting videos

You can also export video synchronization and trigger signals when exporting *.dat* files. You cannot export video trigger signals without their corresponding synchronization signals. Besides simply exporting the signals, you can also choose to embed the relevant videos in the exported *.dat* file by checking the option labeled *"Export videos as part of data file"*.



After loading the exported *.dat* file, no connection is needed to the video server when the videos are embedded.
When embedding the videos, again you cannot export a trigger signal without the corresponding synchronization signal, but you can indicate you do not want the entire video exported by unchecking the synchronization signal. You might want to uncheck the synchronization signal because the entire video will increase the size of the exported *.dat* file considerably and will take significant time to export and embed.

ibaCapture-CAM options

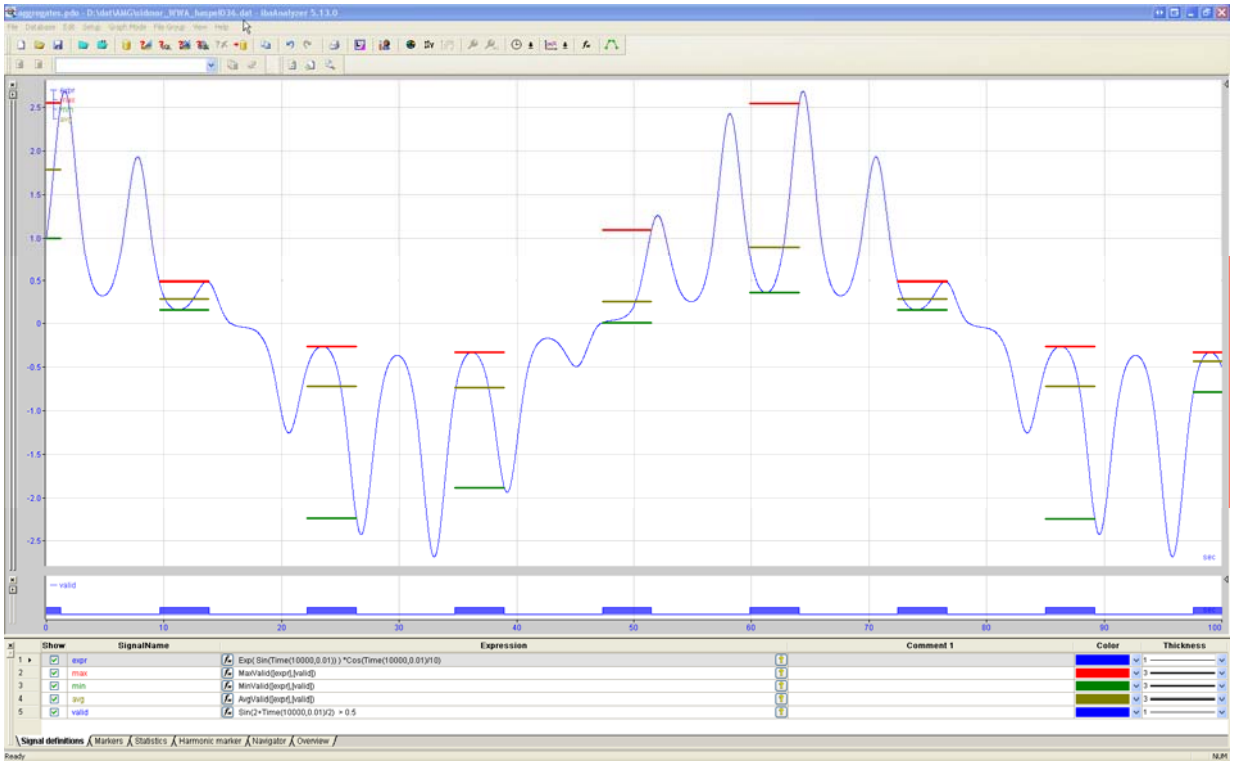


In the preferences -> ibaCapture options, you can specify an alternative video server the videos need to be loaded from than the server specified in the *.dat* file. This is ignored when opening exported *.dat* files with embedded videos in them.

New ibaAnalyzer functions

New aggregate functions

The new functions *MaxValid*, *MinValid*, *AvgValid* and *StdDevValid* calculate resp. the maximum, minimum, average and standard deviation for each interval a given control signal is TRUE. The resulting signal is invalid when the control signal is FALSE and constant for each interval where the control signal is TRUE having as value the evaluated aggregate function for that interval.



The parameters of the MaxValid, MinValid, AvgValid and StdDevValid functions are:

- Expression: the expression to be aggregated.
- Valid: The control signal to determine where the expression needs to be aggregated.

XStretchScale

Similar to the XStretch function, you can use the *XStretchScale* function to stretch or shrink a signal in x-direction, however instead of stretching or shrinking it to match a given other signal in size, you give a constant parameter to control the stretching or shrinking. The resulting size is the original size (including an optional x-offset) times the Scale factor. If the original signal had an x-offset the resulting signal has an x-offset that is the scale factor times the original offset.

The parameters of the XStretchScale function are:

- Expression: the expression to be stretched or shrinked.
- Scale: The factor Expression is stretched (Scale > 1) or shrinked (Scale < 1) with.

XAlignFft

The *XAlignFft* function takes two length based signals and calculates an offset and a scale to be applied to the second signal so that the difference between the two signals is minimized.
The function returns an array of three constant values. You can get the separate values with the *GetRows* function. The first value is the scale to be applied. The second value is the offset to be applied. The third value is a measure for the error that will be achieved between the first and second signal after scaling and applying the offset to the second signal.
The scaling needs to be applied to the second signal before the offset. You can apply the scaling with the new XStretchScale function and apply the offset with the SHR function.

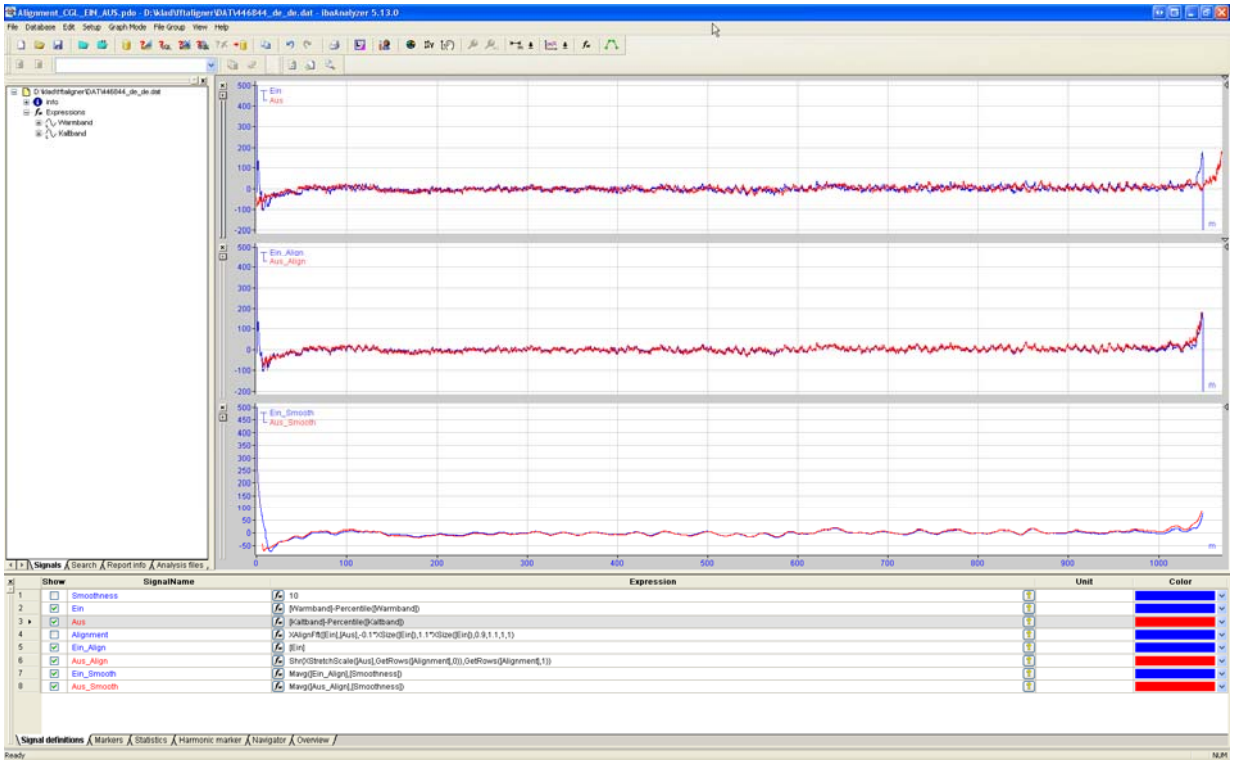
The parameters of the XAlignFft function are:

- FixedExpr: The signal to align against.
- AlignExpr: The signal to be aligned.
- Start: Minimum x-axis position where the resulting aligned signal is allowed to start (in length units).
- End: Maximum x-axis position where the resulting aligned signal is allowed to end (in length units).
- MinScale: Minimum for the scale to be considered. The resulting scale cannot be smaller than this.
- MaxScale: Maximum for the scale to be considered. The resulting scale cannot be larger than this.
- ScaleStep: Initially all scales from MinScale to MaxScale are considered with increments of this parameter divided by the length of AlignExpr. If this parameter is omitted or is smaller than the sample length of AlignExpr, the sample length of AlignExpr is taken instead. If this parameter is larger than the sample length of AlignExpr, the algorithm will try to find the best scale by recursively calling itself around the best found scale with a smaller scalestep.

- QualityParam: If this parameter is 0 or omitted, the mean squared error measure is used to estimate the difference between FixedExpr and a candidate aligned signal. If this parameter is 1 another measure is used that is more appropriate for more noisy data (e.g. a thickness profiles after galvanizing that needs to be aligned with a thickness profile before galvanizing.)

A typical use of the XAlignFft function in ibaAnalyzer is:

Signalname	Expression
Smoothness	10
Ein	[2:2]-Percentile([2:2])
Aus	[7:5]-Percentile([7:5])
Alignment	XAlignFft([Ein],[Aus],-0.1*XSize([Ein]),1.1*XSize([Ein]),0.9,1.1,1,1)
Ein Align	[Ein]
Aus Align	Shr(XStretchScale([Aus],GetRows([Alignment],0)),GetRows([Alignment],1))
Ein Smooth	Mavg([Ein Align],[Smoothness])
Aus Smooth	Mavg([Aus Align],[Smoothness])



Miscellaneous new features

Manual scale for Y-axes can now be expressions

The manual scale values for Y-axes are now expressions similar to the manual scale values for the X-axes.

Markers can be moved by the arrow keys

If the markers are visible and the graph window has the focus, you can move the markers by pressing the arrow keys.
If the Alt-key is pressed while pressing the arrow keys, the X2 marker is moved, otherwise the X1 marker.
If the Shift-key is pressed while pressing the arrow keys, both the X1 and X2 marker are moved simultaneously, keeping the distance between them constant (similar as pressing the Shift-key while moving the markers with the mouse).
If the Ctrl-key is pressed (possibly in combination with the Shift- or Alt-key) while pressing the arrow keys, the markers will only move to sample points of the currently selected signal in the grids (similar as pressing the Ctrl-key while moving the markers with the mouse).
When not pressing the Ctrl-key, pressing an arrow key will initially move the marker one pixel at a time, keeping the arrow key pressed increases the speed at which the marker moves.