New Features in ibaPDA v6.31.0

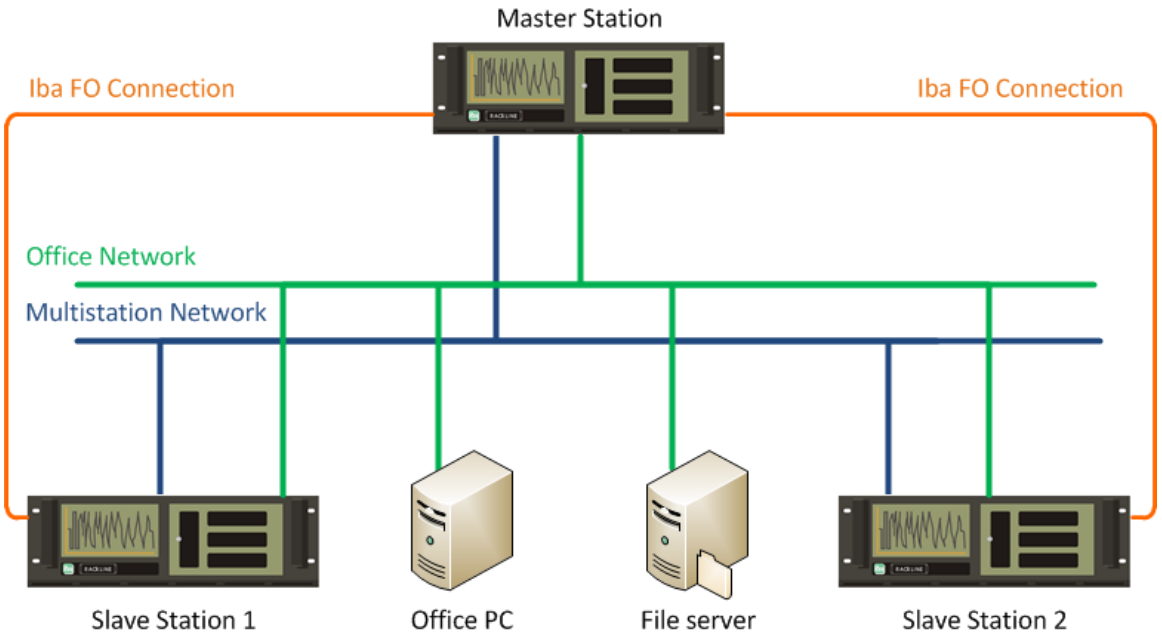
Author: iba Gent

Date: 02/01/2014

Table of contents

1 Multistation

The multistation mode of ibaPDA means that multiple PCs running ibaPDA can synchronously acquire data and trigger data stores on each other. A multistation system consists of 2 to 32 ibaPDA systems. There is 1 master system and the other systems are slaves.

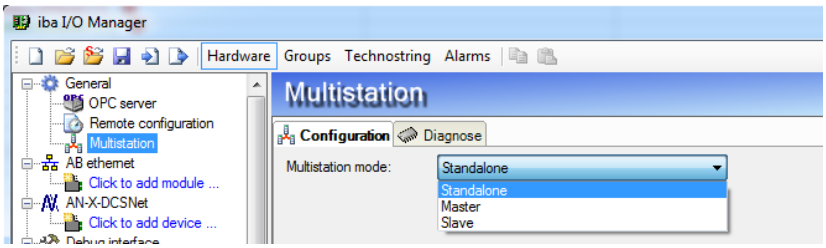


There are 2 connections between the master and each slave:

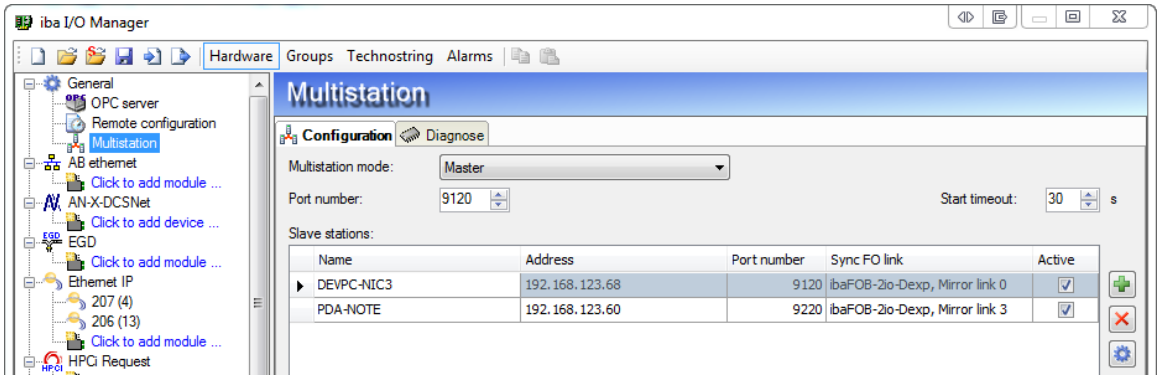
- A network connection that is used to send commands between master and slave (e.g. start, stop, sync data messages …).
- A fiber optic connection between ibaFOB-D boards that is used to transmit the sampling clock.

It is recommended that all systems have 2 networks:

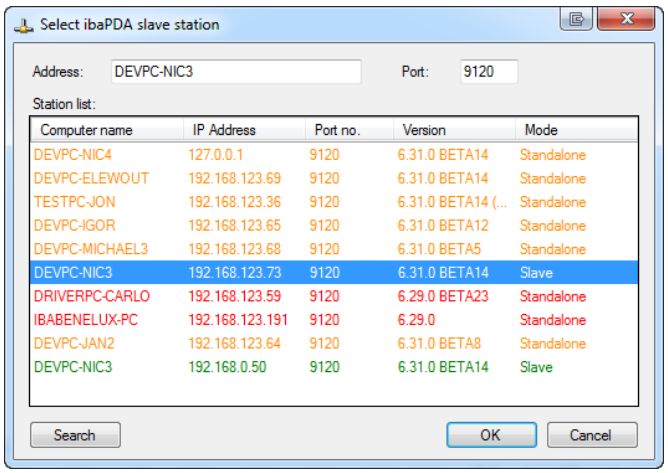
- A network which is only connected to the other ibaPDA systems and is only used for the multistation communication (Multistation Network in figure).
- A network which can be used to access an ibaPDA system from the outside, e.g. to access dat files or to write dat files on a file server (Office Network in figure).



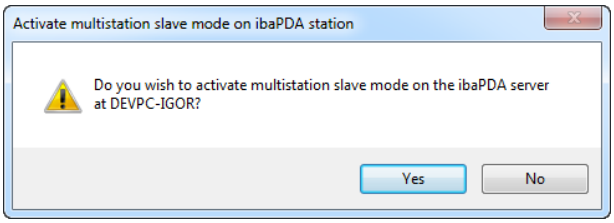
The multistation configuration is done in the I/O manager. An ibaPDA system can be in standalone, master or slave mode. In standalone mode (=default mode) the ibaPDA system is working on its own.



In master mode you have to enter all participating slave stations in a table. You can use the plus button to add a station. This will open the station browser.



When you select a slave that is in standalone mode then the following message box will appear.

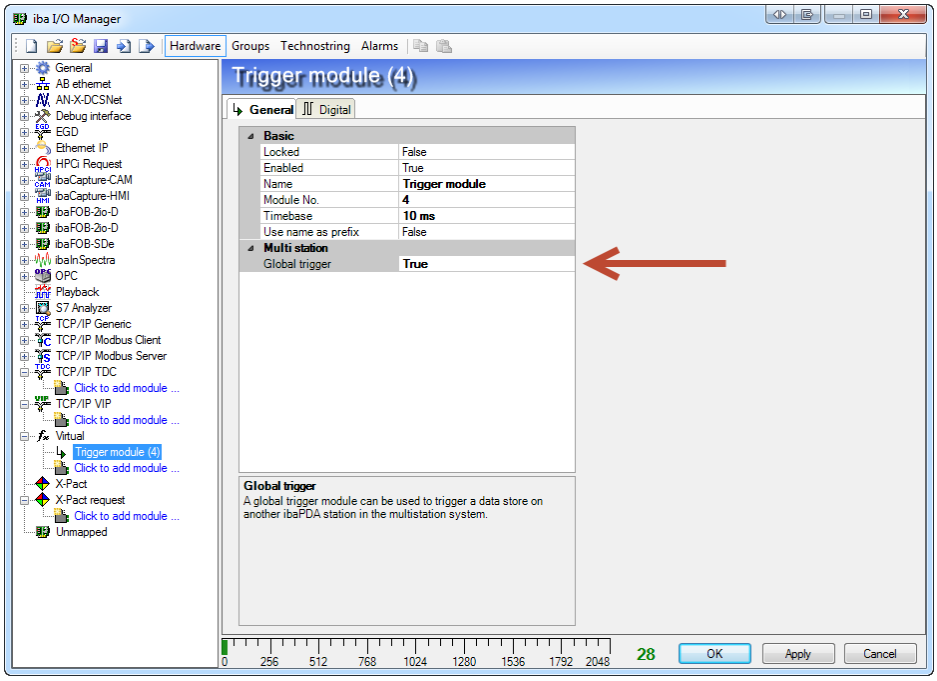


If you select yes then the acquisition will be stopped on the slave ibaPDA server and it will be configured in multistation slave mode so that it is ready to be started together with the master ibaPDA station.

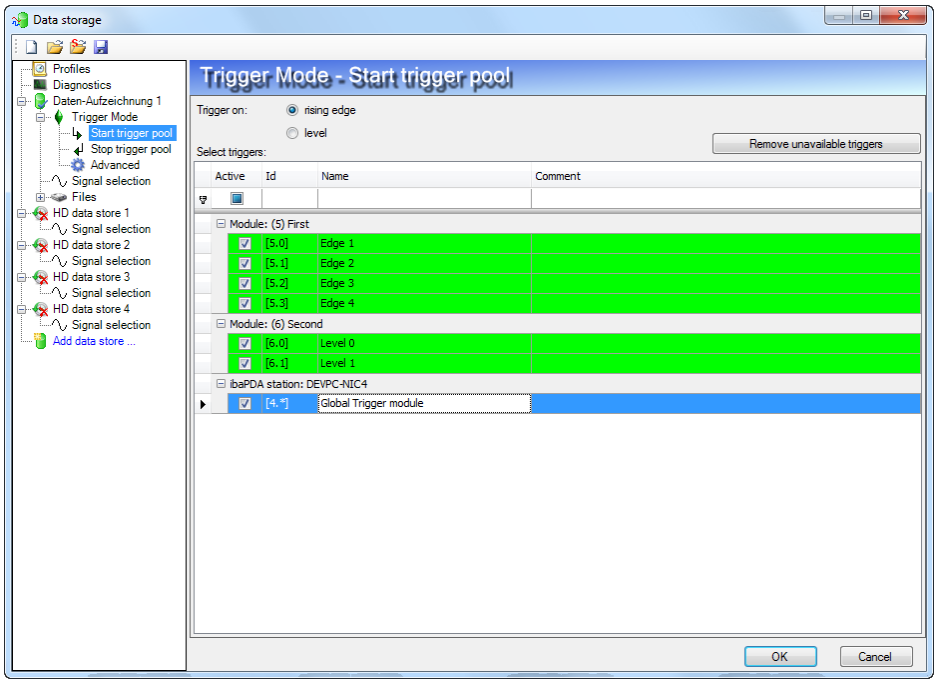
For each slave station you have to define its name (=by default equal to PC name), its address (IP address or host name), its port number and which fiber optic link it is connected to. The fiber optic link has to be connected from a mirror link on an ibaFOB-D board in the master system to the **first** fiber optic input of the **master** ibaFOB-D board in the slave system.

The start timeout parameter determines how long a station will wait until it has a connection with the other stations when starting the acquisition.

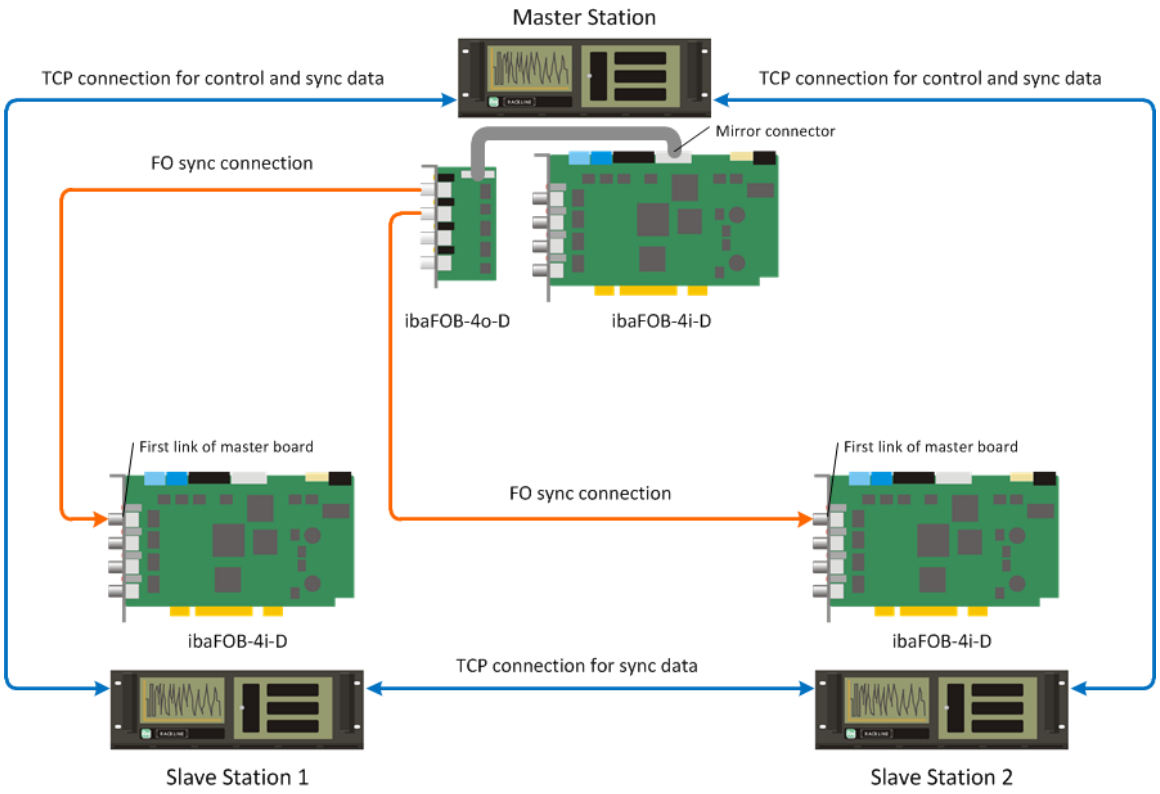
In slave mode you only have to configure your port number and the start timeout. It is always the master that connects to the slaves so a slave doesn't need to know the master's address.



In a multistation system a trigger on one station can start or stop a datastore on another station. This is accomplished via so-called *global triggers*. The trigger module has a new property called *Global trigger*. If you set this to true then all the signals in this trigger module can be used on other stations to start or stop a datastore.



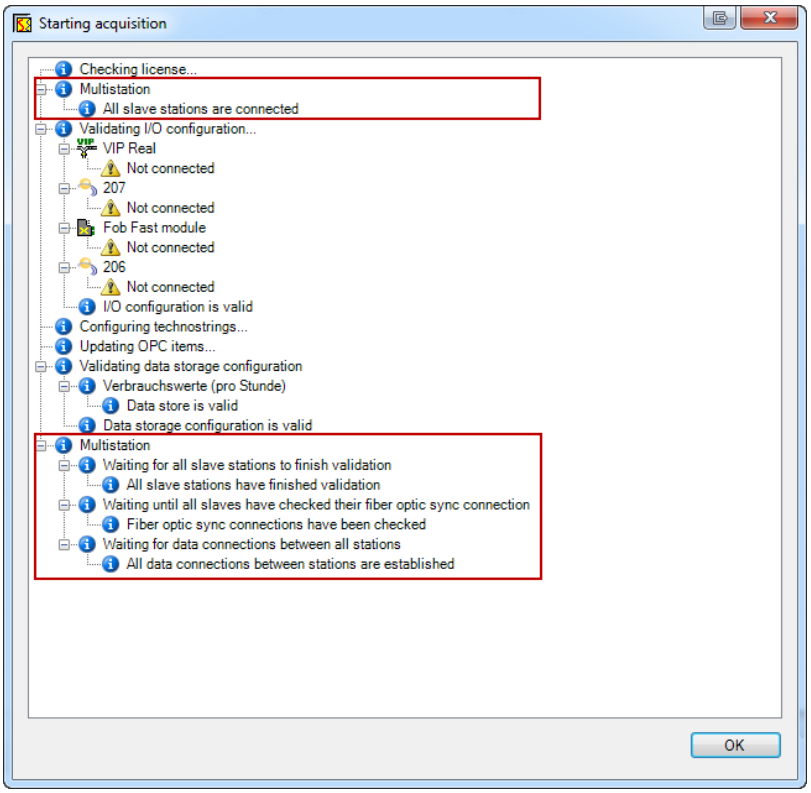
In the data store configuration the external triggers appear in the trigger pool. In the figure above you see a global trigger module called *Global Trigger module* on station devpc-nic4. There is just a single row per external global trigger module. You can't select the single trigger signals of an external global trigger module. The row corresponds with an OR operation of all the trigger signals of the external global trigger module.



The picture above gives a more detailed overview of the connections between the master station and slave stations. The master is connected to each slave with an FO cable going from an ibaFOB-4o-D connected to the mirror connector of a FOB-D board in the master station to the first link of the master FOB-D board in the slave station. The master sends a clock signal over this FO cable that allows the different stations to sample the data at the exact same time. This clock signal is also forwarded to the attached Padu-M and flex devices like the ibaPADU-S-CM, ibaPADU-S-IT, ibaLink-VME, …. The attached devices use the clock signal to all measure at the exact same time. Pda knows the transport delay corresponding with the different FO protocols and it can compensate these delays so that signals coming from different devices using different FO protocols can be perfectly synchronized.

There is also a network connection required between the different stations. The master station connects to the different slave stations. This network connection is used to send start and stop commands. It is called the control connection. It is also used during the start to negotiate timing between the stations, to check for validation errors and to start the measurement perfectly synchronized. During the start procedure there is a second TCP connection established between the master and the slaves. This second connection is called the sync data connection. If there is more than 1 slave then slave stations also create sync data connections to each other. Each station in the multistation system will have a sync data connection to all the other stations. The sync data connection is used to let all the stations process the data synchronously. The master station determines how much data coming from the driver will be processed. Each station reads this amount of data and calculates its own locally defined global triggers. The trigger events are gathered in a sync data message and this is sent to all other stations via the sync data connections. On the master station the actual system time corresponding to the first sample received from the driver is determined and this is also put in the sync data message. All stations wait until they have received a sync data message from each station and then they start processing the data. The system time of all the stations doesn't

need to be very accurately synchronized to each other because all stations use the system time of the master station for the dat files.

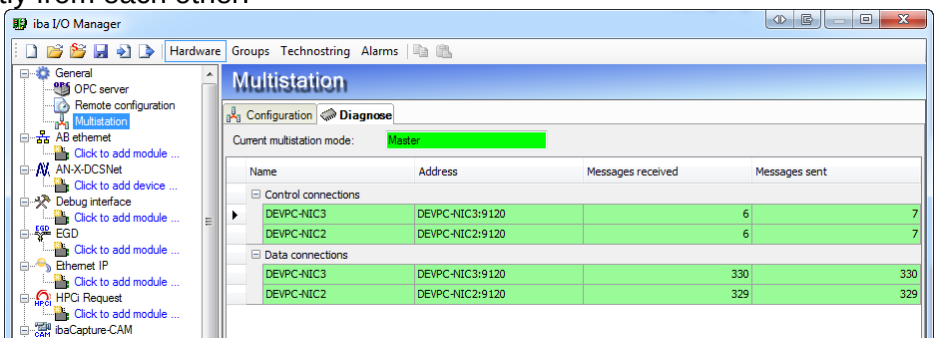


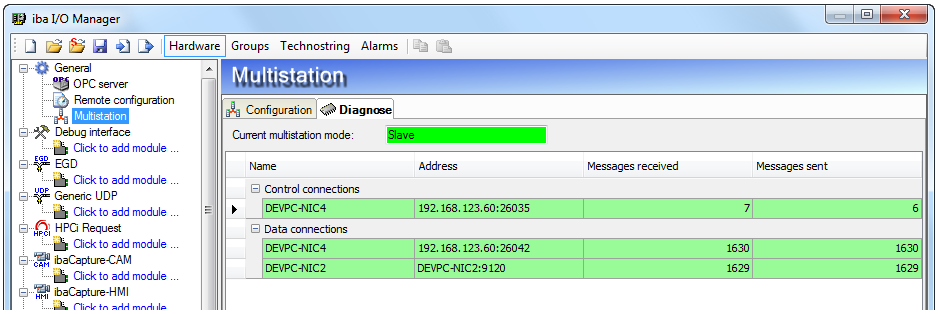
In a multistation system all stations are started and stopped together. If you start the acquisition on one station (no matter if it is a master or a slave) then it will start the acquisition on all the other stations. The same applies to stopping the acquisition. When a master starts the acquisition and he can't connect to a slave within the start timeout then this slave station will be disabled. When no slave is connected then the master will start in standalone mode. When a slave starts the acquisition then he waits until the master connects. When this doesn't happen within the start timeout then he starts the acquisition in standalone mode. When the acquisition is started and the configuration in one of the stations is not valid then this station will be excluded from the multistation system. If this is the master station then all slaves will start in standalone mode. If it is a slave station then this station will not be part of the multistation system until the next start.

When the master station has started with 1 or more slave stations missing then he will automatically restart the acquisition when one of these slave stations reconnects. If the network connection between any 2 stations is disconnected during the acquisition then the acquisition will be restarted after a timeout of 5 seconds. The same applies to the FO sync connection between master and slave. If a slave couldn't start in multistation mode because the FO sync connection wasn't ok during the start of the acquisition then it will monitor the FO sync connection. If the FO sync connection is restored and stays ok for 2 seconds then the slave station will automatically restart the acquisition and become again part of the multistation system.

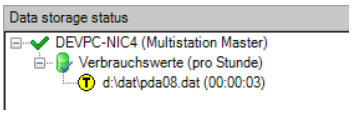
When the acquisition is running and the ibaPDA service on a station is stopped (because e.g. the PC is shutting down) then the acquisition will be stopped on all stations and it will automatically be restarted without this station. If the station was a slave then the master and the

other remaining slaves will start together. If the station was the master then all slaves will start independently from each other.





The two screenshots show the diagnose information for multistation. It shows the mode the acquisition is currently in. This could be master, slave or standalone. It shows the different control and data connections between the stations. The master station has control connections and data connections to all slave stations. The slave station has a control connection to the master and data connections to the master and to all other slave stations.

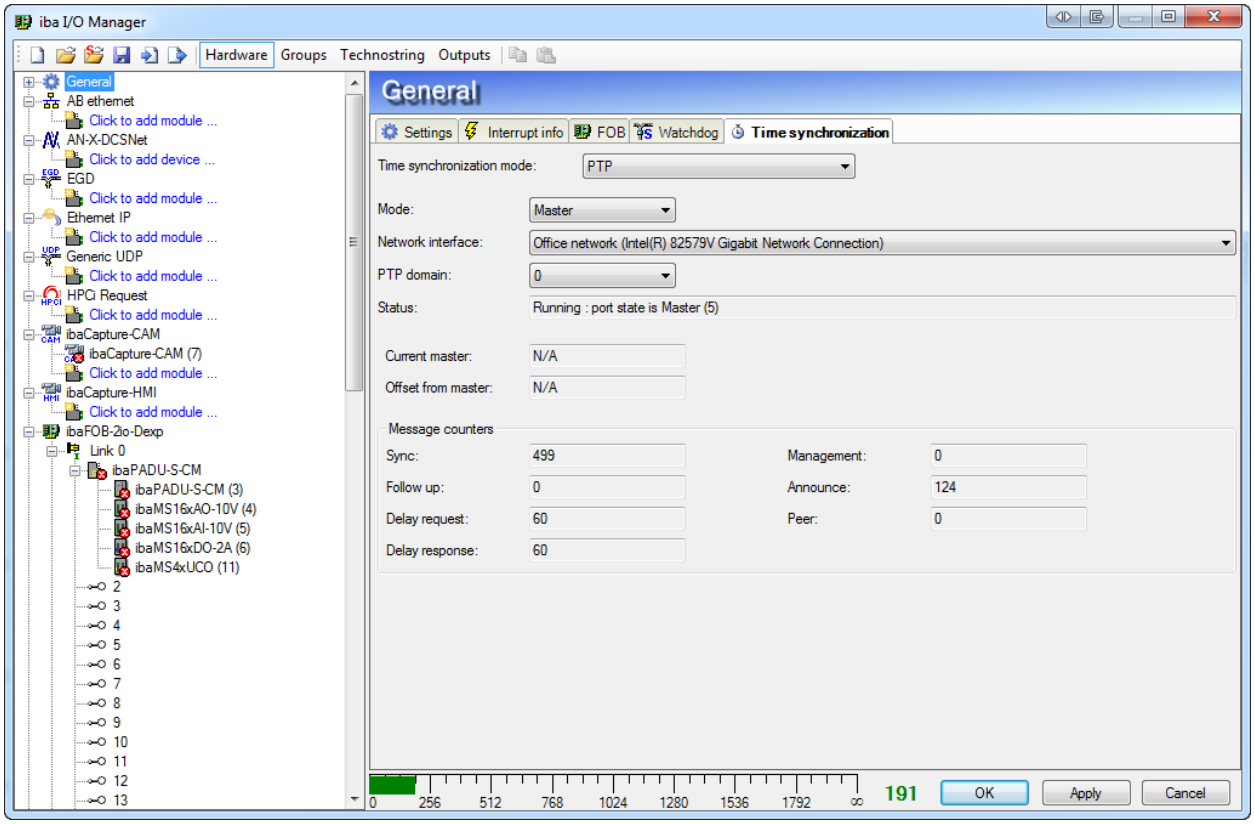


The current multistation mode is also visible in the data storage status window.

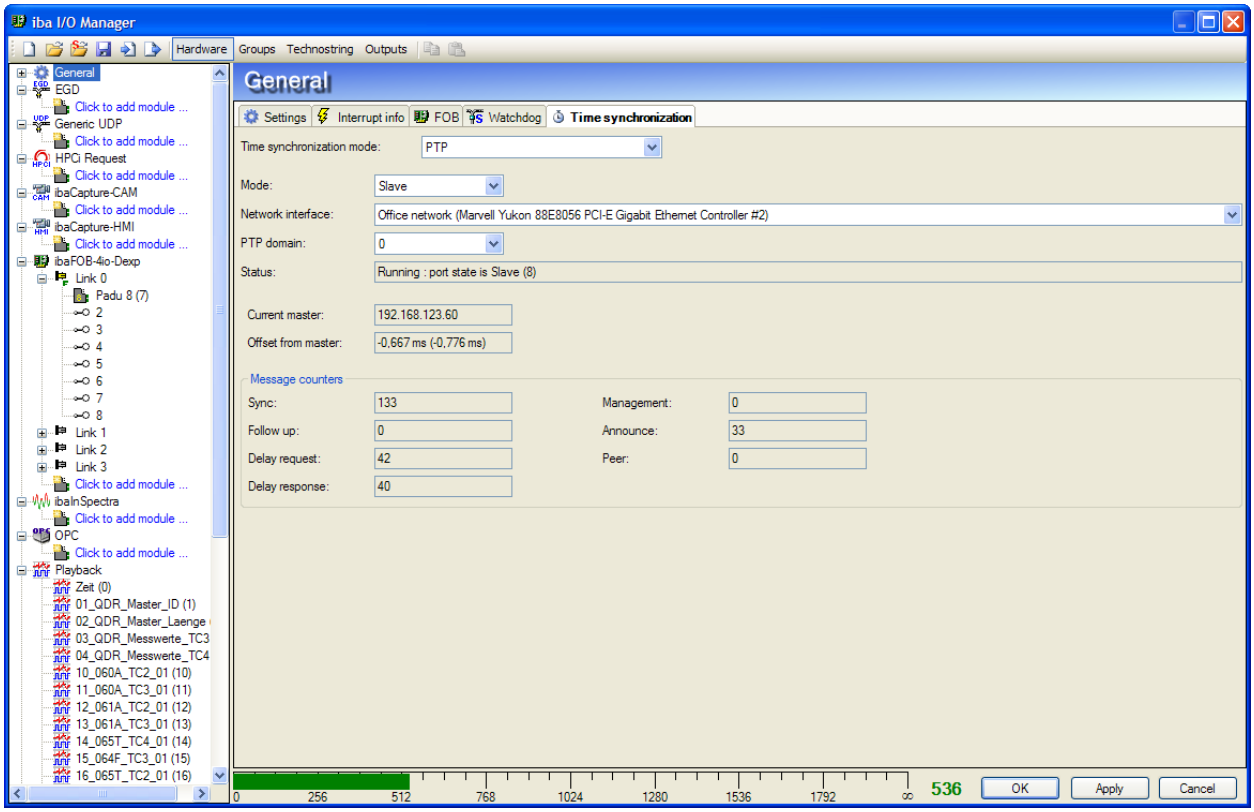
There is a new function called MultiStationStatus that you can use in expressions. It returns the current multistation mode: 0=standalone, 1=slave and 2=master.

2 PTP time synchronization

The precision time protocol (PTP) is a protocol to synchronize the system time of different systems (PC, PLCs, other devices) via the network. It is standardized as IEEE 1588. In a typical PTP system there is one master clock and the others synchronize themselves to it. Software based PTP like implemented in ibaPDA can be used to synchronize no better than 5ms. ibaPDA can be configured as a master clock or as a slave clock. When ibaPDA is configured as master then it distributes its clock and other (ibaPDA) systems can synchronize themselves to it. When ibaPDA is configured as slave then it will synchronize to a master clock if it is available.



In ibaPDA you can enable PTP time synchronization on the „Time synchronization“ tab of the general node in the I/O manager. You first have to select if this ibaPDA is going to be a PTP master or PTP slave. PTP uses UDP multicast to send messages between master and slaves. So you have to select which network interface needs to be used to send the multicast messages. The multicast address is determined by the PTP domain. There are 4 standardized domains each one corresponding to a specific multicast address.



In slave mode the current connected master is shown and the offset between the master‘s system time and the slave’s system time. The offset between brackets is the filtered value and the other one is the last measured value. In master mode these 2 fields show N/A which means not available.

Finally there is a list of message counters. There is one message counter per message type. Normally the sync, announce, delay request and delay response messages should be counting when there is a master-slave connection.

3 Bachmann M1 request

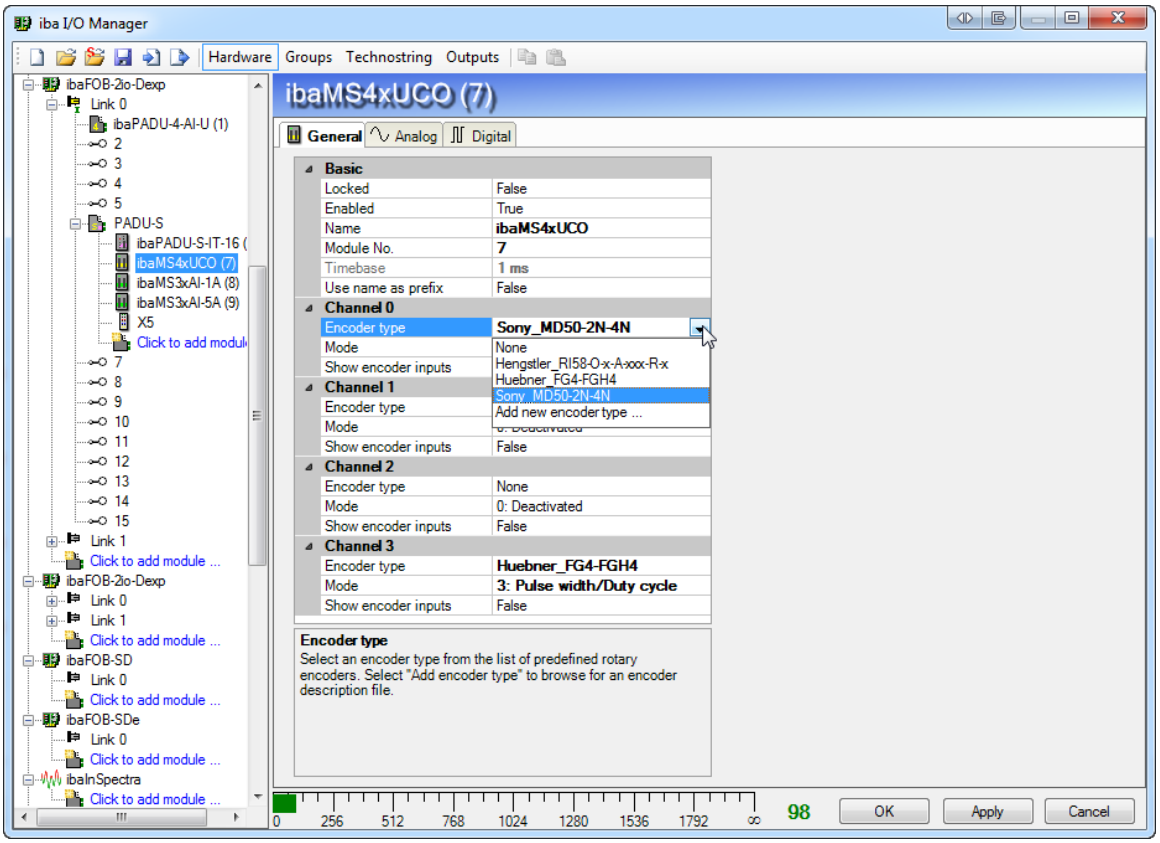
See separate document sw_COMM_ibaPDA-BachmannRequest_v1.0_en_A4.doc

4 Support for new devices

4.1 ibaMS4xUCO

The ibaMS4xUCO counter module has several inputs and outputs:

- Connector X28: 4 counter channels. Per channel there are 4 digital inputs: A, B, N and MF. Depending on the connected encoder and the selected mode you have to connect 1 or more digital inputs.
- Connector X5: 8 digital inputs
- Connector X27:4 digital outputs



On the general tab you have to configure the counter channels. First you have to select the connected encoder. IbaPDA has 3 preconfigured encoder types: Hengstler RI58 series, Heubner FG4 and FGH4 series and the Sony MD50-2N-4N series. If you have a different encoder type then you should provide the technical documentation of the encoder to your local iba support and they will provide you with an encoder descriptor XML file. Once you have such a file you can select “Add new encoder type…” and browse to the XML file. This encoder descriptor XML file contains the necessary settings for the digital inputs that belong to the counter channel.

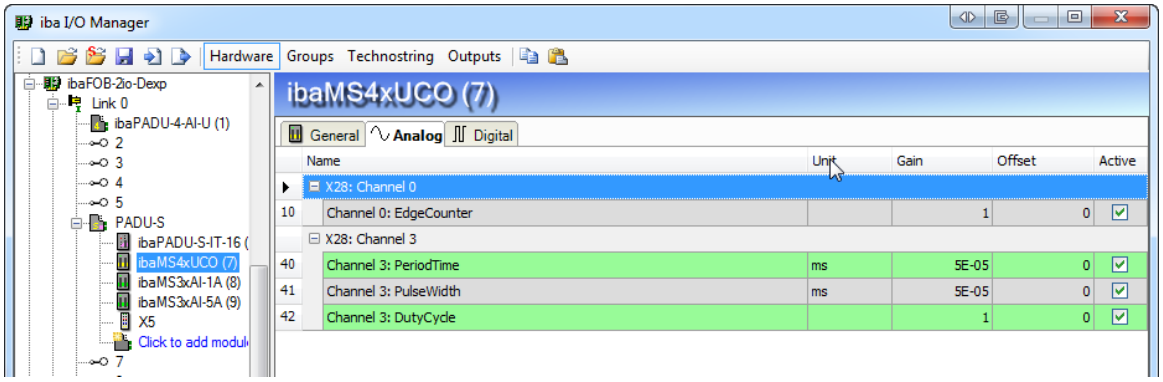
Once you have selected an encoder type you can select the mode you want it to work in. There are 6 modes supported:

- Pulse counter
- Period/Frequency
- Pulse width/Duty cycle
- Up/down counter
- SSI slave

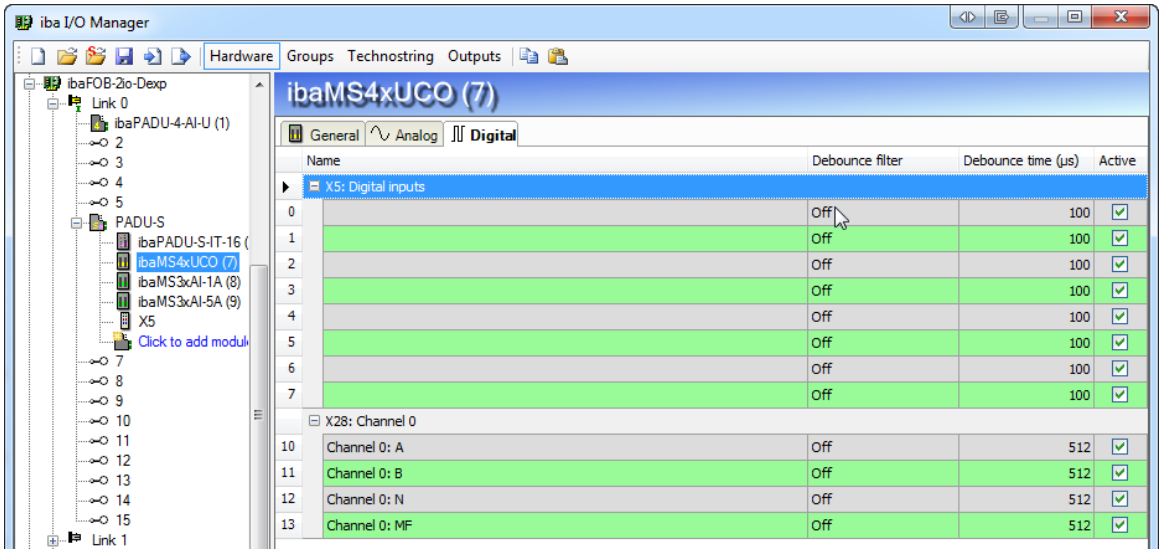
- SSI master

The modes you can select depend on the connected encoder type. Each mode will result in a different set of analog input signals. For more information about the different modes consult the ibaMS4xUCO hardware manual.

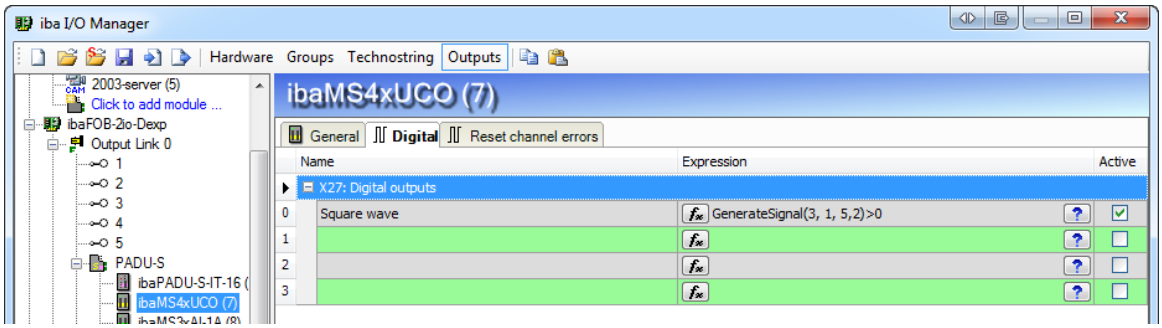
The last property for a counter channel is “Show encoder inputs”. If this is enabled then the raw A, B, N and MF digital signals of the counter channel can be measured as well.



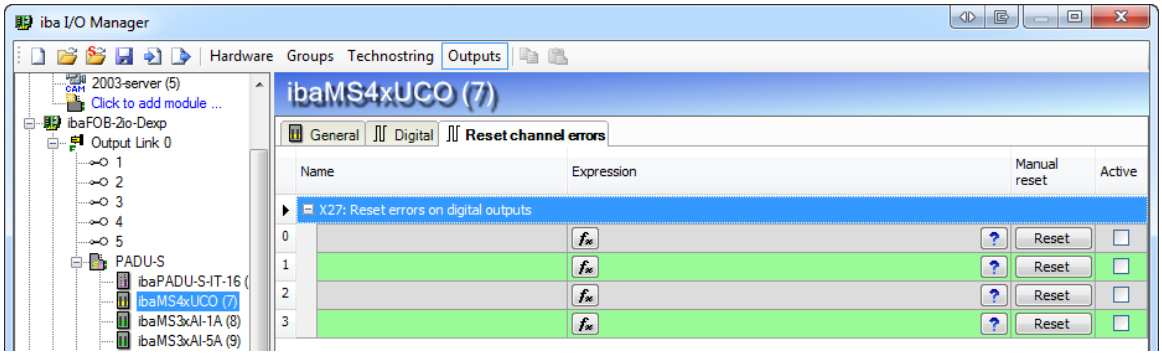
The analog inputs are grouped by counter channel. In the screenshot you see the signals for pulse counter mode and pulse width mode.



You can configure a debounce filter on the digital inputs of connecter X5. You have to select the debounce filter mode and the debounce time. The hardware manual describes these different debounce modes. The debounce filter can’t be set for the raw counter digital signals.



On the outputs tab you can configure the 4 digital outputs of the X27 connector. You can enter an expression for the value of the output signal. The outputs are sent to the module maximum every 50ms. It can be slower if there are slower modules defined.

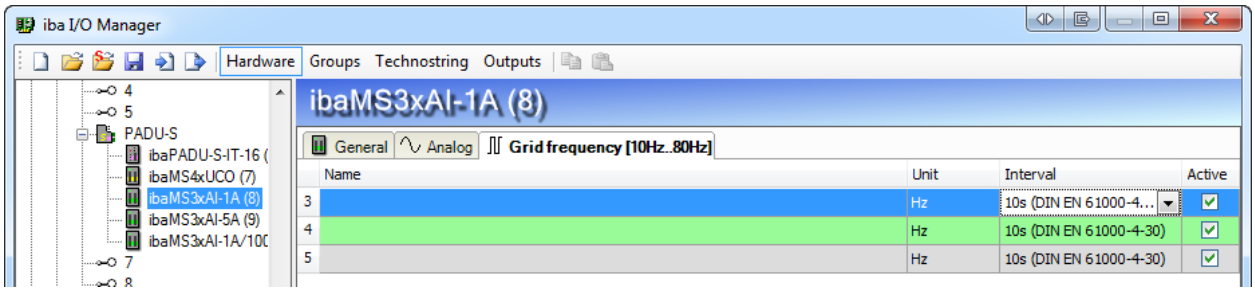


The outputs are protected against short circuit. If this is detected then they are switched off automatically. In order to re-enable them you can restart the acquisition or you press the Reset button on the “Reset channel errors” tab. You can also reset it automatically via an expression. You could e.g. create a QPanel button to reset the channel errors.

4.2 ibaMS3xAI-1A and ibaMS3xAI-5A

The ibaMS3xAI-1A and ibaMS3xAI-5A are modules that have 3 analog current inputs. Each one has a different input range.

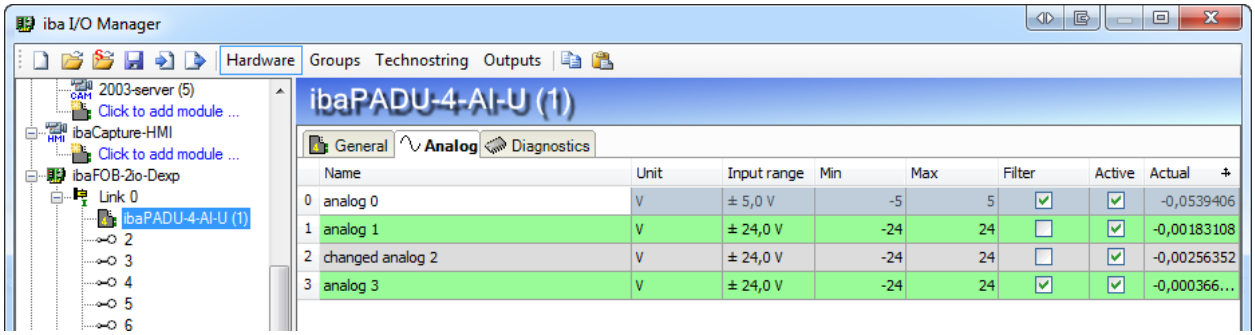
- ibaMS3xAI-1A: 1A effective current or ±3A peak current
- ibaMS3xAI-5A: 5A effective current or ±15A peak current



They can also optionally perform a frequency measurement of the current signals.

4.3 ibaPADU-4-AI-U

The ibaPADU-4-AI-U is a PADU with 4 analog inputs that can measure at 100kHz. It uses the bidirectional 32Mbit/s flex protocol.



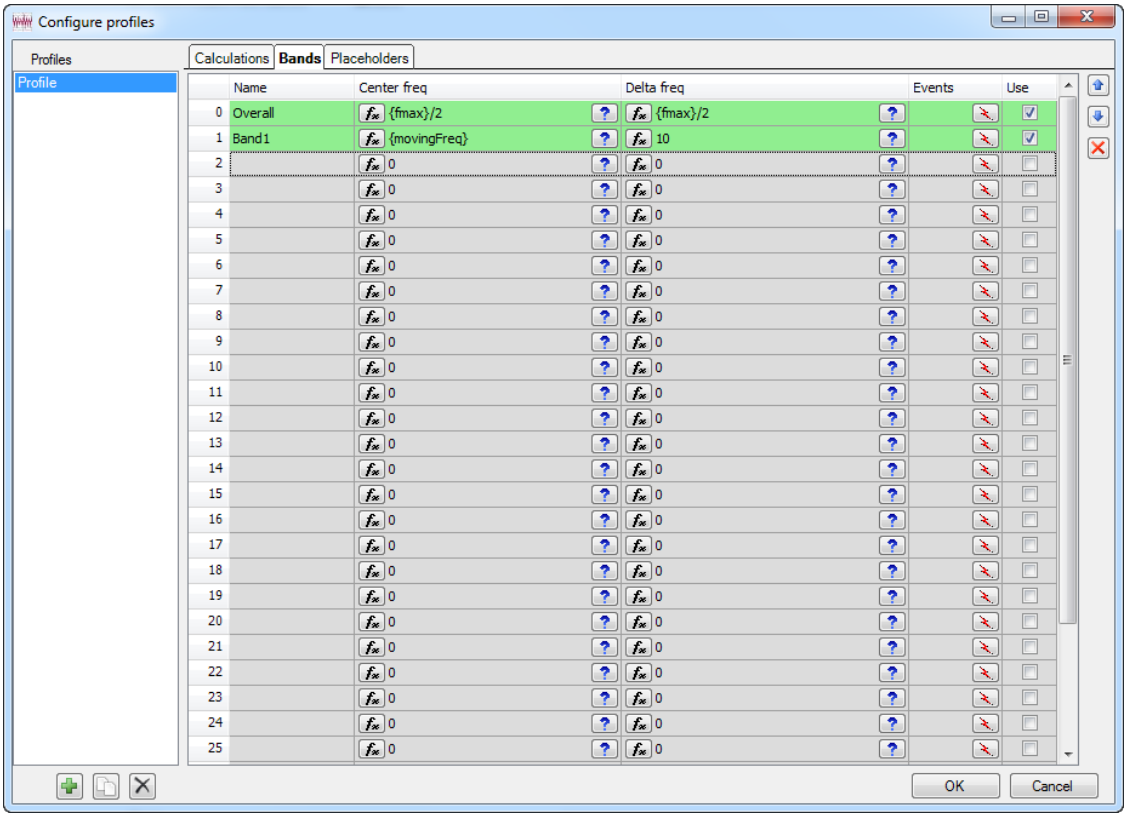
In ibaPDA you can define the input voltage range per channel. You can choose between ±0.25V, ±0.5V, ±1.0V, ±2.5V, ±5.0V, ±10.0V and ±24.0V. You can also enable an anti-alias filter per channel. This filter is automatically adjusted in relation to the sample frequency. After you applied the configuration you will see the actual values in the last column of the signal grid.

5 InSpectra Expert

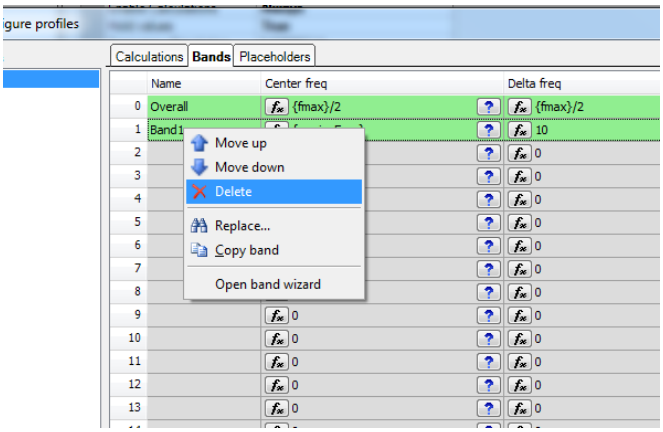
5.1 Moving, adding and deleting bands

In the 6.30 version, the number of bands was limited to 32. Also, one could not remove or move bands. Now, bands can be moved and deleted and you can add up to 512 bands.

You can move or delete bands by using the buttons on the right:



Alternatively you can use the context menu:



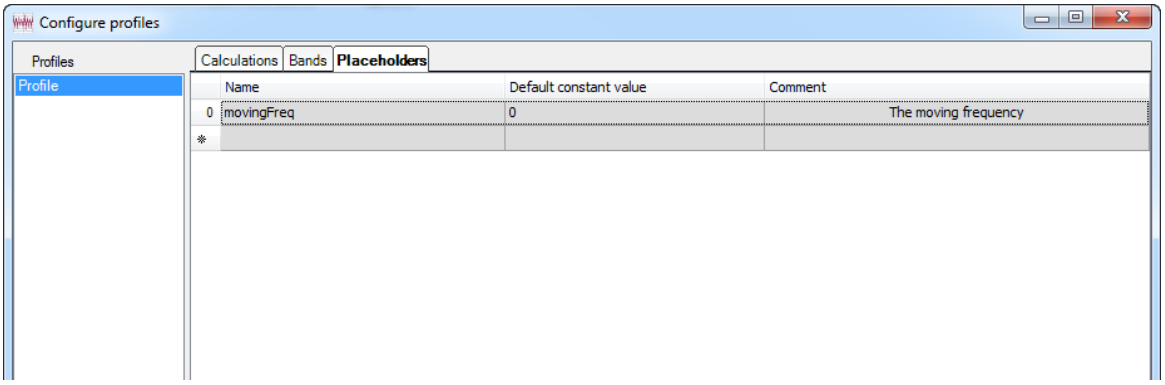
Note that moving or deleting bands can cause the band numbers of some other bands to change. This will also affect the signal ids of the analog and digital signals related to these bands.

It is possible to select multiple bands by using SHIFT and CONTROL and move or delete them simultaneously.

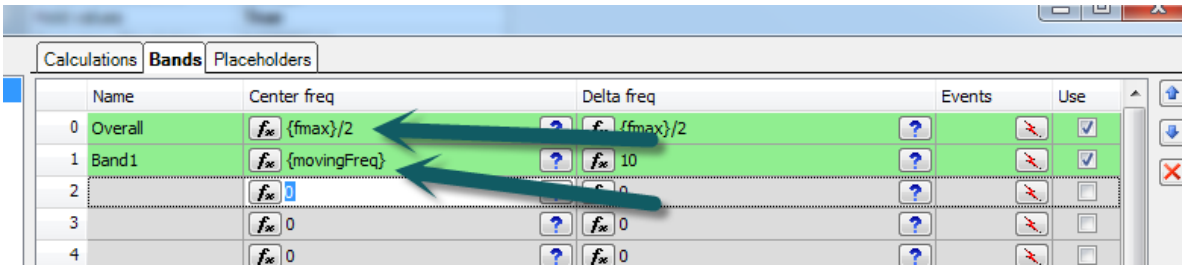
5.2 Placeholders

From version 6.30.1 on, placeholders are supported in the profile of an InSpectra Expert module. The expressions used in the InSpectra Expert band definitions often depend on device specific parameters such as speeds, diameters, sizes, etc. Placeholders facilitate the reuse of an InSpectra Expert profile to monitor similar devices each having their own parameters. For each InSpectra Expert module, you can set each placeholder either to a constant value or a signal.

The placeholders of a profile can be defined in the most right TAB of the profile dialog:



The placeholder will be replaced by the default value if the user omits to set the value of the placeholder for some InSpectra Expert module using the profile. The text in the Comments field should explain the meaning of the placeholder and it is displayed when the user assigns the placeholder for some module. A placeholder can be used in the expressions of a profile by using curly brackets:



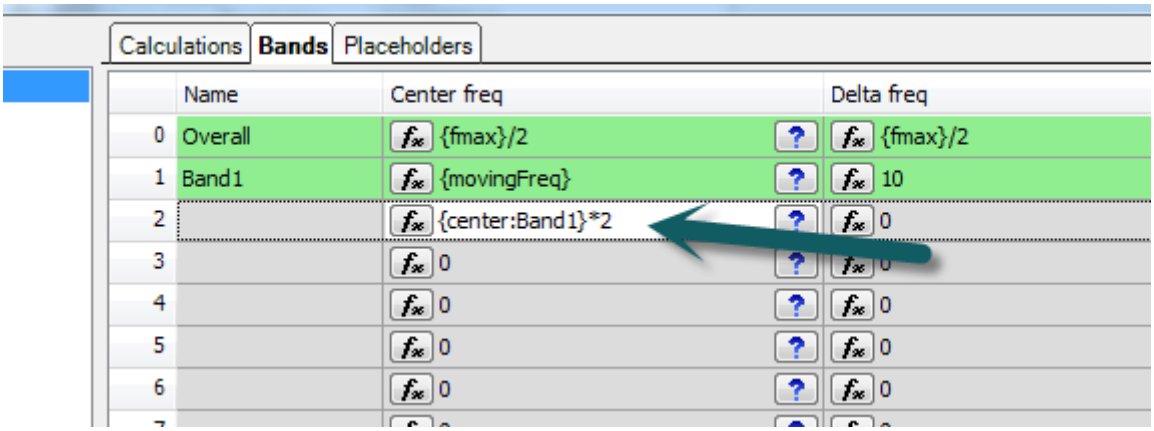
The {fmax} placeholder is a special placeholder; it always refers to highest frequency that is supported by InSpectra Expert, this is more or less equal to the sample frequency of the input signal divided by 2.56.
The placeholder can also be used in the expressions of the events. You can also type a valid expression containing a placeholder not yet defined. If the user subsequently navigates to the Placeholders TAB, then this placeholder is automatically added to the placeholders list.

5.3 Band placeholders + band wizard

From version 6.31 on, you can use band placeholders in each expression of an InSpectra Expert profile. A band placeholder refers to the center or delta value of another band. There are two ways to refer to a band:

- By band number: e.g. {center:0} or {delta:0}
- By band name: e.g. {center:bandName} or {delta:bandName}

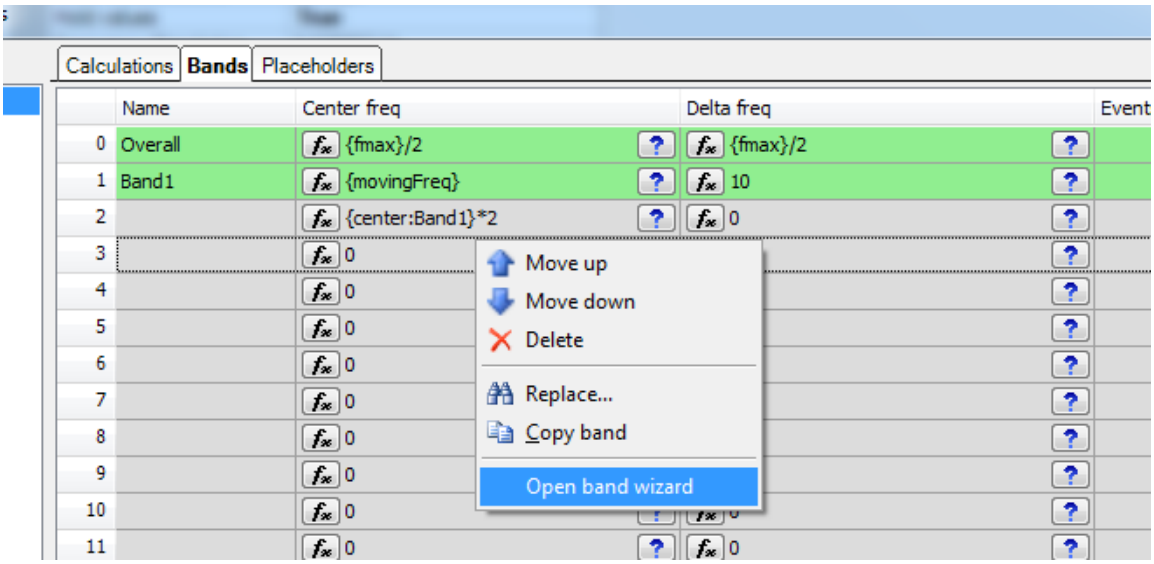
Band placeholders make it easy to define harmonic bands:



If you refer to a center or delta value of another band, the calculated band value is reused multiple times. Alternatively, you could copy an expression several times and multiply it with different constants each. The latter approach however is not efficient as the copied expression will be calculated multiple times.

If you move or delete a band, then the band number of some bands can change. The band numbers used in the band placeholders are updated automatically.

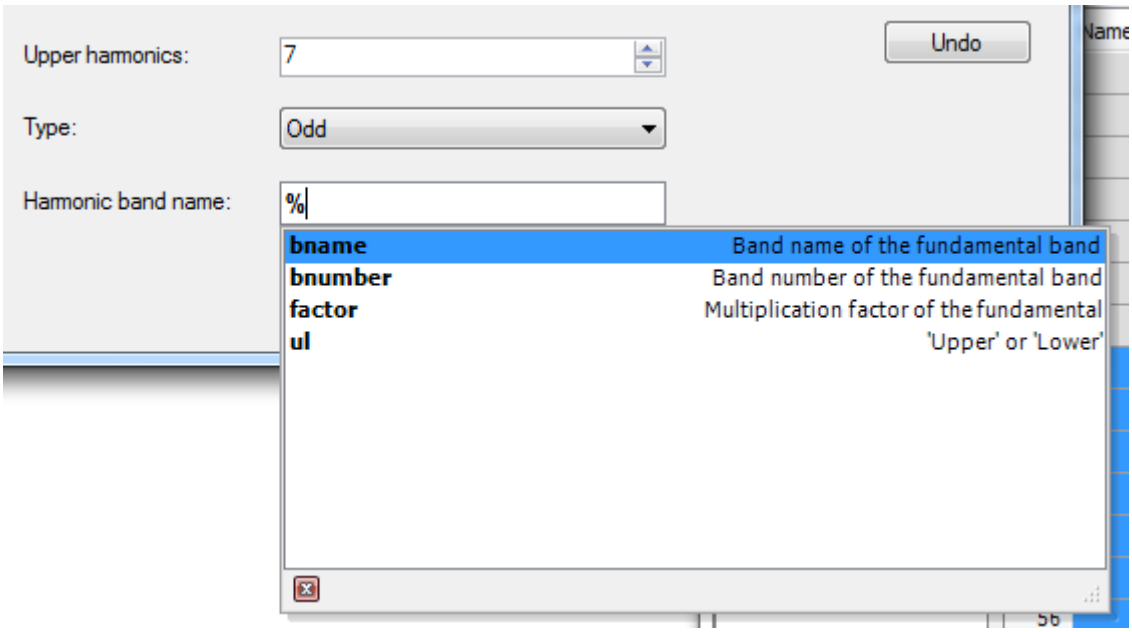
A band wizard was added to facilitate the generation of harmonic bands. You can open this wizard through the context menu of a band's row:

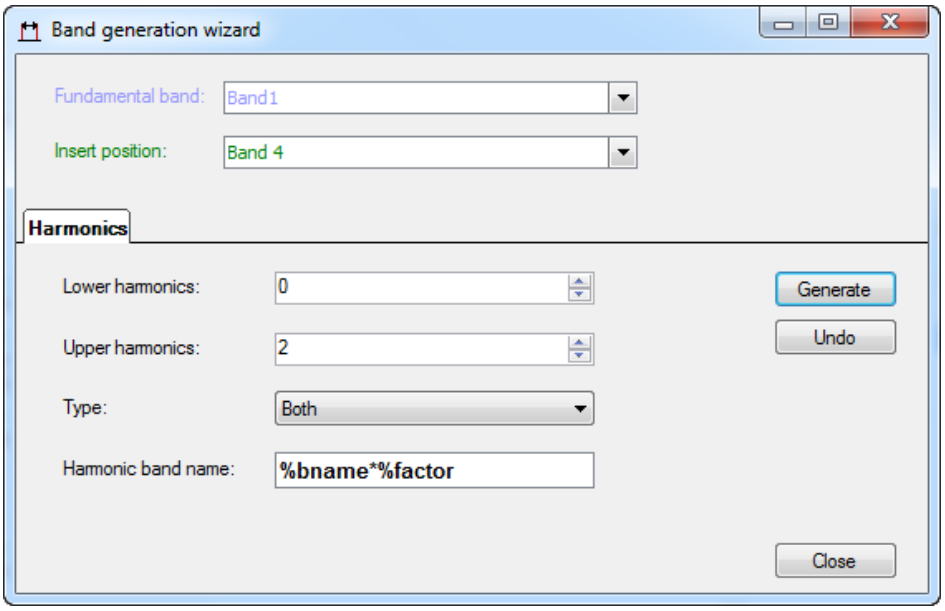


The wizard allows generating multiple harmonics at the same time by pressing the Generate button. You can define:

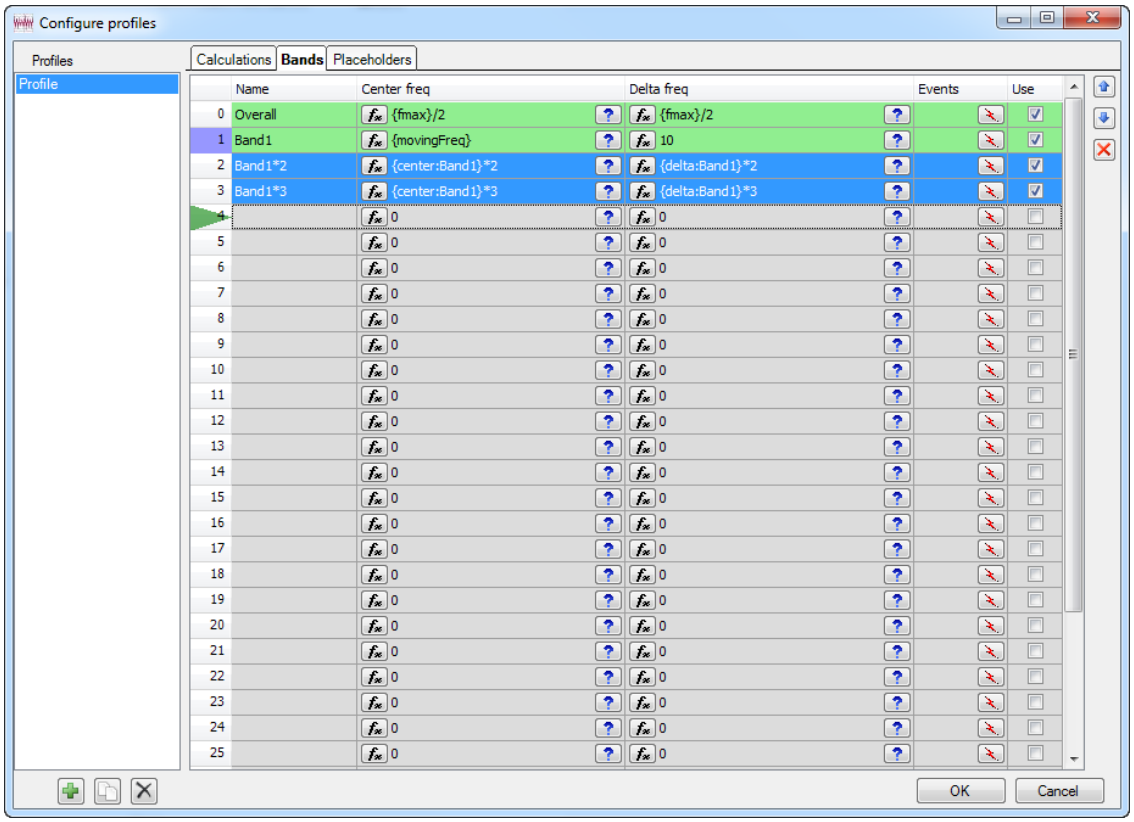
- the number of lower harmonics
- the number of upper harmonics
- type of harmonics: only **even** harmonics, only **odd** harmonics, **both** harmonics
- the name of the band to generate

To define the name of the harmonic bands, you can use some special placeholders. By combining these placeholders, you can give the correct name to each of the harmonic bands. Placeholders here start with a % character. The following placeholders are available:



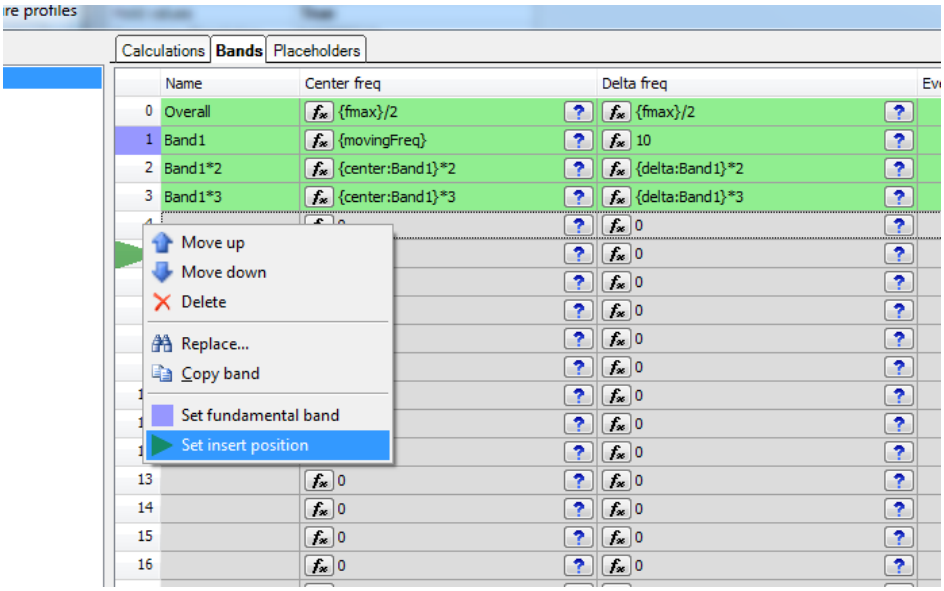


After clicking on the Generate button, you get the following result:



At the top of the band wizard, there are two dropdowns. The fundamental band is the band on which the harmonics are based. The position is the position where the harmonic bands are inserted if the Generate button is pressed. The fundamental band is indicated with a blue zone,

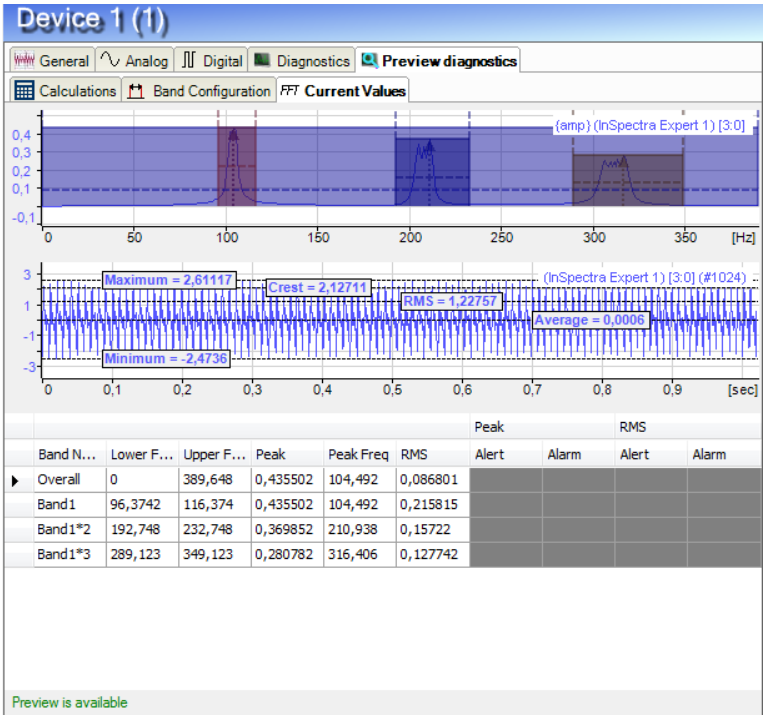
the insertion band is indicated by a green arrow. You can also set the fundamental and the position through the context menu of a band:



The parameters chosen in the band wizard are saved in the registry. With the Undo button you can undo the insertion of the generated harmonics. You can change the selected profile in the Profile Dialog while the band wizard is open.

5.4 Preview diagnostics

The preview diagnostics TAB was added to allow previewing the FFT and the bands of a module in case the acquisition is already running. If the acquisition is running, you change one or more settings in the module and then go to the preview diagnostics TAB to preview the effect of the changes:



More specifically, the Current Values sub TAB shows a live visualization of the FFT together with the band and event values. The two most left sub TABS contain the calculation and band settings used for the preview diagnostics.

The preview calculations are not always available. For instance, if you change the input signal of the module to a signal that is currently not active at the server, then no preview can be calculated.
The text at the left bottom of the Current Values sub TAB indicates whether the preview is available. Note that it can take a while for the current values of the preview to appear. Once enough samples of the input signal are buffered, the first result is displayed.

The live results shown in the preview diagnostics are only a preview, they do not affect the acquisition and so they are not stored in any dat file. For the changes visualized in the preview diagnostics to take effect at the server, one needs to reapply the complete I/O Manager configuration.

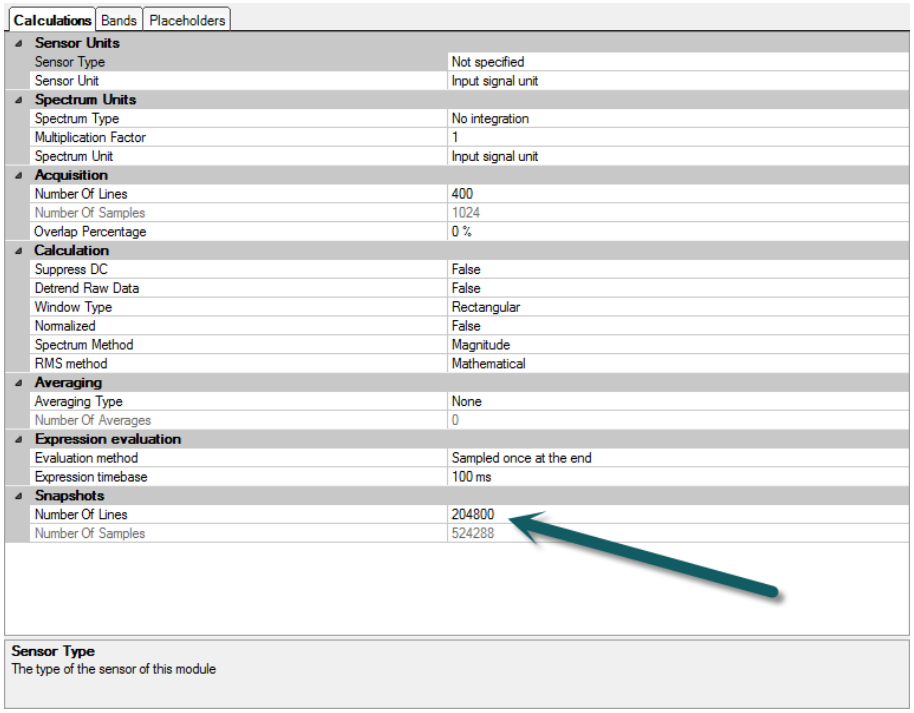
5.5 Snapshots

Next to the normal operation of an InSpectra Expert module, it is now possible to calculate an additional snapshot FFT with a very high resolution (very large number of samples, up to 1 million). For this FFT, the band values are also calculated, but the event values are not. You can configure the InSpectra Expert module to do these additional calculations periodically or based on the rising edge of a signal. The results cannot be visualized in ibaPDA, but are exported to two separate dat files. Each FFT corresponds to two dat files. The first file contains the input samples used for the FFT; the second one contains the FFT together with the values of the bands. These dat files can be opened in ibaAnalyzer.

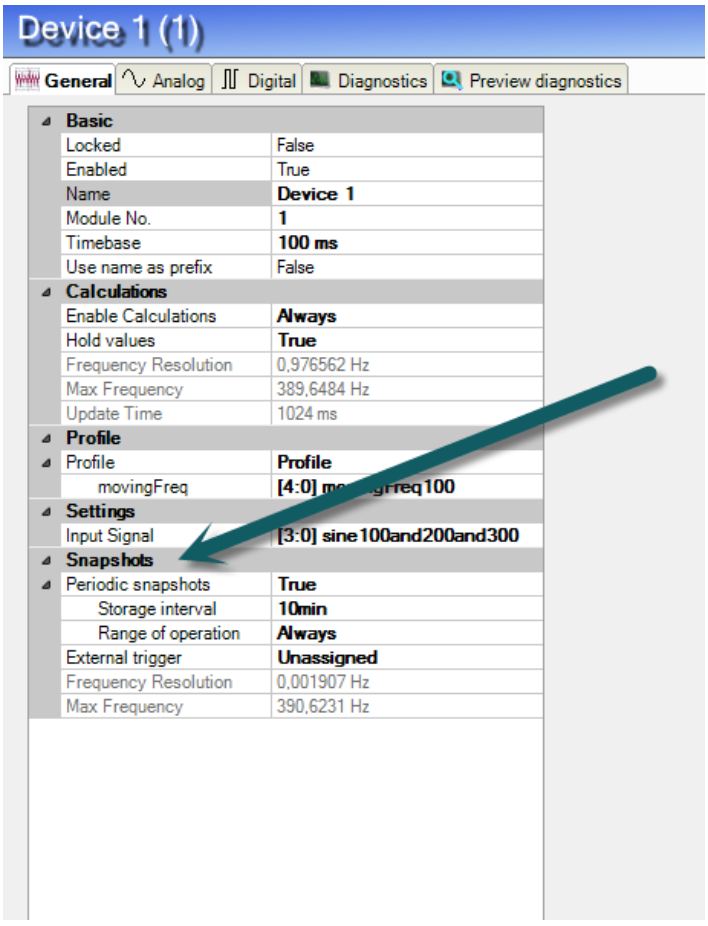
This snapshot dat file functionality is very specific as it was created to exchange high resolution FFT data between PDA and the HAICMON Server. Later, we will add the functionality to create more generic dat files containing the normal output of the InSpectra Expert modules.

When a snapshot is initiated this means the server starts buffering the input data for the FFT. Once the buffering process is completed, the FFT and the bands values are calculated and the two dat files are created.

InSpectra Expert has some configuration settings to determine
- The number of samples for such snapshot FFT:



- When a snapshot is initiated:

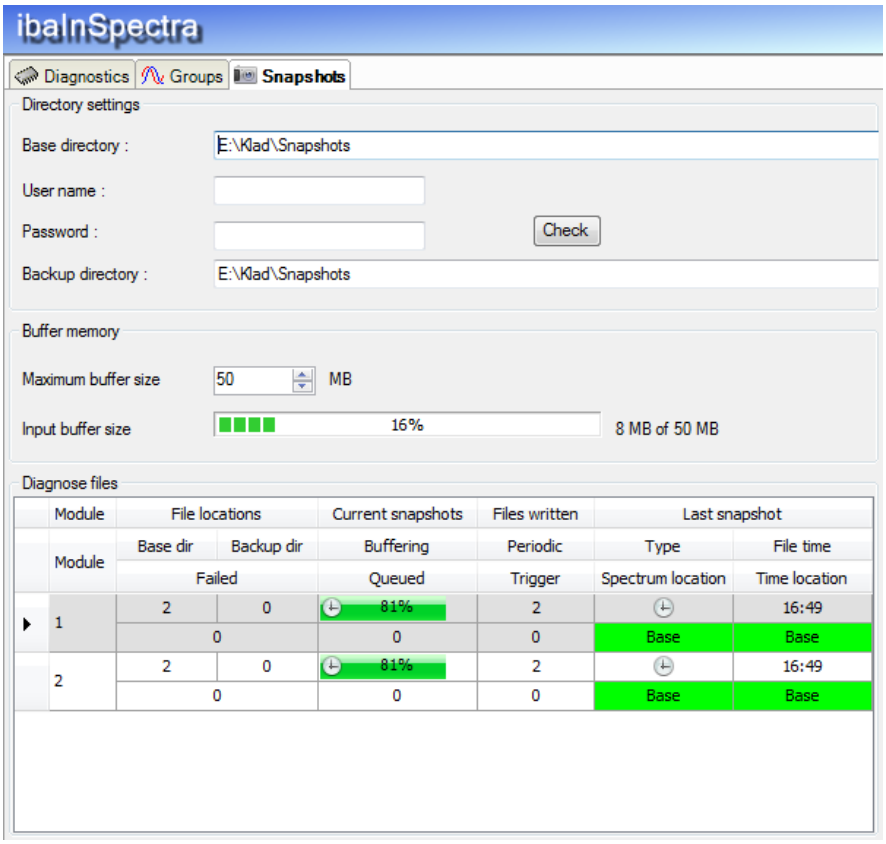


You can put periodic snapshots on or off. If they are on, you can set the storage interval (= period) by combining days, hours and minutes. If the period is one hour, then a first snapshot will be initiated (= start of the buffering for the FFT) at the start of the acquisition and thereafter every hour.

The range of operation signal is optional. Its default value is Always. When it is set to a digital signal, then this signal must be high to allow the buffering for the FFT to begin. If it is low or becomes low while buffering for a snapshot, all buffering processes are reset and the snapshots are delayed until the range of operation signal is high again. This means that the periodic snapshots can drift away in time. So, the storage interval setting (= period) must be considered as an indication for the system, not as a hard rule. Typically, the range of operation indicates whether the monitored plant or device is running or not. If not, we shouldn't calculate the FFT because it does not contain any useful information.

The external trigger can be set to a digital signal. A snapshot is initiated when this digital signal rises from low to high. The range of operation setting is not applicable here; it is only applicable for periodic snapshots.

The location of the snapshot dat files is the same for all InSpectra Expert modules. It can be set in the directory settings in the Snapshots TAB of the ibaInSpectra interface:

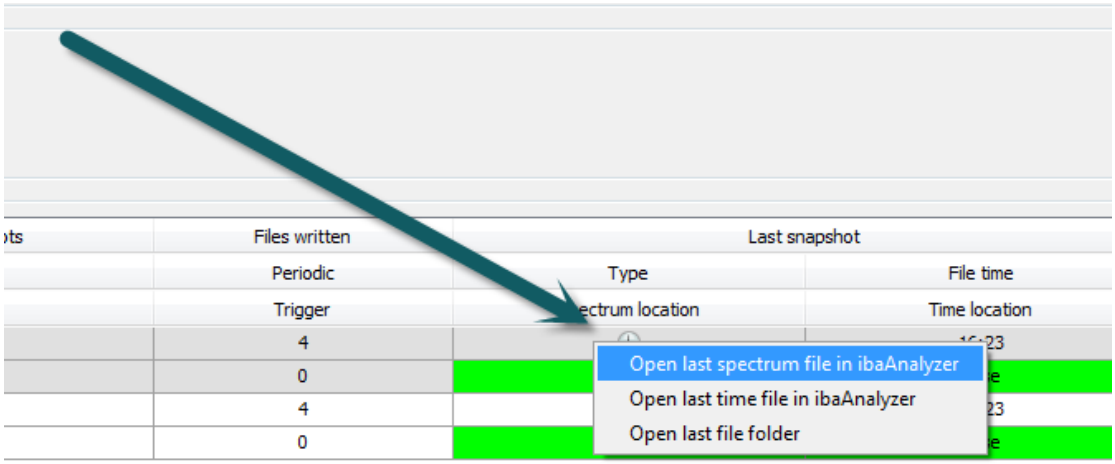


The base directory is the directory where the dat files are written to. If this is a network location, there are two fields to provide the username and password. You can set the backup directory to a local path only.

The maximum buffer size provides an upper limit for the memory used by the snapshot buffering processes of all InSpectra modules together. With this setting we want to prevent that the ibaPDA server uses too much memory if a lot of InSpectra Expert modules are buffering at the same time. This memory bound can cause one or multiple snapshots to be delayed!

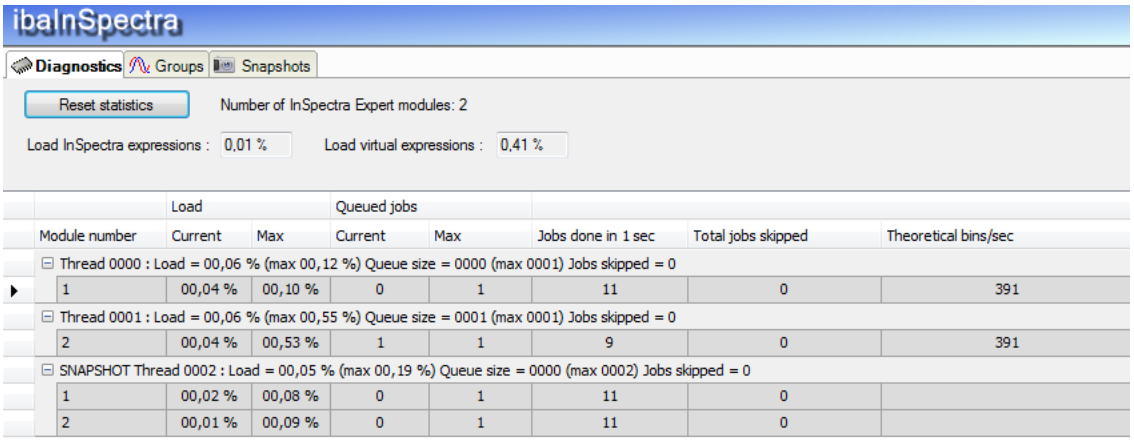
The input buffering size progress bar indicates the current memory use for buffering in case the acquisition is running.

In the table at the bottom, you can monitor the current buffering processes in case the acquisition is running. To interpret the table it is important to know that each snapshot generates two files. You can open the files of the last snapshot directly in ibaAnalyzer through the context menu.



5.6 Diagnosing the load of InSpectra Expert calculations

In the ibaInSpectra interface you can monitor the load of the InSpectra Expert calculations on the server:



	Load		Queued jobs				
Module number	Current	Max	Current	Max	Jobs done in 1 sec	Total jobs skipped	Theoretical bins/sec
Thread 0000 : Load = 00,06 % (max 00,12 %) Queue size = 0000 (max 0001) Jobs skipped = 0							
1	00,04 %	00,10 %	0	1	11	0	391
Thread 0001 : Load = 00,06 % (max 00,55 %) Queue size = 0001 (max 0001) Jobs skipped = 0							
2	00,04 %	00,53 %	1	1	9	0	391
SNAPSHOT Thread 0002 : Load = 00,05 % (max 00,19 %) Queue size = 0000 (max 0002) Jobs skipped = 0							
1	00,02 %	00,08 %	0	1	11	0	
2	00,01 %	00,09 %	0	1	11	0	

The calculations for InSpectra can be split up into two parts:

- The evaluation of the expressions of the InSpectra expert profiles

- The calculation of the FFT's

The evaluation of the expressions normally uses the most resources. At the top of the Diagnostics TAB, you can monitor two values:

- **Load InSpectra expressions**: This is the time (in percent) spent by the ibaPDA server's main thread to evaluate the InSpectra Expert band and event expressions. This value depends on the complexity of the band and event expressions, on the number of InSpectra Expert modules and on the time base of the expressions. To lower the load, you can increase the "Expression timebase" in the Calculation settings of an InSpectra Expert profile. Note that the "Expression timebase" is inversely proportional to the time accuracy of the expression evaluation. Preferably, the "Expression timebase" is lower than the update time of the module (you can read the update time in the general TAB of an InSpectra Expert module).
- **Load virtual expressions:** This is the time (in percent) spent by the ibaPDA server for evaluating the virtual expressions from the virtual modules. This value depends on the complexity of the virtual expressions, on the number of expressions and on the time base of the expressions. To lower the load, you can increase the timebase of one or more virtual modules, but this will lower the time accuracy of the virtual signals.

Though calculating an FFT does not use many resources, the system can calculate several FFT's for different InSpectra Expert modules at the same time (multithreaded). In case snapshots are enabled for some module, there is one additional thread for the snapshots only. Each row in the table corresponds to an InSpectra Expert module. InSpectra Expert modules using the same thread are grouped. Modules using snapshots have two rows.

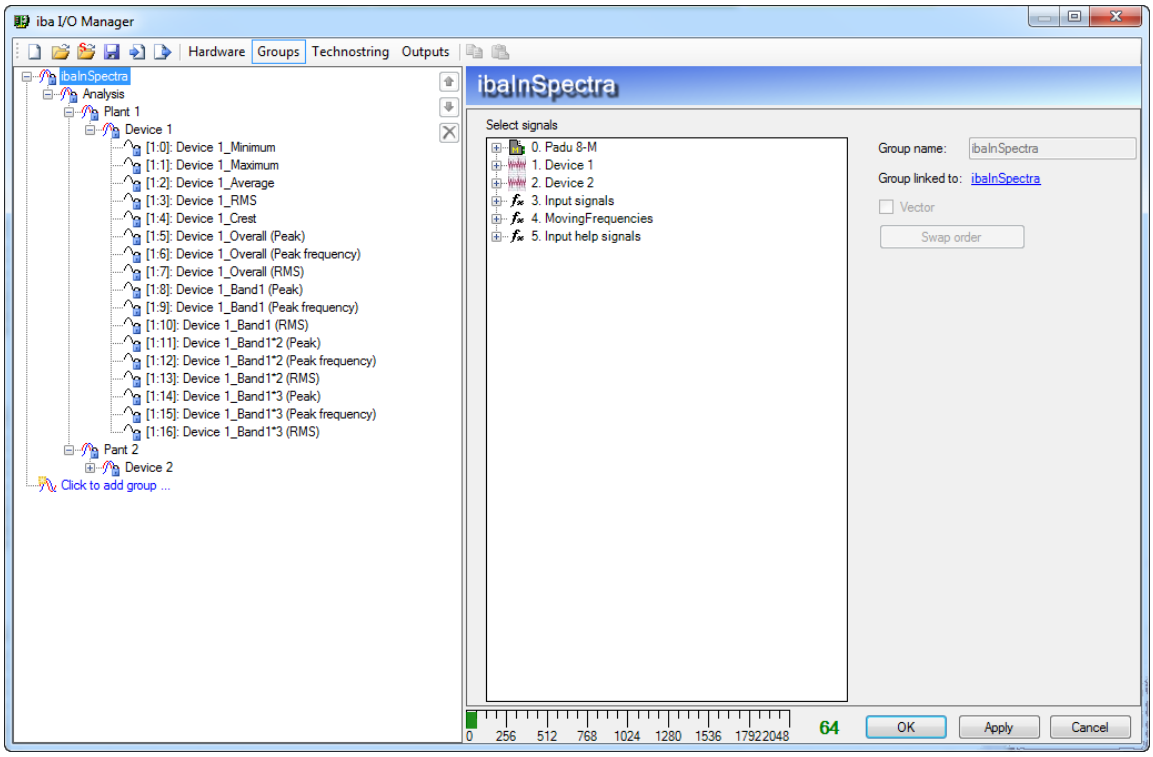
The two load columns indicate the time in percent used by the thread to calculate FFT's. The "Current" column gives the actual percentage; the "Max" column gives the maximum percentage. It is the maximum since the acquisition was started, or, in case the "Reset statistics" button was clicked, the maximum since the time of resetting. The "Queued jobs" column indicates the number of jobs queued. Normally, this number should be between 0 and 2. If it keeps increasing, this means that the server's CPU is overloaded. Note that each FFT consists of multiple jobs; the number of jobs for one FFT is dynamic, it can change during the acquisition. The "Total jobs skipped" column is important to diagnose any problems. If jobs get lost, then FFT's get lost. Jobs can get lost if the server is overloaded. The "Theoretical bins/sec" column can be used to better interpret the other row values. It contains the number of frequency bins to calculate per second. The percental value in the "Current load" column should be proportional to this number. In case of the snapshot thread, this column is not filled in.

With the "Reset statistics" button at the top you can reset the following statistics in the table: "Max load", "Max queued jobs" and "Total jobs skipped".

5.7 ibaInSpectra Groups

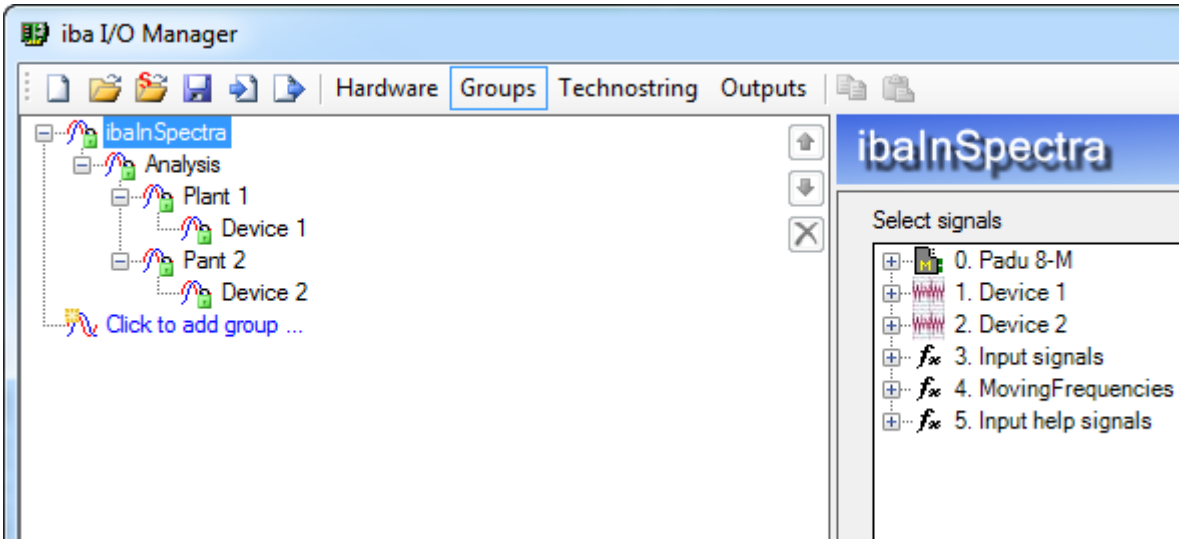
Already in version 6.30, a group structure of the ibaInSpectra interface tree was automatically created in the groups TAB of the I/O Manager. This functionality can now be customized:

You can choose whether these groups are generated or not. If so, the user can determine if the ibaInSpectra interface level is included as a top group. The name of this top group is also customizable. In the groups TAB, the groups that were automatically generated can be recognized by the small blue lock icon:

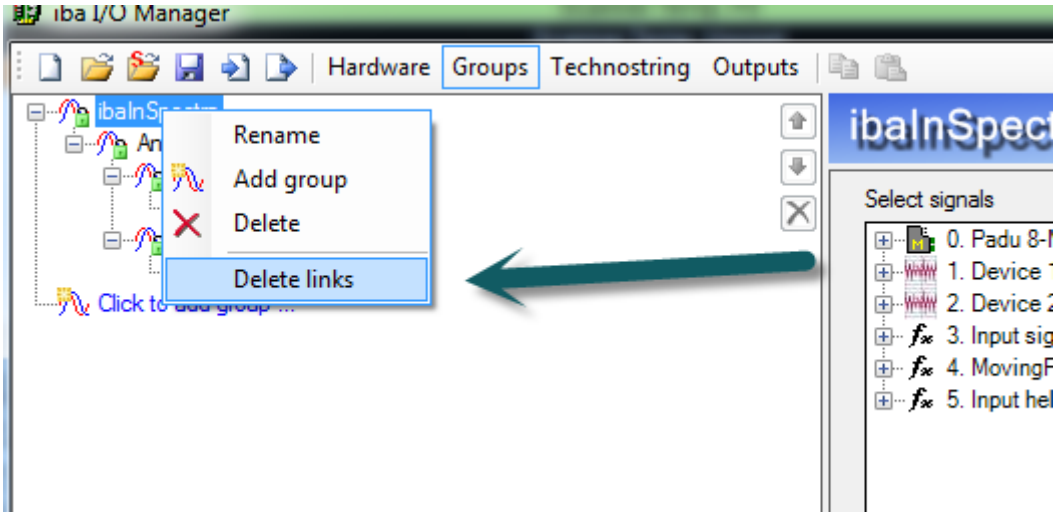


These locked groups cannot be moved, renamed or deleted. You can navigate to the object (i.e. interface, directory or module) that generated the group by clicking the blue link on the right panel. You can add other custom groups or signals to the automatically generated groups.

If the “Generate groups automatically” option is switched off, all automatically generated groups will disappear, except the ones containing groups that where added manually. These groups will remember to which object (i.e. interface, directory or module) they correspond. You will see a green lock icon:

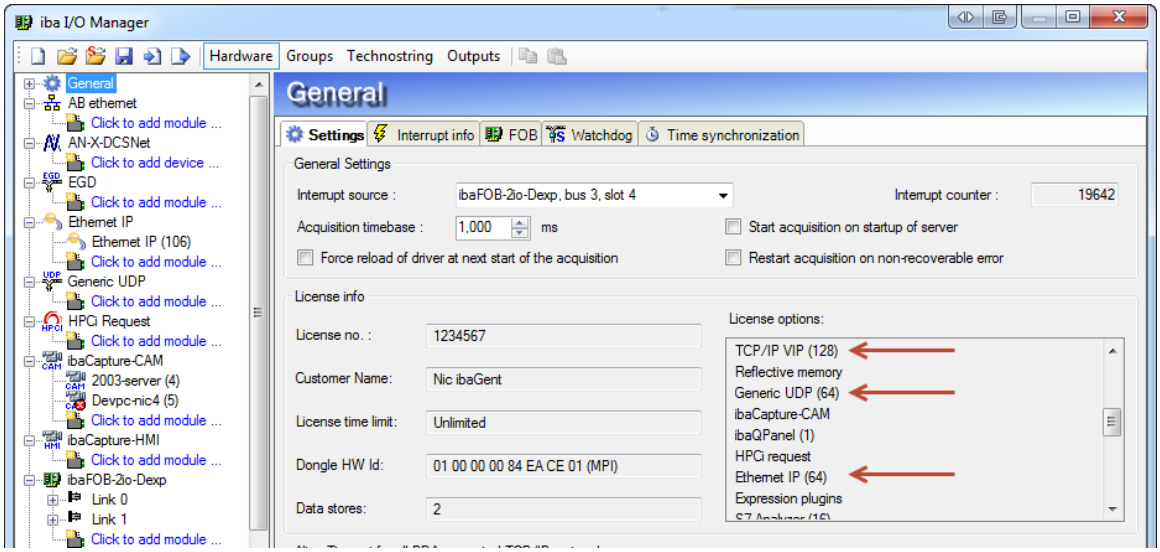


Moving such groups is inhibited. If you reactivate the “Generate groups automatically” option, the system will use the remembered links to reconstruct the previous group structure perfectly, even if some ibaInSpectra modules or directories were renamed in the meanwhile. You can remove the green lock (and so the link) by right clicking the group and choosing “Delete links”:



6 Network protocols changes

In versions before v6.31.0 all network protocols were limited to 64 connections. In 6.31.0 you can have up to 256 connections for TCP/IP VIP, TCP/IP S7, TCP/IP TDC, EGD, Ethernet/IP, Generic TCP and Generic UDP. For each license you buy for such a protocol you get 64 extra connections. In the license list in the I/O manager you can see the number of licensed connections you have for each protocol.

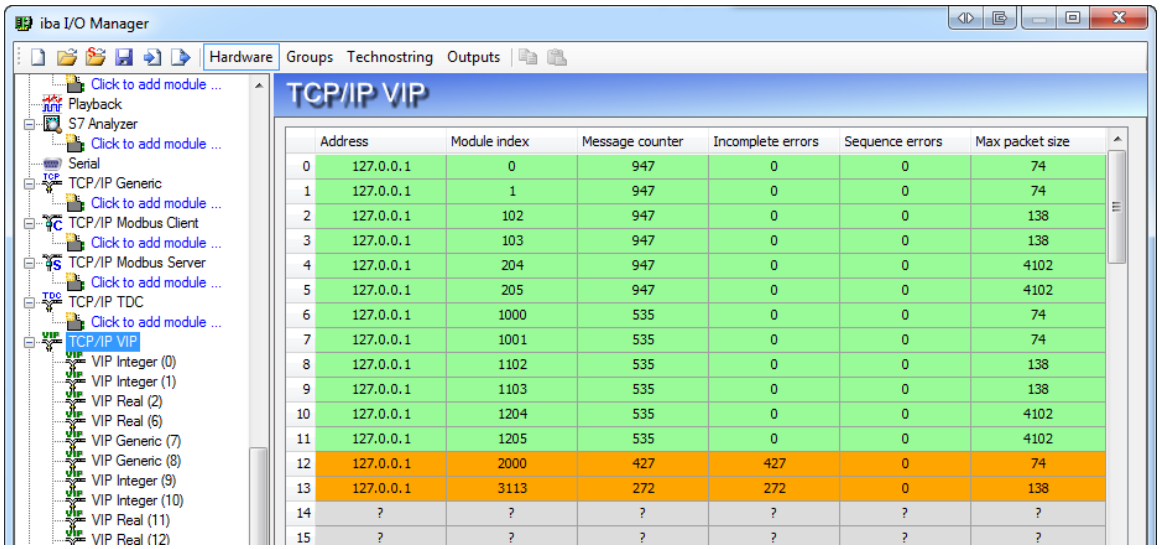


6.1 TCP/IP VIP, TCP/IP S7, TCP/IP TDC

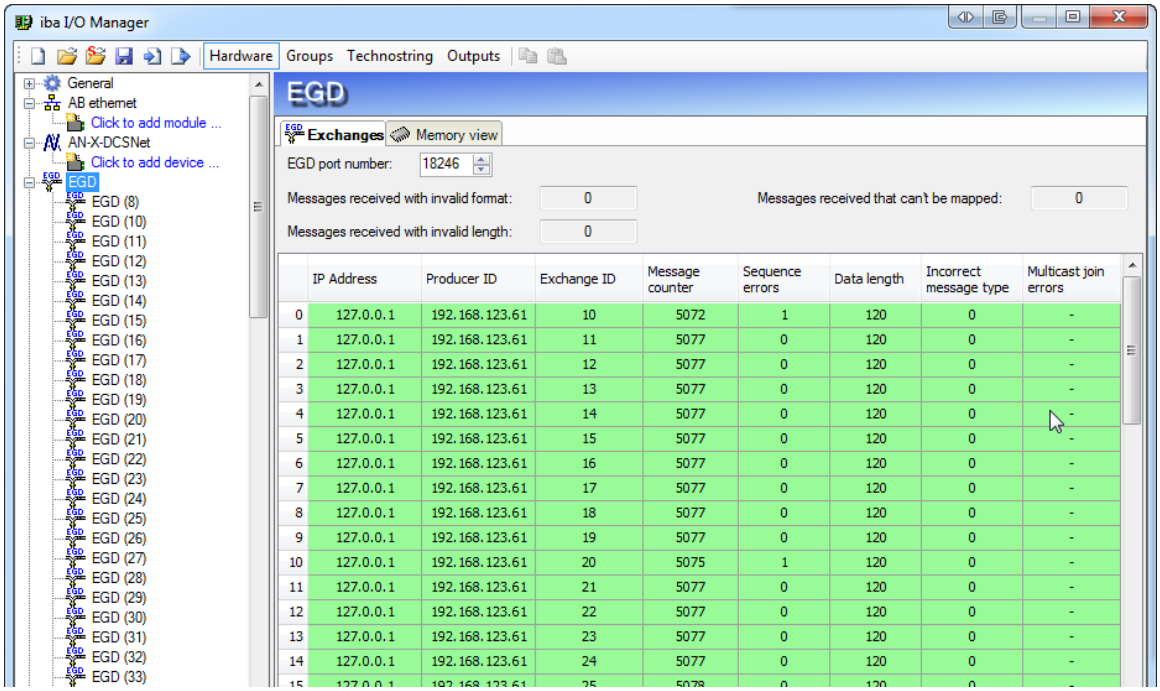
These three interfaces use the same network protocol. There are 3 types of modules that can be mapped to these interfaces: integer, real and generic modules. The integer modules have 32 analog signals with 16 bit integer data type. The real modules have 32 analog signals with real data type. The generic modules can have a user defined number of analog signals with user-selectable data types. In previous versions the maximum data size of a generic module message was 512 bytes. This has now been increased to 4096 bytes.

The message contains a “module index” that is used to identify the connection. The module indexes from 0 to 63 correspond with integer modules. The module indexes from 100 to 163 correspond with real modules and the module indexes from 200 to 263 correspond with generic modules. Per license of these interfaces you get 64 connections. The second group of 64 connections corresponds with the module indexes 1000-1063, 1100-1163 and 1200-1263. The third group corresponds with 2000-2063, 2100-2163, 2200-2263 and of course the fourth group corresponds with 3000-3063, 3100-3163, 3200-3263.

The following screenshot shows the connection table in the I/O manager. The green rows correspond with valid active connections. The orange rows correspond with active connections that are using a module index that isn’t licensed. Data from these connections can’t be measured. The gray rows correspond with empty connection slots. The connection table is sorted by module index.



6.2 EGD



For each EGD license you have you can make 64 EGD connections. There are no limits to producer ID or exchange ID. The connection table is now sorted by producer ID and exchange ID. You can double click on a row to go to the corresponding offset in the memory view.

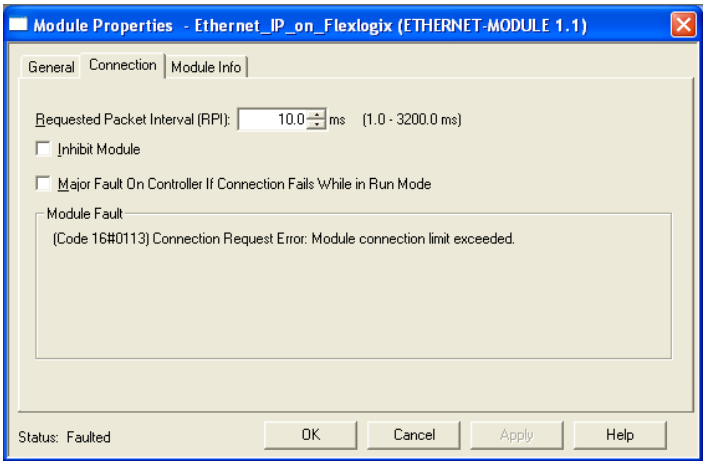
6.3 Ethernet/IP

For each Ethernet/IP license you have you can make 64 Ethernet/IP connections. The assembly instance must be within the range 1 to 64 for the first license, 65 to 128 for the second, 129 to 192 for the third and 193 to 255 for the fourth license. The assembly instance corresponds with the row number in the connection table. You can double click on a row to go to the corresponding offset in the memory view.

When a PLC tries to connect to an assembly instance higher than the number of licensed connections then you will see these messages in the the Ethernet/IP log file:

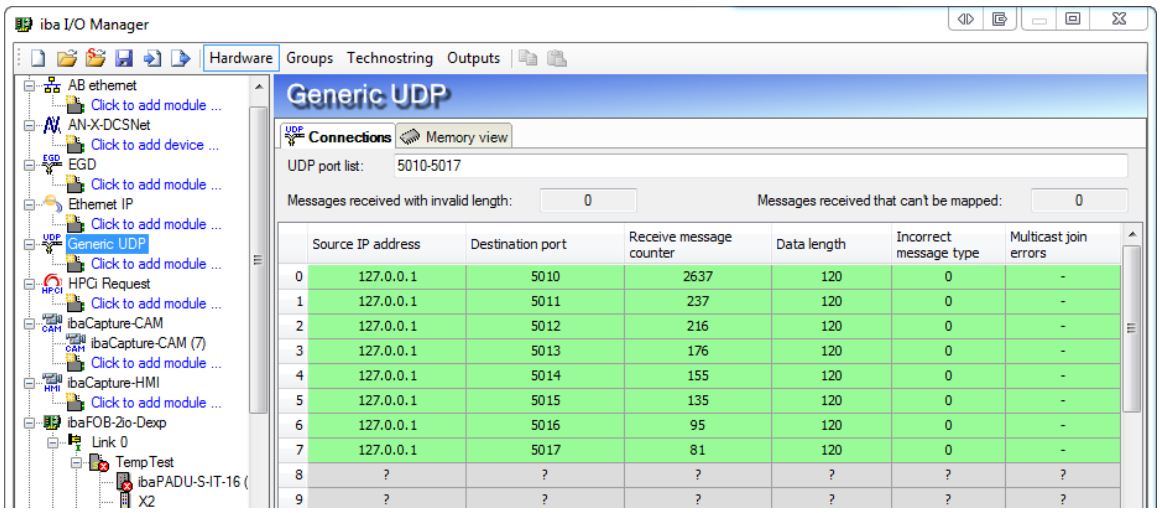
2013-08-28 13:43:13.990 [14: EthernetIP listening TCP thread] : **** ERROR **** : Forward_Open: Connection not established because driver didn't allow connection (The parameter is incorrect.
(0x00000057))
2013-08-28 13:43:13.990 [14: EthernetIP listening TCP thread] : **** ERROR **** : Forward_Open: Connection not established, ConnectionInstance could not created or configured

In RSLogix 5000 you will see a module fault: Module connection limit exceeded.



Each Ethernet/IP module now also has outputs. You can define the number of analog and digital outputs. For each output signal you have to define its value via an expression, its address and its datatype. Like all other outputs in ibaPDA these outputs are only updated maximally every 50 ms.

6.4 Generic TCP, Generic UDP

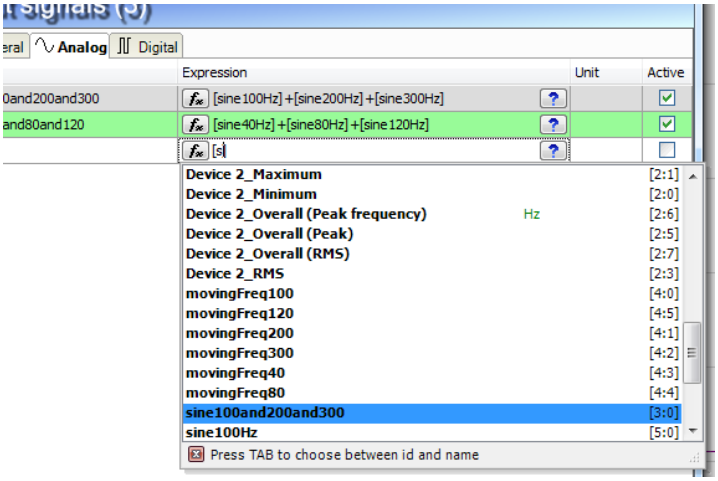


For each generic TCP or generic UDP license you have you can make 64 connections. The connections in the table are sorted by source IP address and destination port number. You can double click on a row to go to the corresponding offset in the memory view.

7 IntelliSense

You can use IntelliSense in all expression editors in the I/O Manager. The following expression components are supported:

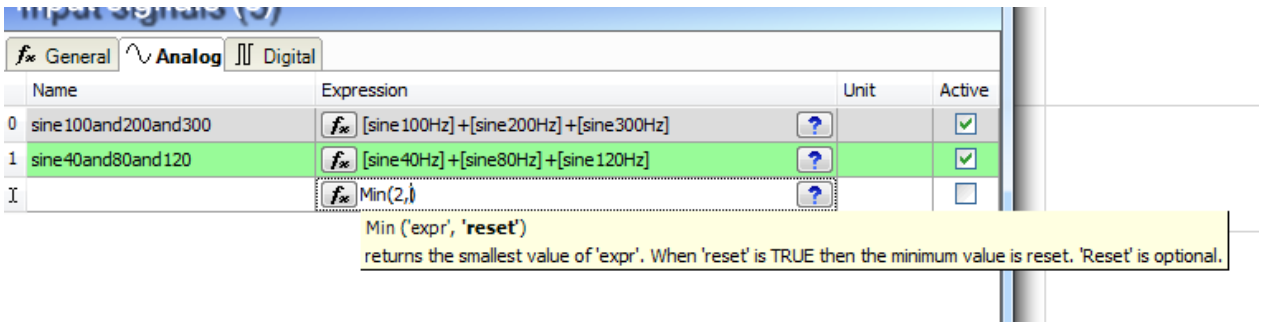
- Functions
- Signals: either by name or id (starting with a [character)
- Technostrings (starting with a “ character)
- Placeholders (starting with a { character)



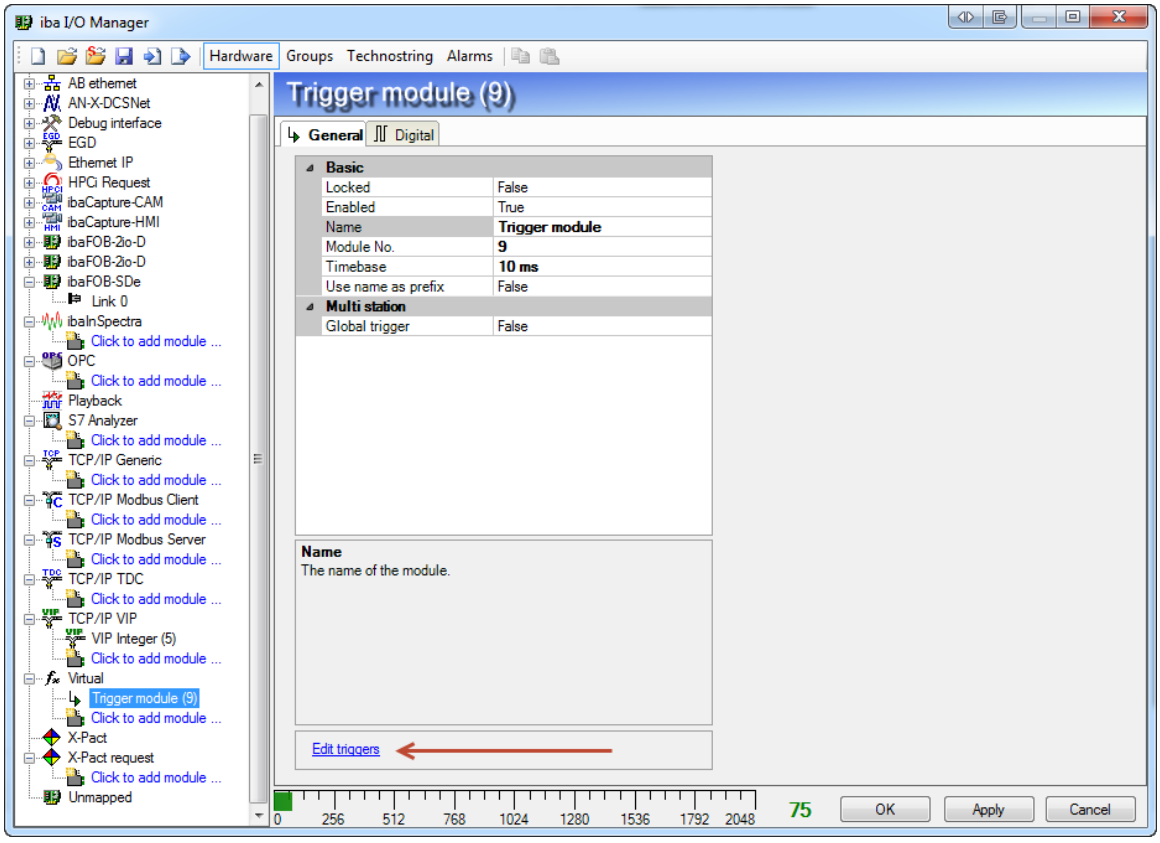
The IntelliSense drop down window appears as soon as you have typed the first character of the component:

- x to open the function IntelliSense window
- [x to open the signal id or signal name IntelliSense window
- “x to open the technostring IntelliSense window
- {x to open the placeholder IntelliSense window

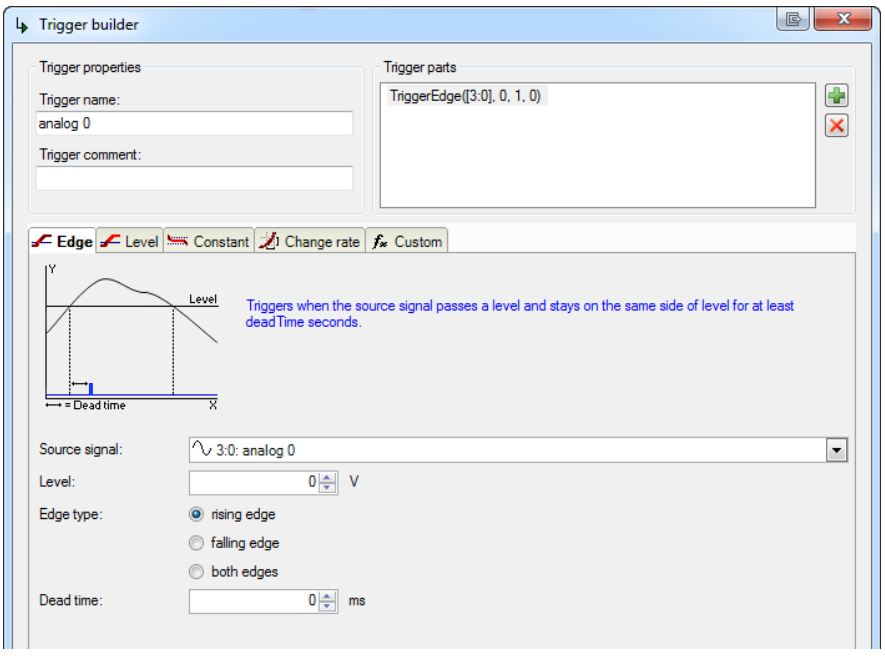
Alternatively you can press CONTROL + SPACE to open the IntelliSense window. The drop down window for signals also displays the unit of the selected signal. In case you are typing a signal name or id, you can switch between id or name by pressing the TAB button. In case you are filling in the arguments of a function, you will see a tooltip with the prototype of the function. This tooltip helps you to easily see what function arguments are needed at the cursor position of that moment:



8 Trigger module improvements



The trigger module now has a hyperlink on the general tab that allows the user to open the trigger expression builder to add multiple triggers.

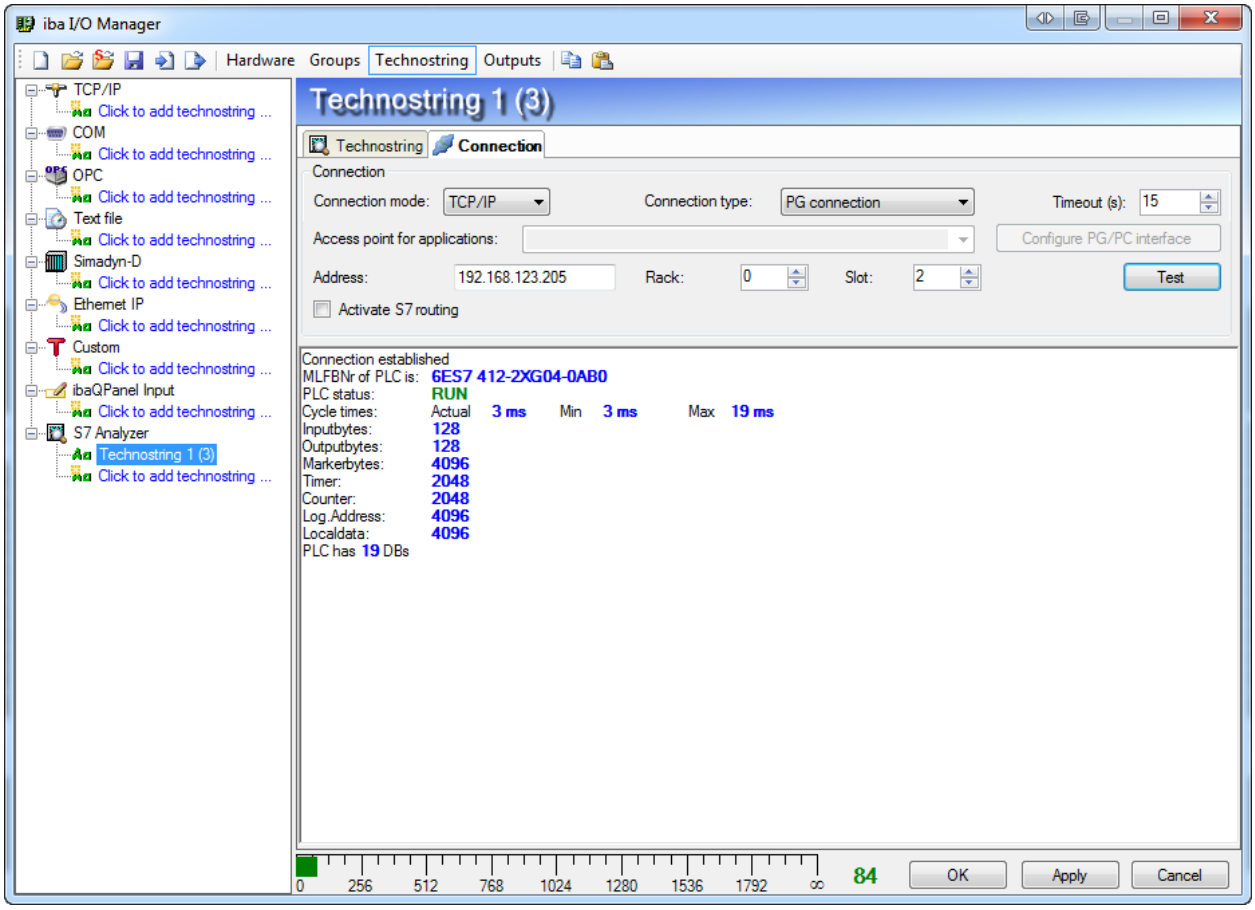


When the expression builder is opened a new trigger will be automatically created from the previous trigger. It will have the same type but will take the next signal as source signal. The

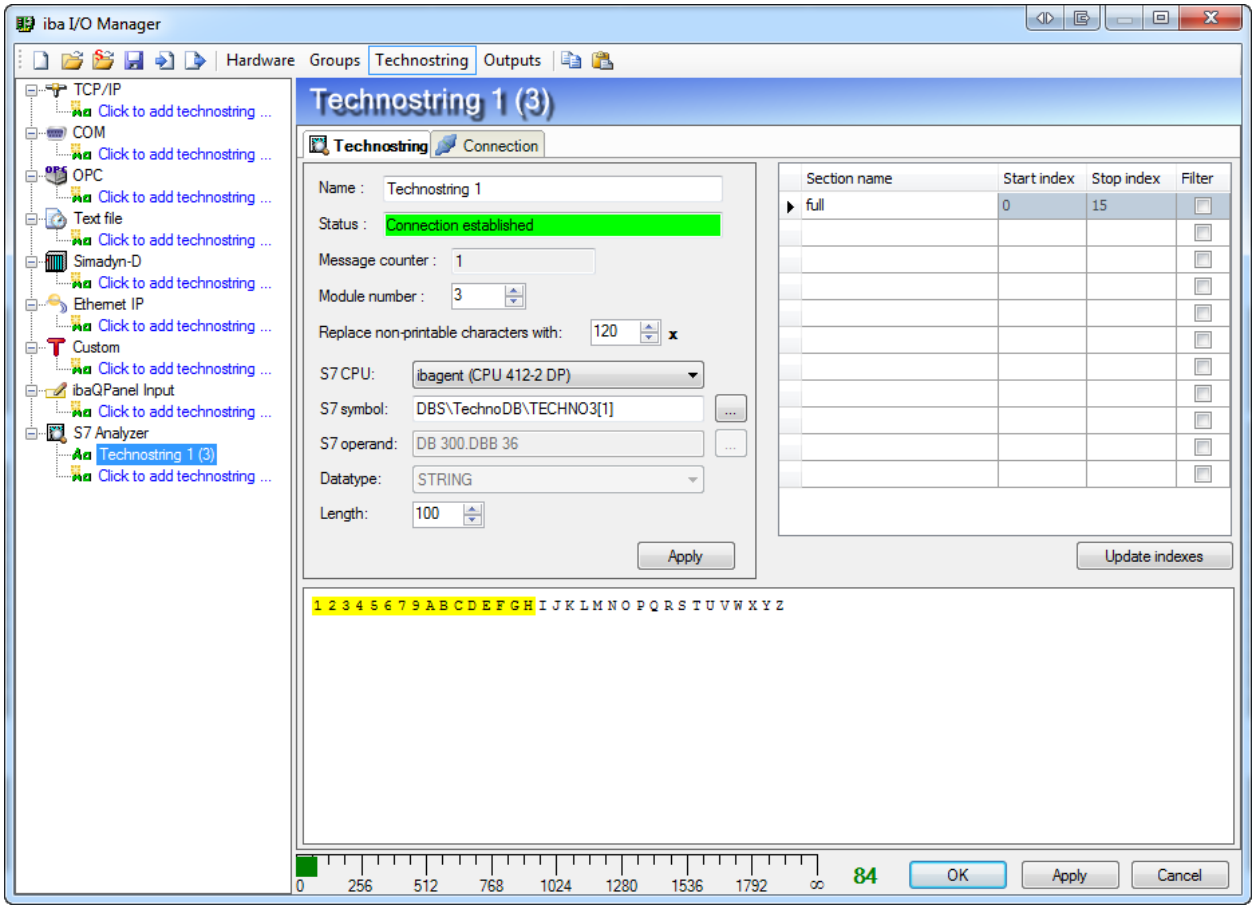
name of the trigger will also be automatically set to the name of the source signal. When you switch the source signal and the trigger name is still the default name then the name of the trigger will also change automatically. Each time you click OK the settings for the trigger will be applied and the next trigger will be selected. If you were at the last trigger then a new trigger will be added. At any time you can select another trigger in the signal grid and the trigger builder will update to that trigger.

9 S7 Analyzer Technostring

The S7 analyzer technostring can be used to read texts from a connected S7 PLC.



On the connection tab of the technostring you can configure how to connect to the S7 PLC. The configuration settings are the same as on an S7 analyzer module. You can also use the “Test” button to test the connection to the S7.



The technostring tab allows you to define which operand you want to read for the technostring. If you have created an addressbook for the connected S7 CPU via the S7 analyzer interface then you can select the CPU from the list and then use the symbol browser to select a symbolic address. When you select a symbol the corresponding operand, datatype and length are filled in. All datatypes except for the STRING datatype are converted to an ARRAY OF CHAR of the corresponding length. You can still change the length manually if you want to read another number of characters.

If you haven't selected an S7 CPU then you can manually enter an S7 operand. You will also have to choose the datatype. There are 2 options:

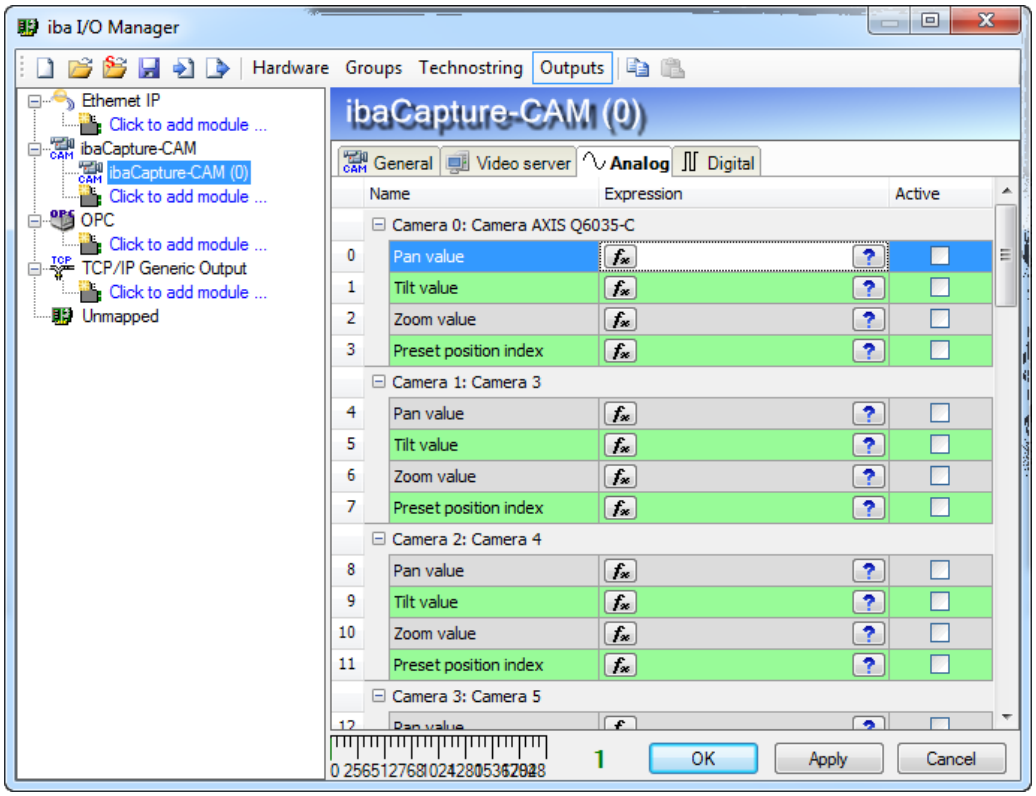
- STRING: This datatype consists of an integer containing the actual length of the string followed by the string characters. IbaPDA reads length + 2 bytes from the S7 and uses the actual length as the size of the text.
- ARRAY OF CHAR: This datatype just contains the string characters without a length.IbaPDA reads length bytes from the S7 and it checks for a terminating zero to get the length of the text. If no terminating zero is found then it just takes all bytes as the text.

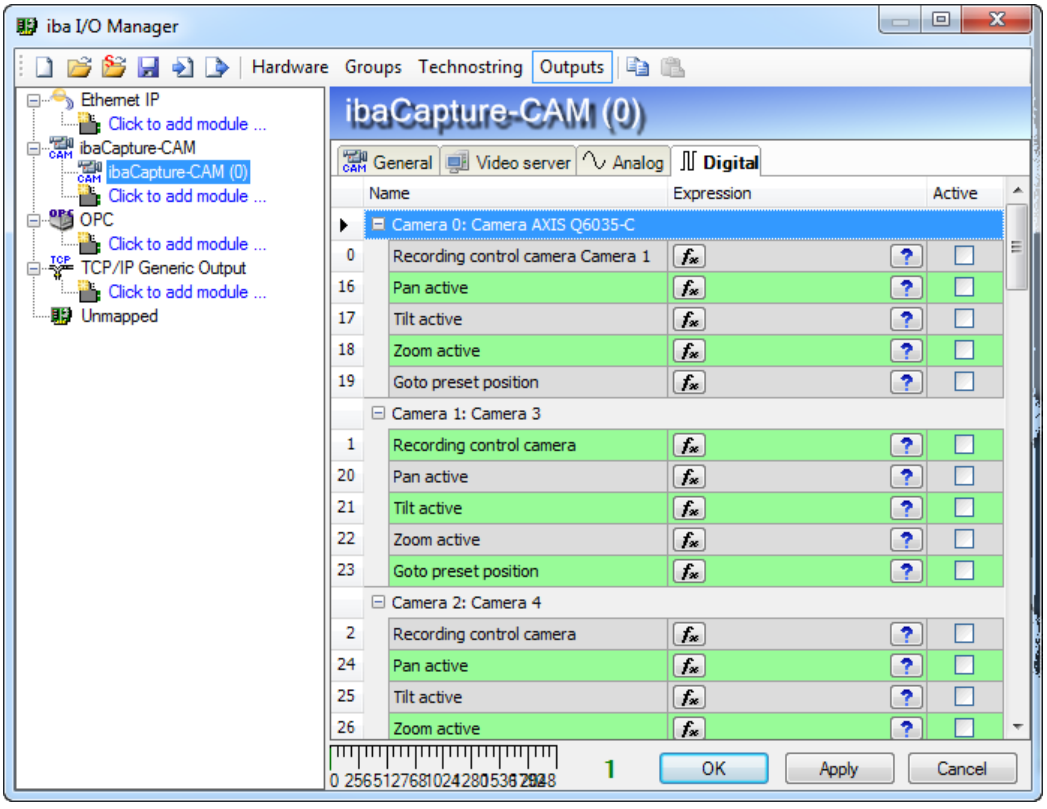
A new technostring is received if the read text is different than the previously read text.

Technostrings can now also be copied via the context menu. This works on any type of technostring.

10 PTZ support

IbaCapture-CAM v3.6.0 supports PTZ (pan-tilt-zoom) cameras. In ibaPDA there are digital and analog outputs signals that allow you to control the PTZ functionality of cameras connected to an ibaCapture-CAM server.





The output signals are grouped by camera. Per camera there are 4 analog outputs and 5 digital outputs. The first digital output is for the recording control of the camera.This signal has been available since ibaPDA version 6.29.1.

The next 4 outputs control pan, tilt, zoom and goto preset.There is each time an analog signal that gives the value and a digital signal that determines if the value is active. A command will be sent to the camera on the rising edge of the digital active signal. A command will also be sent when the digital active signal is high and the analog value signal changes. There are 4 possible commands:

- Pan: move horizontally
- Tilt: move vertically
- Zoom: zoom in or out
- Goto preset position: goto a previously saved position. The saved positions can be configured in the ibaCapture-CAM manager on the ibaCapture-CAM server.

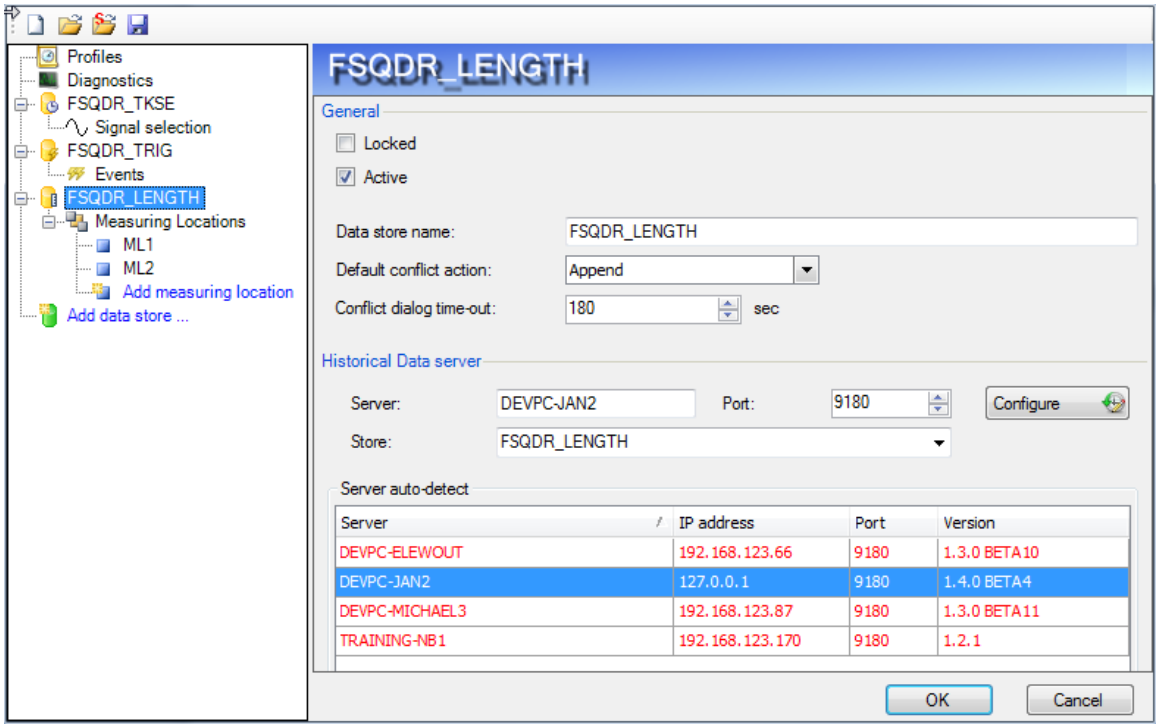
These different commands can be combined.

A command can only be generated if both the analog and digital signal are active. This is checked during the validation of the module. During the validation ibaPDA also checks if the ibaCapture-CAM server supports PTZ and if the camera supports PTZ and if ibaPDA is allowed to send PTZ commands. In the ibaCapture-CAM manager you have to explicitely allow ibaPDA to send PTZ commands.

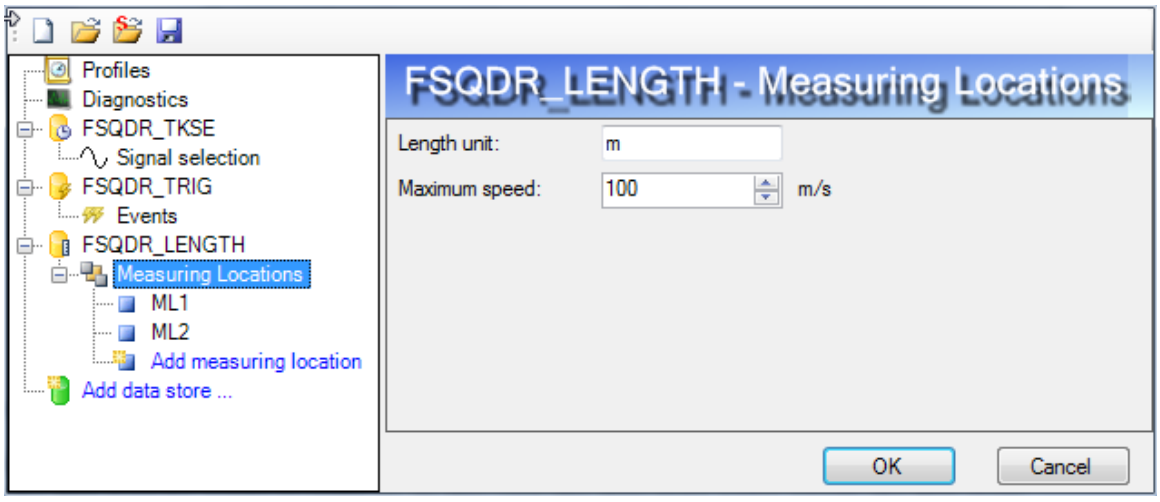
In the camera view on the ibaPDA client you can also interactively control PTZ via the mouse or a connected joystick. This functionality works with ibaCapture-CAM player v3.4.0 or higher (so no need to have player v3.6.0 installed). For more information about this functionality check out the new features document of ibaCapture-CAM 3.6.0.

11 Length-based HD data store

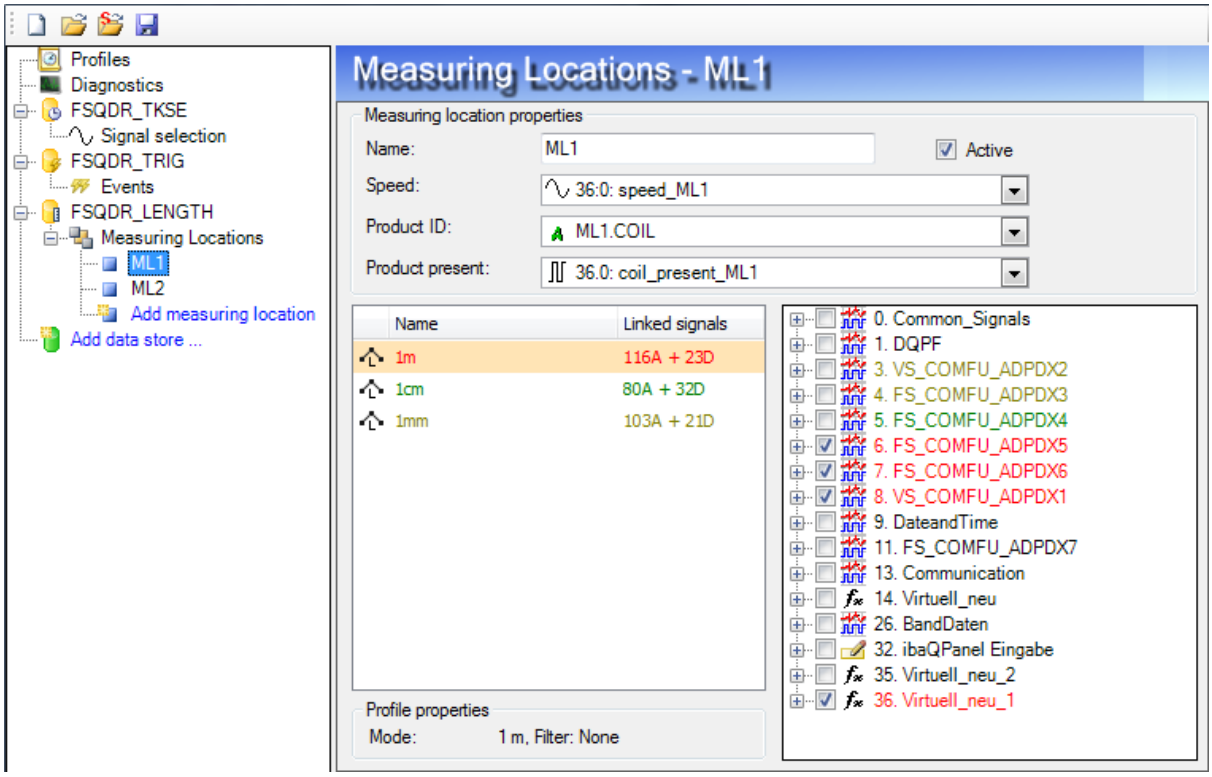
A new data store type has been added to the data storage manager: the HD length-based data store. It can be used to write data to a HD length store on a HD server.



As for other HD data store types, the user needs to specify the server name, server port and store name of the HD store he wants to write to. In this case the data store name equals the HD store name.



Multiple measuring locations can be configured for a HD length store. All locations must use the same length unit and maximal recording speed. The actual speed that is used to calculate the length data in a measuring location equals MAX(0,MIN(maximum_speed,speed_signals_value)).



In the measuring location configuration one can specify the signals that should be recorded according to a certain length profile. During one acquisition a signal can only be recorded in one measuring location under one length profile.

A measuring location requires the following information to be able to calculate length data:

- unique name for the location

  Signal references used in PDA client will incorporate the measuring location name into their signal ID. Renaming a location might thus make PDA client signal references invalid.

- product ID technostring

  The technostring value determines the name of the product that is currently processed at the measuring location.

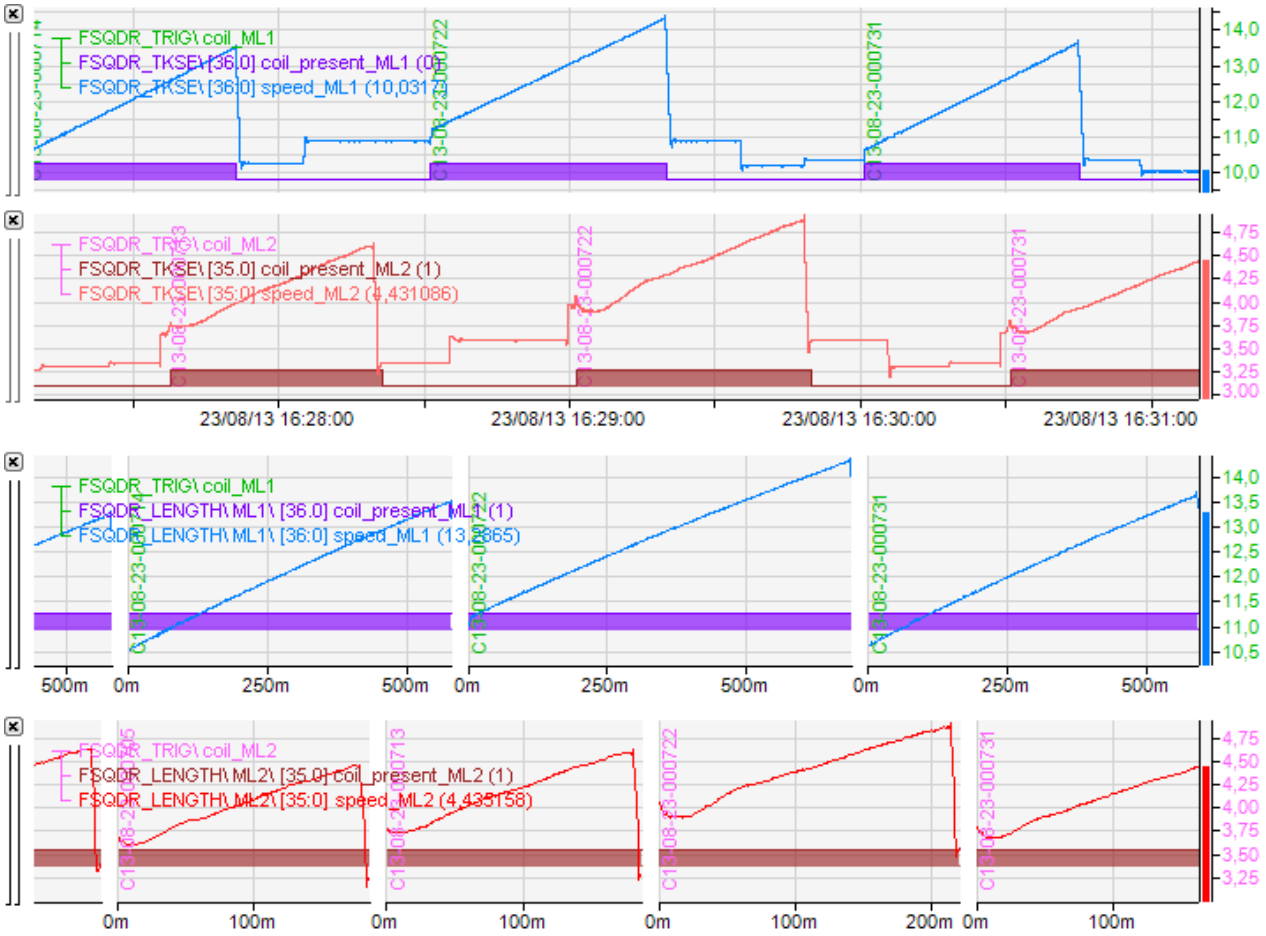
- analog speed signal

  The speed that is measured at the measuring location.

- digital product present signal

  This signal determines when a product is present in the measuring location. On the rising edge, the measuring location will start a new product. The product name will equal the value of the product ID technostring at the moment the edge was raised. From this moment on the speed signal is integrated to calculate the length data.

The following picture displays the relation between the signals mentioned above:



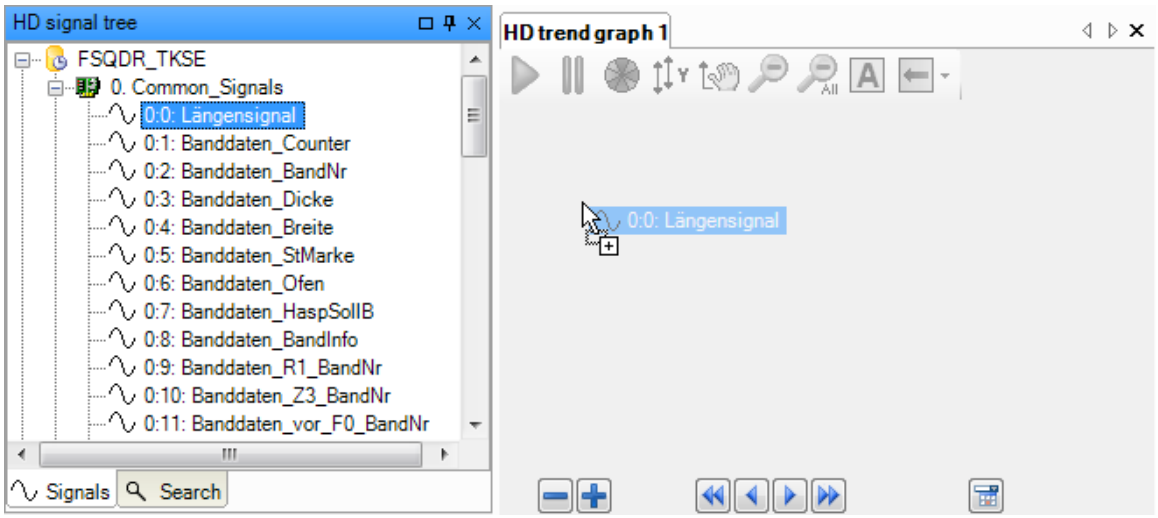
- the first graph displays the configuration signals for measuring location ‘ML1’ on a time-based graph
- the second graph displays the configuration signals for measuring location ‘ML2’ on a time-based graph
- the third graph displays the configuration signals for measuring location ‘ML1’ on a length-based graph
- the graph displays the configuration signals for measuring location ‘ML2’ on a length-based graph

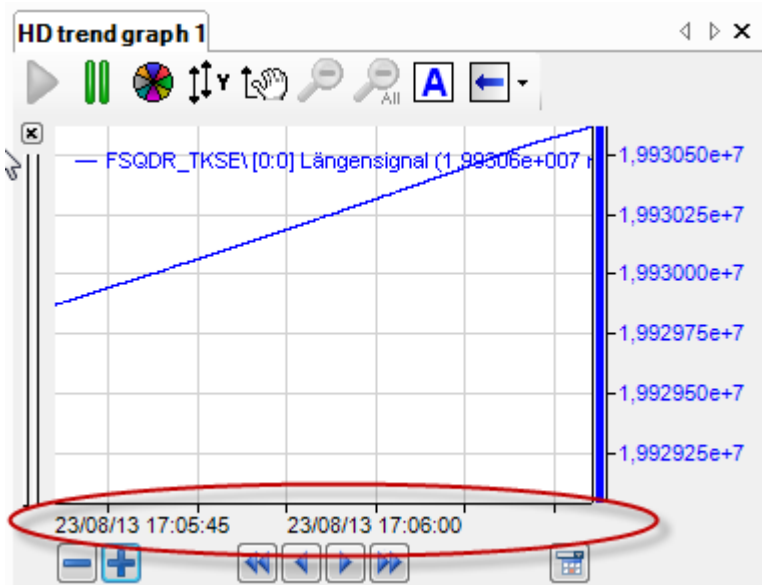
Note that the HD length data store did not record data when the ‘product present’ signal value was low. Instead the length graphs display gaps with fixed width. Of course the ‘product present’ signal value is always high on the length graphs.

As you can see, the events that are displayed on both the time and length graphs originate from one and the same HD event store. This is possible because the HD length store doesn’t only record data in length, it also records the time relation at sampling rate precision.

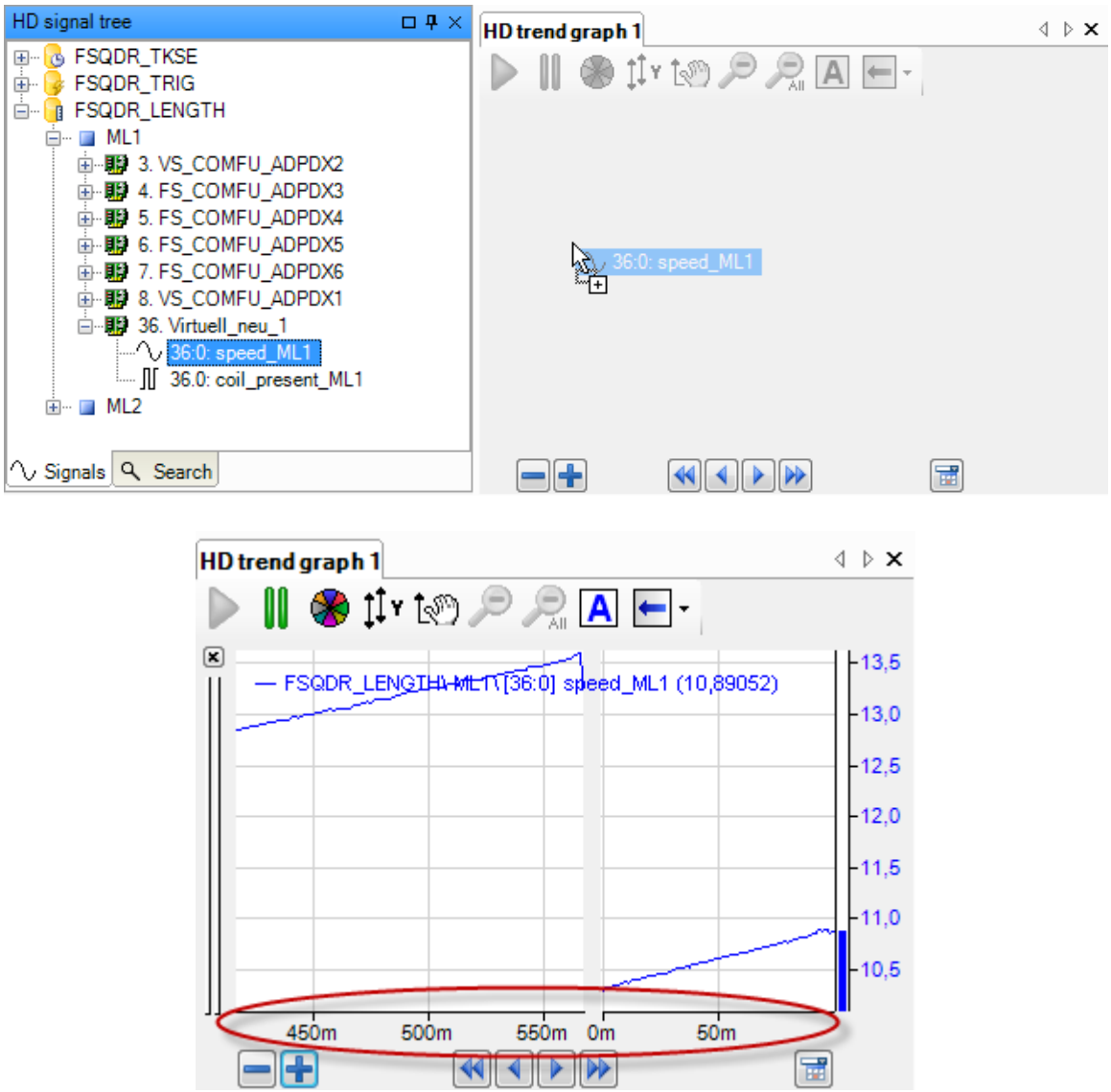
12 HD length graph

The HD trend graph in ibaPDA client has been expanded to display length based signals. When you create an HD trend graph and drop a signal from a HD time store on it, a time axis will be created. If you drop a signal from a HD length store, a length axis will be created.





Drop a HD time signal on HD trend graph to create a time-based HD trend graph

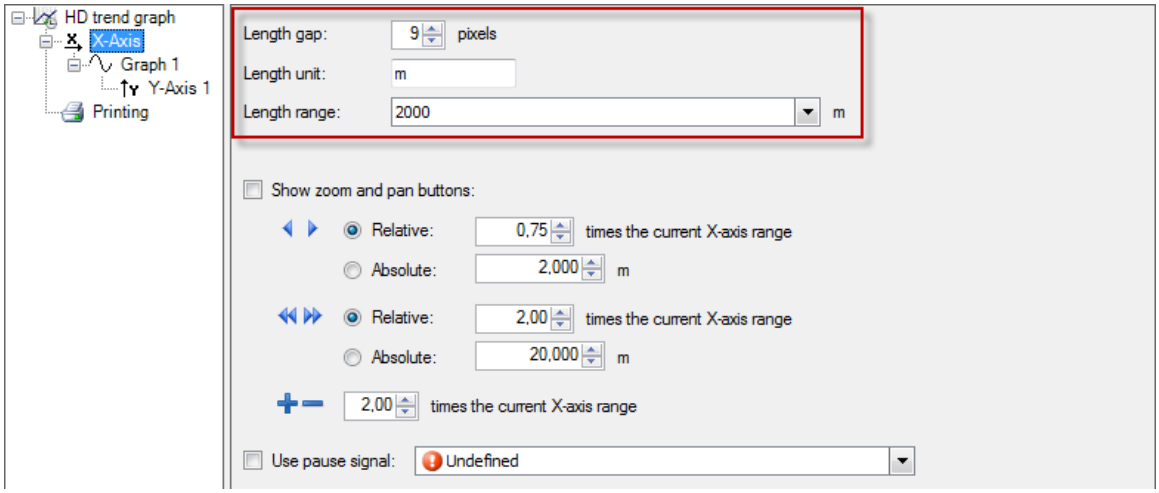


Drop a HD length signal on HD trend graph to create a length-based HD trend graph

Once a time signal has been dropped on a HD trend graph, you won't be able to add length signals. Once you've dropped a length signal originating from a certain measuring location, you'll only be able to add other signals originating from that measuring location.

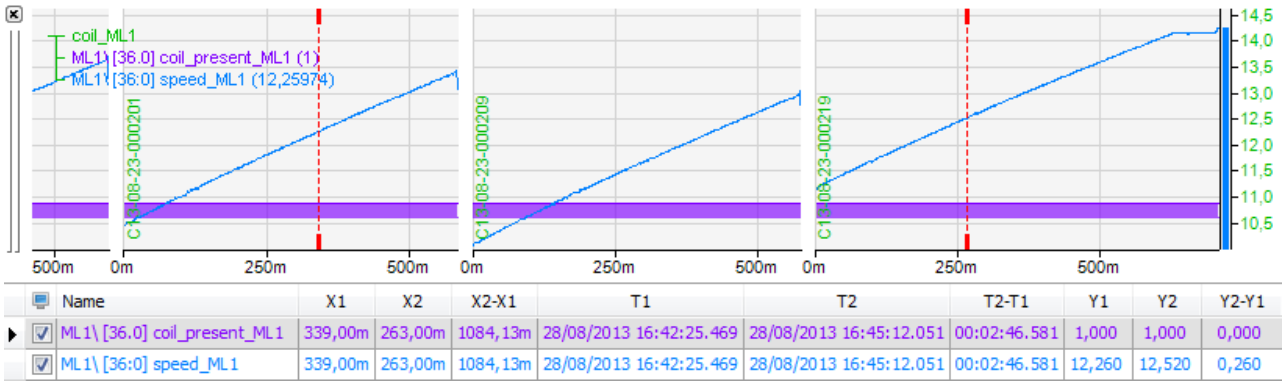
On a QPanel you can add HD event store signals on both a time-based and a length-based HD trend graph.

- A length-based HD trend graph (≈ length graph) has the following X axis properties:



- Length gap: number of pixels used to display length gaps
- Length unit: unit displayed on X axis ticks
- Length range: the length range to be displayed on the X axis. This range is not always used as length range though. Depending on the length base of the signals that are displayed on the graph, the actual x axis range might be smaller than the range that is specified. This because a length-based HD trend graph will maximally display 50000 samples for each signal on the graph. If the length base of a signal equals 1mm the maximal x-axis range will thus be 50m.

- The length graph shows 3 extra marker grid columns: T1, T2 and T2-T1.



Name	X1	X2	X2-X1	T1	T2	T2-T1	Y1	Y2	Y2-Y1
ML1\ [36.0] coil_present_ML1	339,00m	263,00m	1084,13m	28/08/2013 16:42:25.469	28/08/2013 16:45:12.051	00:02:46.581	1,000	1,000	0,000
ML1\ [36:0] speed_ML1	339,00m	263,00m	1084,13m	28/08/2013 16:42:25.469	28/08/2013 16:45:12.051	00:02:46.581	12,260	12,520	0,260

When a length graph is paused:

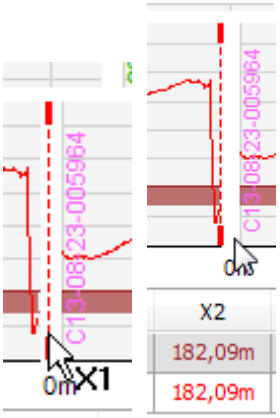
column X1 displays the product length at marker 1

- column X2 displays the product length at marker 2
- column X2-X1 displays the accumulated length between marker 1 and marker 2
- column T1 displays the time stamp at marker 1
- column T2 displays the time stamp at marker 2
- column T2-T2 displays the time span between T2 and T1 (including gap time spans)

When a length graph is in live-mode:

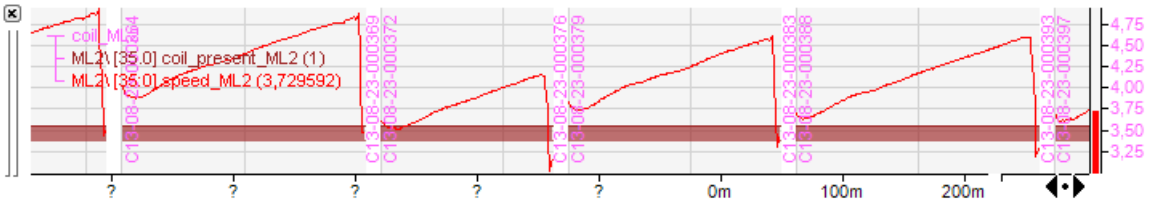
- column X1 displays the live product length
- column T1 displays the live time stamp

- column T2-T2 displays the total axis time span (including gap time spans)



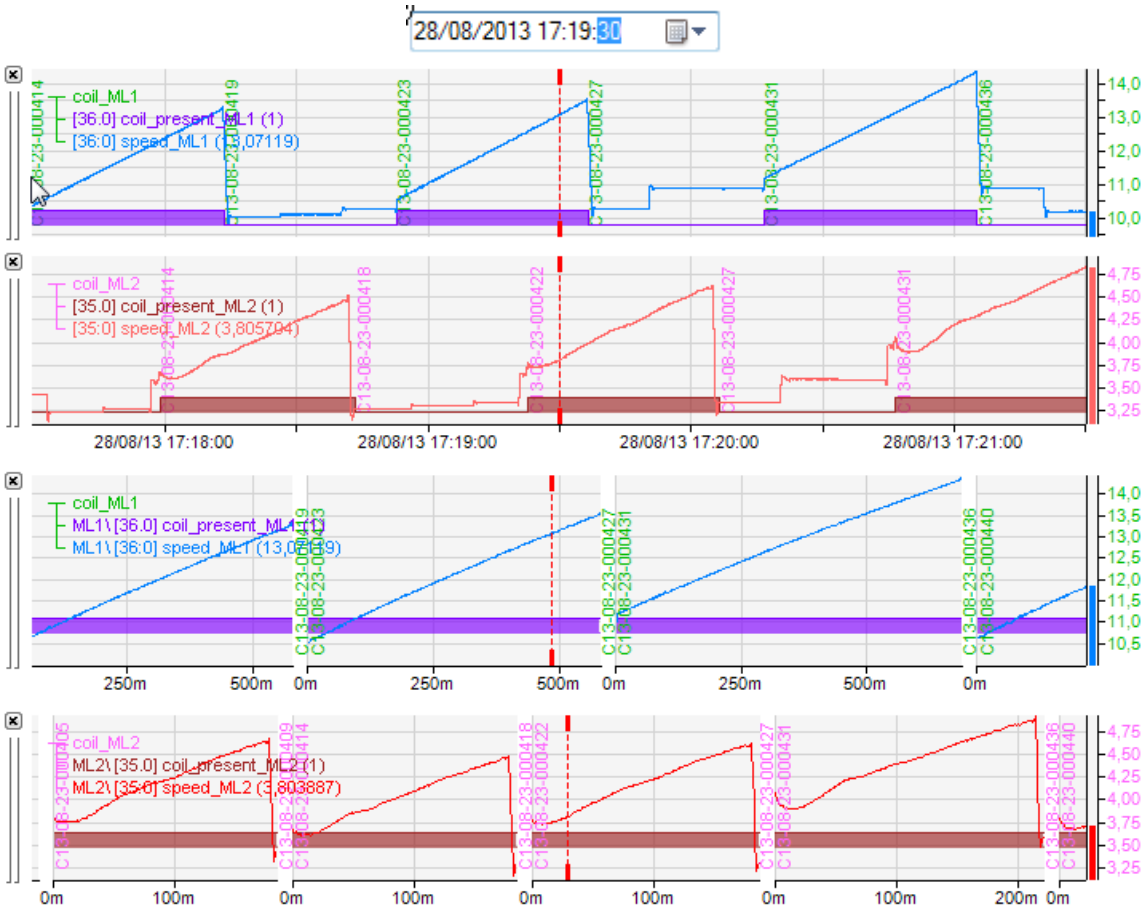
When a marker is dragged over a length gap and it is released the marker will snap to the end of the product before it. This way the total length of a product can easily be inspected.

- When scrolling the X axis on a length graph, certain ticks on the axis might display a question mark:



This is because the tick's values depend on the signal length data. When the user releases the mouse button to stop the scroll operation, the graph will request the data from the HD server and the axis will be updated with the correct product information.

- Time jumps are possible both on the time- and length-based HD trend graph. A time jump can be executed via a time picker or via an event table.



Using the time picker both time- and length-based HD graphs can perform a time jump

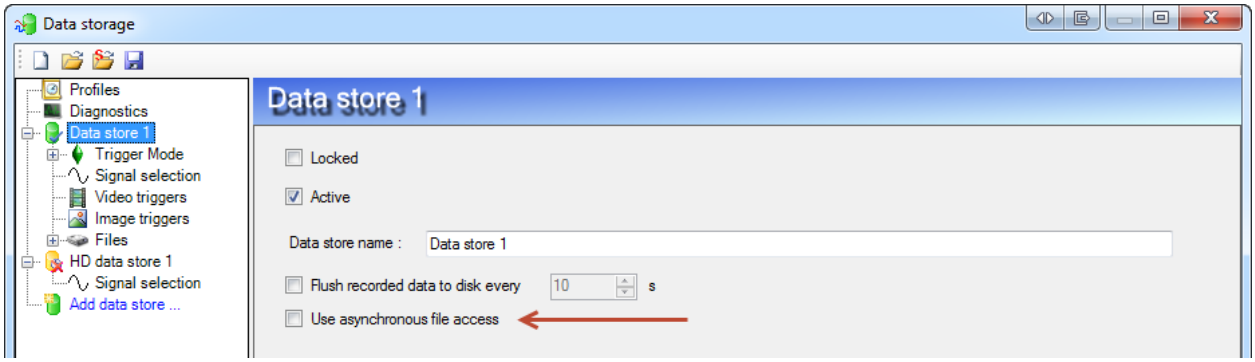
- When an HD event signal is displayed on a length graph, the event occurrences that are recorded when the measuring location's 'product present' signal value is low are all displayed inside the length gap.

13 Datastore changes

The datastore file writing consists of two parts:

- The monitoring of triggers and run-length encoding of data
- The actual writing of run-length encoded data to disk

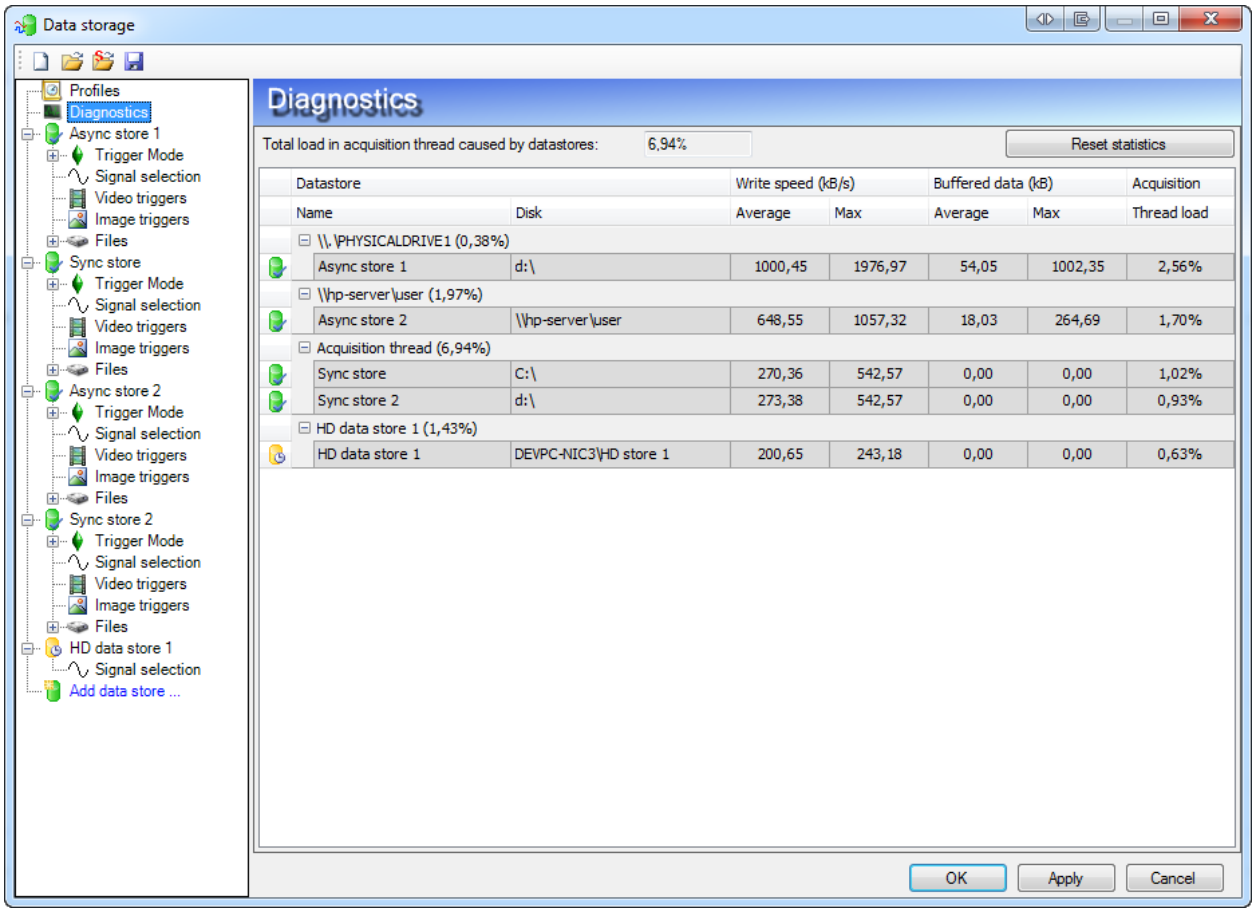
In ibaPDA versions prior to version 6.31.0 both these parts were executed on a single thread called the acquisition thread. The different datastores were handled one after the other. This same thread is also responsible to trigger data updates on the clients and to generate alarms. If one datastore took a long time then this affected the other datastores and also the data update on the connected clients. If e.g. a network drive was no longer available then the clients would stop updating for several seconds. Also the generation of alarms would stop for a certain amount of time. In extreme cases the measurement could be stopped as well if the buffer in the driver would overflow.



In version 6.31.0 the actual writing of data to disk can be performed on a different thread than the acquisition thread by enabling the asynchronous file access option. There will be a disk writing thread per physical hard drive. If e.g. the D: and E: partition belong to the same physical hard drive then all datastores that write to D: and E: will do their writing on the same writing thread. If the D: and E: partitions belong to different physical disks then there will be 2 threads writing data. One for the D: hard drive and one for the E: hard drive. This has some advantages:

- Different data stores can write in parallel to their own disk
- Network failures no longer affect other data stores, client updates and alarm generation

The disadvantage of asynchronous file access is that more memory is used. If a disk buffer reaches 500MB then the acquisition will be stopped and automatically restarted after 5s.



There is also a new diagnostics node in the data storage manager. The table shows the performance of all data stores. There is one row per datastore and the rows are grouped by the thread that is writing data. The name of the thread and the percentage of load of the thread are shown in each group row. The displayed thread loads are by default average loads. Via the context menu you can also switch between average and actual values.

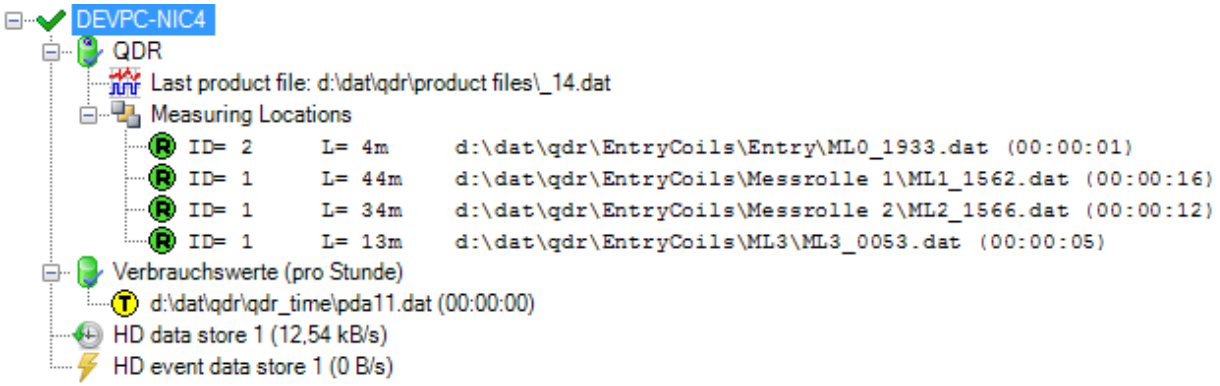
Normal **timebased datastores with synchronous file access** rows show the write speed of data written to disk. The buffered data columns will show zeros because the data is immediately written to disk. The acquisition thread load column shows the time spent generating the run-length encoded data and writing it to disk.

Timebased datastores with asynchronous file access rows show the write speed of data written to disk. The buffered data columns show the amount of buffered data that is waiting to be written to disk. The acquisition thread load column shows the time spent generating the run-length encoded data. The actual writing is done on the write thread and the load you can read from the group row.

The **QDR datastore** row shows the write speed of all entrycoil files at all measuring locations. The write speed of the product files and entry product files is not shown. The QDR product thread group row shows the load of the thread generating the product files. The buffered data contains all the length-based data of the entrycoils that haven't been used in a product file yet. The acquisition thread load shows the load generated by saving timebased data to entrycoil files and by generating length based data on all measuring locations.

The **HD datastores** (timebased, lengthbased and event) rows show the write speed of data written to the ibaHD server. The buffered data shows the amount of data that is waiting to be written to the ibaHD server. Normally the values should be close to zero. When the connection to the ibaHD server is lost then the amount of buffered data will rise. The acquisition thread load column shows the time spent generating the data that needs to be written to the ibaHD server. This time includes runlength encoding for timebased stores, event triggers evaluation for event stores and lengthbased data generation for lengthbased stores.

At the top of the diagnostics tab there is the load in the acquisition thread caused by the datastores. This value is the sum of all the values in the Acquisition Thread Load column. It shows how much time is spent processing data that needs to be stored. If this goes to 100% then alarm generation and client updates will get paused. If it constantly stays at 100% then ibaPDA is overloaded and the acquisition will be stopped when the buffer in the driver overflows.



The datastorage status tree has been updated to show the icons corresponding with the datastore type.

14 Chart view

14.1 Introduction

The *Chart View* is a new view in the ibaQPanel. It contains new features that make monitoring much more comfortable.

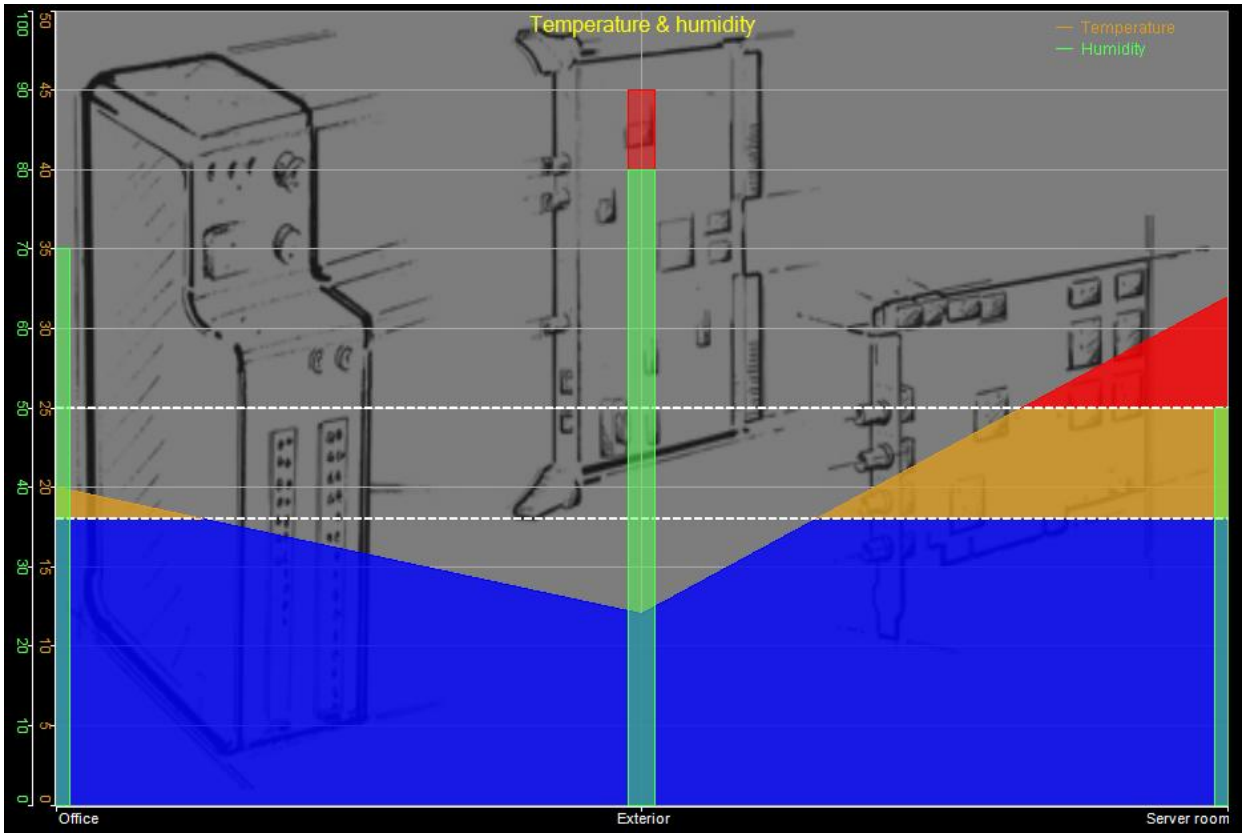
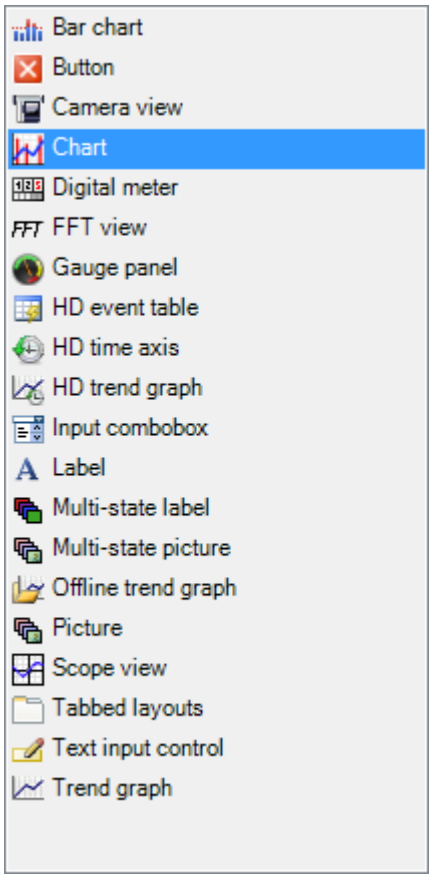


Figure 1 - The Chart is a versatile and powerful new view to display signal data

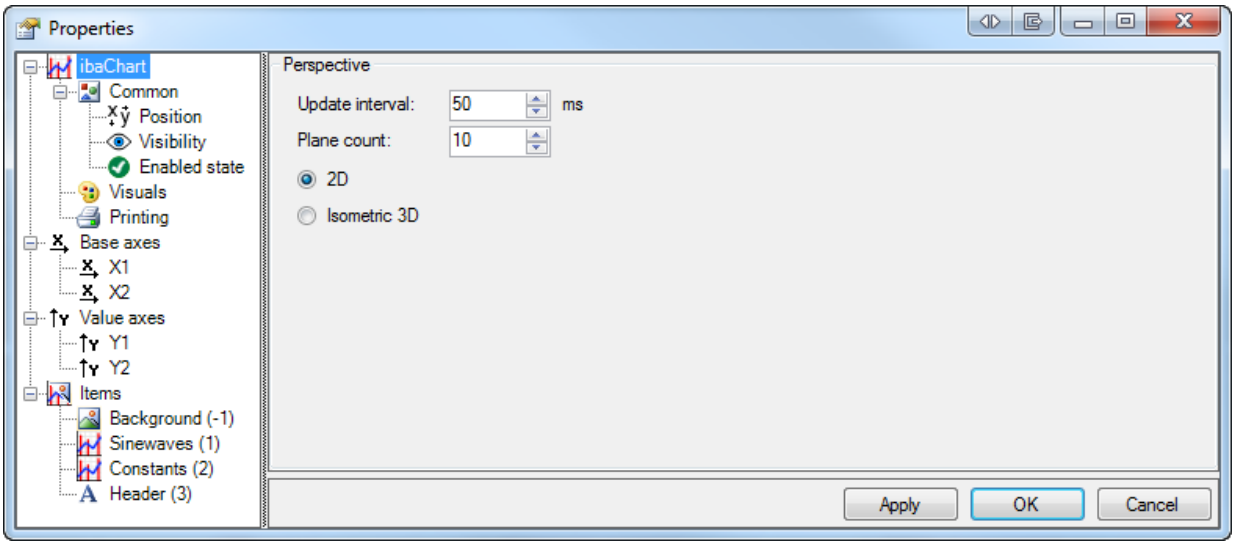
Apart from curves in a variety of representations, a Chart can for example also display images or texts. In Figure 1 we see an example of a Chart representing temperature and humidity at three locations. Threshold values have been set for both curves to make it visually clear when we are outside of normal operation regime. In the background an image file is displayed and finally, a text header gives information on the contents of the Chart. The curves, the image and the text are called *Items* and represent the actual data of the Chart. We will start by explaining how to configure the base and value axes after which we will explain in detail all Items that are available in the Chart.

14.2 General

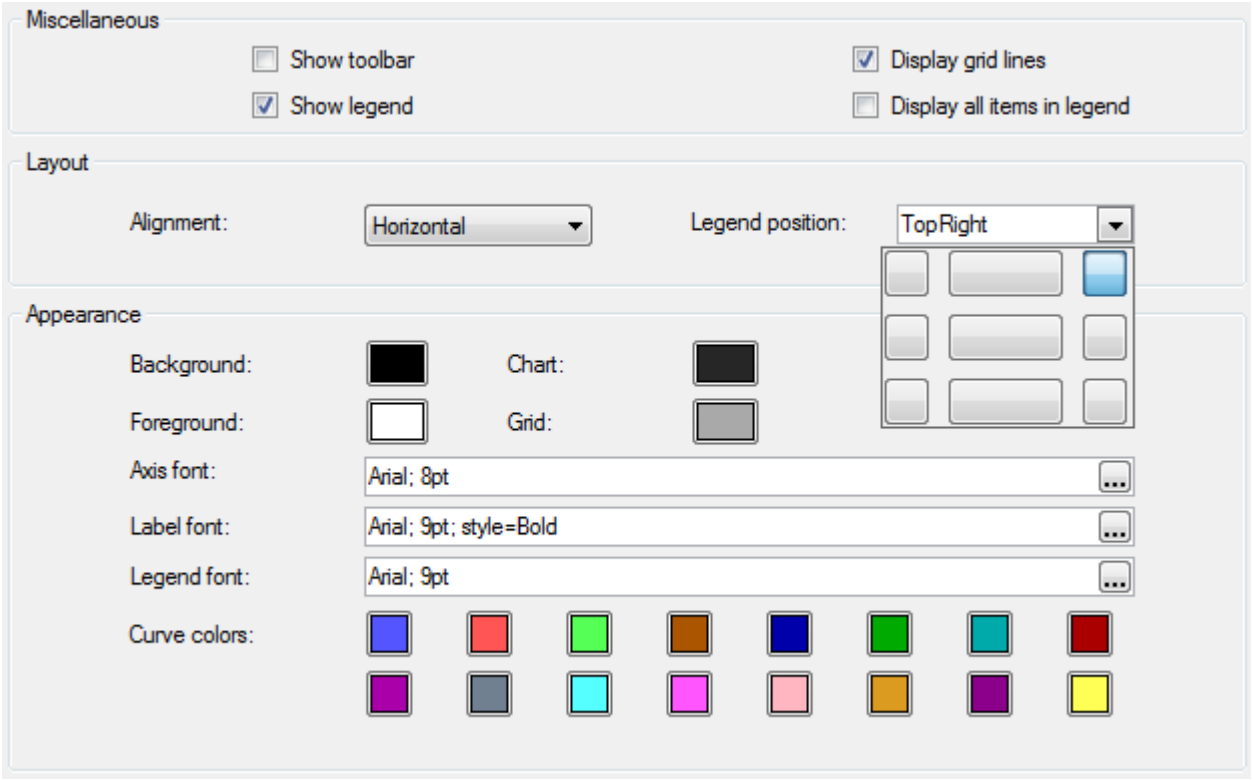
The Chart can be found in the QPanel toolbox so it can be placed in a QPanel by either double-clicking it or dragging it to a panel.



Editing the properties of a Chart produces the following dialog:



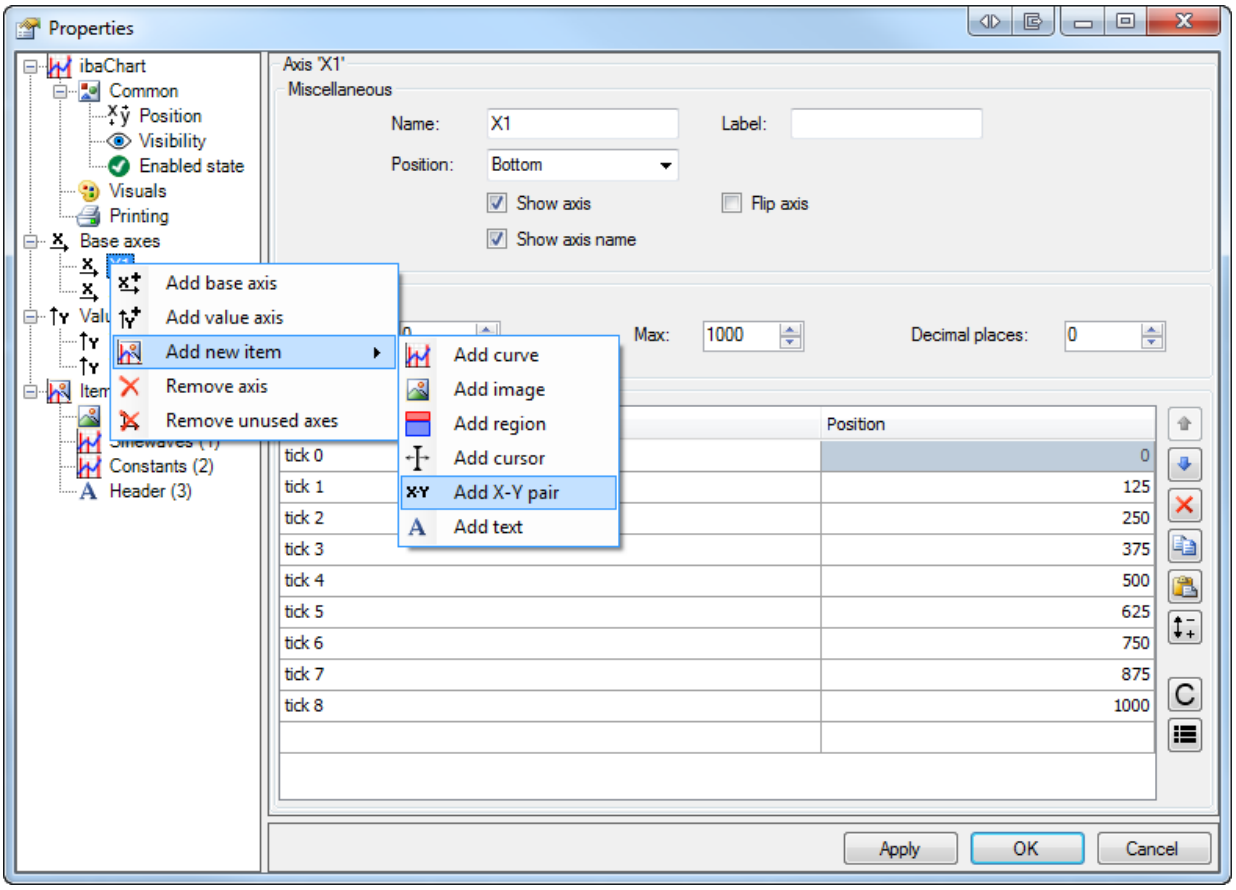
The main node of the Chart only contains an option to enable isometric 3D view and will be discussed in section 14.6. Apart from the general Chart node, there are three other main nodes: base axes, value axes and items. Details of these three nodes will be covered in the following sections.



In the *Visuals* node, general colors for the Chart as well as the different curve colors can be configured. Fonts used for labels and the legend are also defined here.

The *Alignment* parameter determines whether curves advance in the horizontal or vertical direction.

By default, only curves and X-Y pairs are displayed in the legend. This behavior can be changed by enabling *Display all items in legend*.



When right-clicking a node, a context menu appears which allows the user to either remove the selected axis or item or to add a base axis, a value axis or an item. Depending on whether an axis node is selected, it is possible to remove unused axes i.e., axes that are not linked to any item.

14.3 Axes

14.3.1 Base axes

Several base- or X-axes can be configured per Chart.

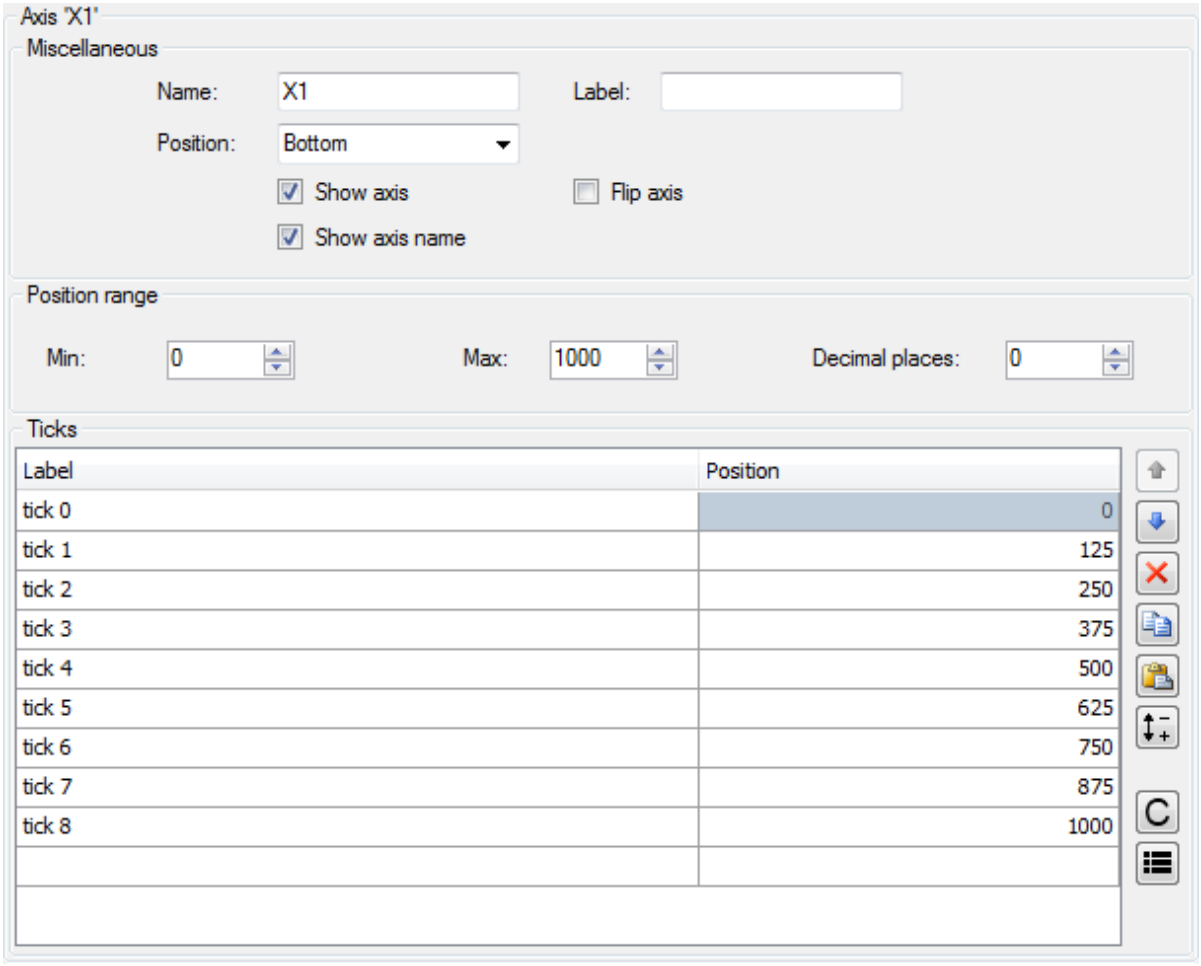


Figure 2 - The properties dialog of a base axis

In the configuration dialog, as shown in Figure 2, some general and layout settings can be configured for the axis as well as the actual ticks.

Using the *Position range* section, the user can define the minimum and maximum value of the current base axis. Note that the minimum value cannot be lower than -1000000 and the maximum value cannot be higher than 1000000. The decimal precision can also be set here (with a maximum precision of 10).

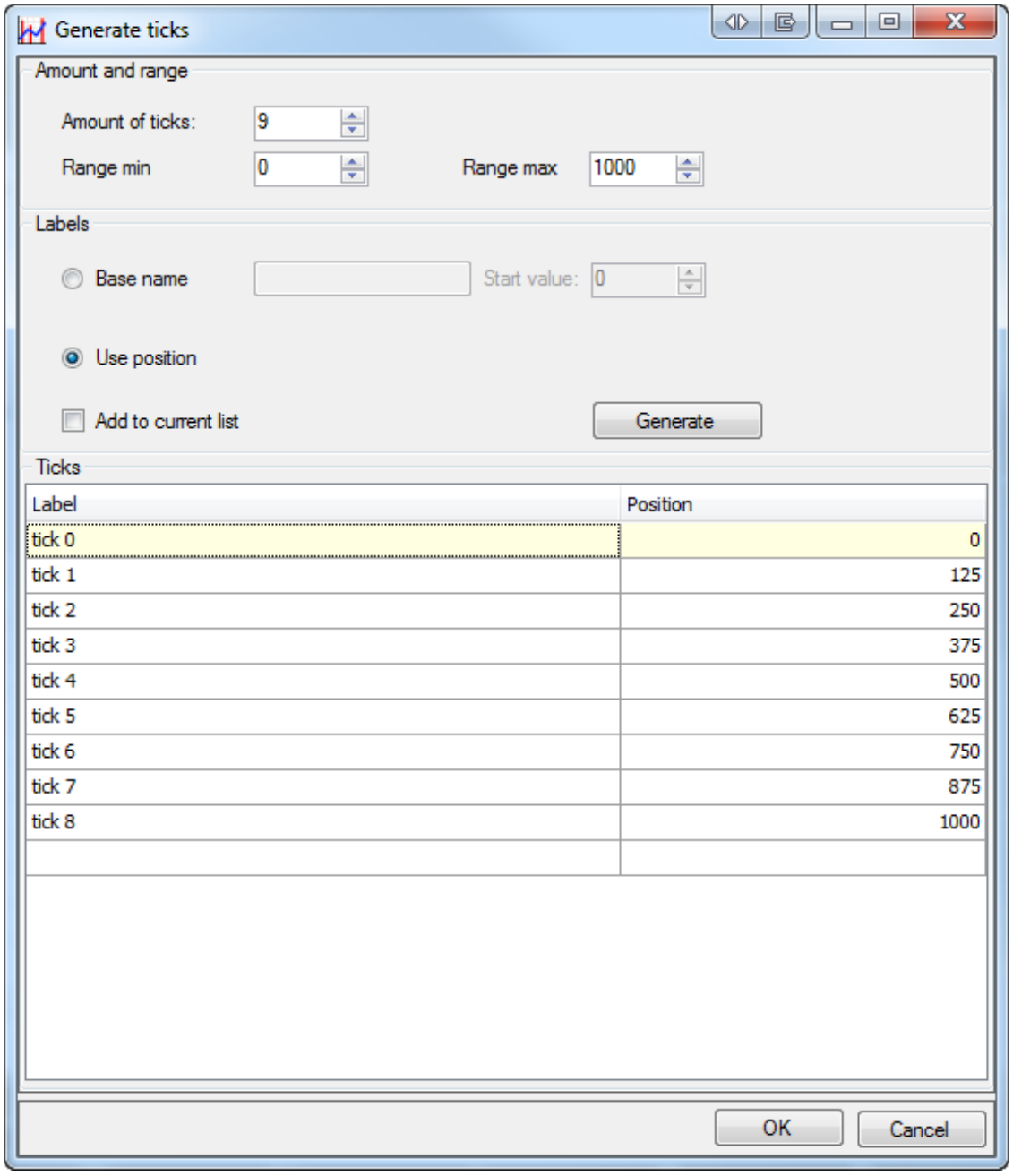
Once the range is set, the actual ticks can be configured. This can be done by clicking on an empty row in the list in the *Ticks* section. The position represents a point in the defined range where a tick should be placed. Each tick also has its own text label which can also be configured here.

The buttons on the right of the grid can be used to efficiently manipulate the tick list:

-  and  are merely used to change which tick entry is focused in the grid
-  is used to delete a tick entry

- allows the user to copy the tick list to the computer's clipboard so it can be pasted in e.g. Microsoft Excel
- does the reverse and pastes a tab-separated grid list in the base axis tick list
- sorts the ticks in ascending order
- is used to quickly clear the entire tick list
- opens the tick list generator

The tick list generator is a useful tool to rapidly obtain an entire list of equally spaced ticks. When clicking the tick list generator button, the following dialog appears:



Range min and range max correspond to the range of the base axis for which the tick list generator was opened. By setting the number of required ticks and clicking Generate, the tick list will be automatically generated.

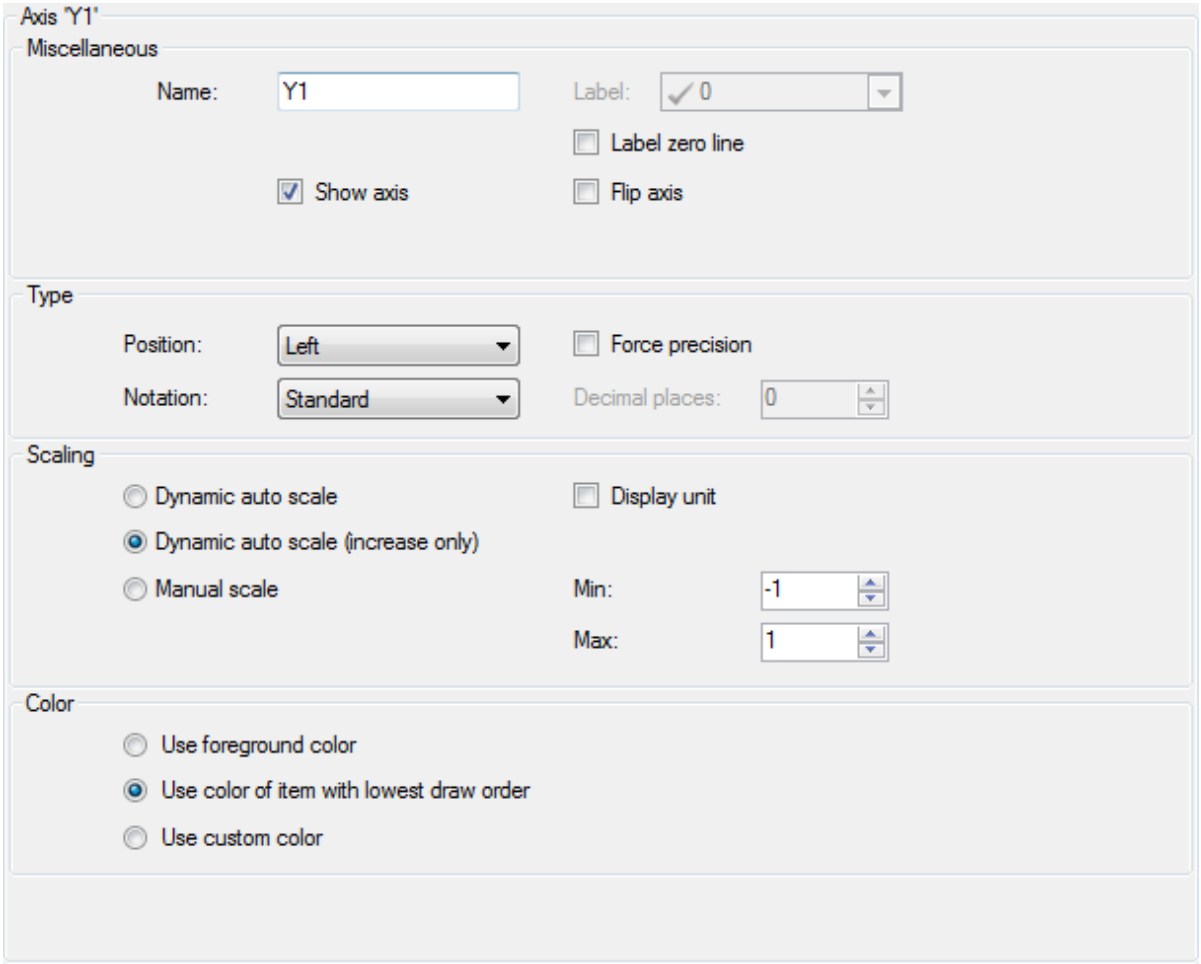
By default, the labels corresponding to the ticks will contain the tick position. By selecting *Base name*, a more dynamic name can be obtained. Of course, it is stil possible to manually edit the labels in case a non-logical naming scheme is desired.

Checking the *Add to current list* box will result in the generated list to be added to the already present ticks. Otherwise, the previous ticks will be discarded.

14.3.2 Value axis

Similar to base axes, it is also possible to link several items to the same value axis. The properties of a value axis can be configured using the dialog displayed below:

The user is able to change an item's value axis by using the legend. A value axis has multiple scaling modes, like dynamic auto scale, dynamic auto scale (increase only) and a manual scale, which is adjustable within the properties dialog.



In contrast to the base axis a value axis can be zoomed and panned. The user can pan a value axis by dragging the axis. By scrolling the mouse wheel, the user can zoom in or out around the mouse cursor. One can also perform scale manipulations using the value axis' popup buttons, which will appear when hovering over the right side of a horizontal axis or the

top of a vertical axis. The outermost icons will halve or double the scale range around its center value. The arrows perform a similar operation but with a smaller zoom factor. The center button auto scales the axis.

If dynamic auto scaling (or the *increase only*-mode) is selected, axis manipulations have no effect. If a value axis is removed, by using the axis' context menu, all items that are assigned to the selected axis will be linked to the next available axis.

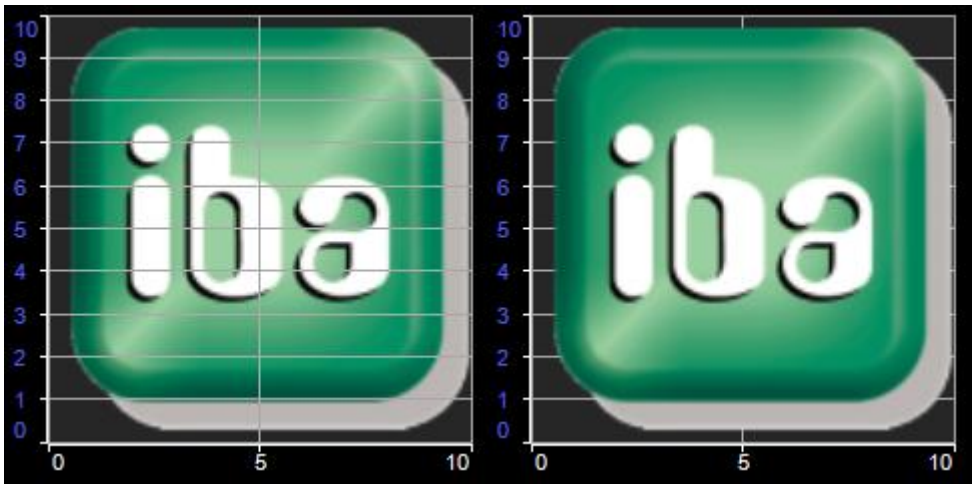
Another handy feature is that the user is able to label the 0 value of a value axis by a signal. If the measured values only measure the deviation from a reference value (e.g. ±5) the user can assign the reference value (or signal) to the base line of the value axis.

The user also has the opportunity to set the axis' color. *Use color of item with lowest draw order* means that the color of the value axis is the color associated to the item with the lowest draw order (excluding image items and text items) assigned to this axis. *Use foreground color* means that the color of the value axis will correspond to the color selected in the *Visuals* node of the properties tree.

Note that if a value axis does not contain any item, the axis will not be drawn.

14.4 Items

All items have two properties in common. A first one is, similar to axes, a name that is used to identify the item in the tree structure and the legend. The second one is the *Draw position*: this parameter determines the order in which the items will be drawn onto the Chart. A draw position smaller than zero indicates that the item is drawn underneath the Chart's grid lines. A draw position greater than (or equal to) zero indicates that the item will be drawn on top of the grid lines. The item with the highest draw position will be frontmost object. Note that for transparent items, the grid lines will always be visible.



14.4.1 Curve

The curve is the most efficient and configurable item to display signal values.

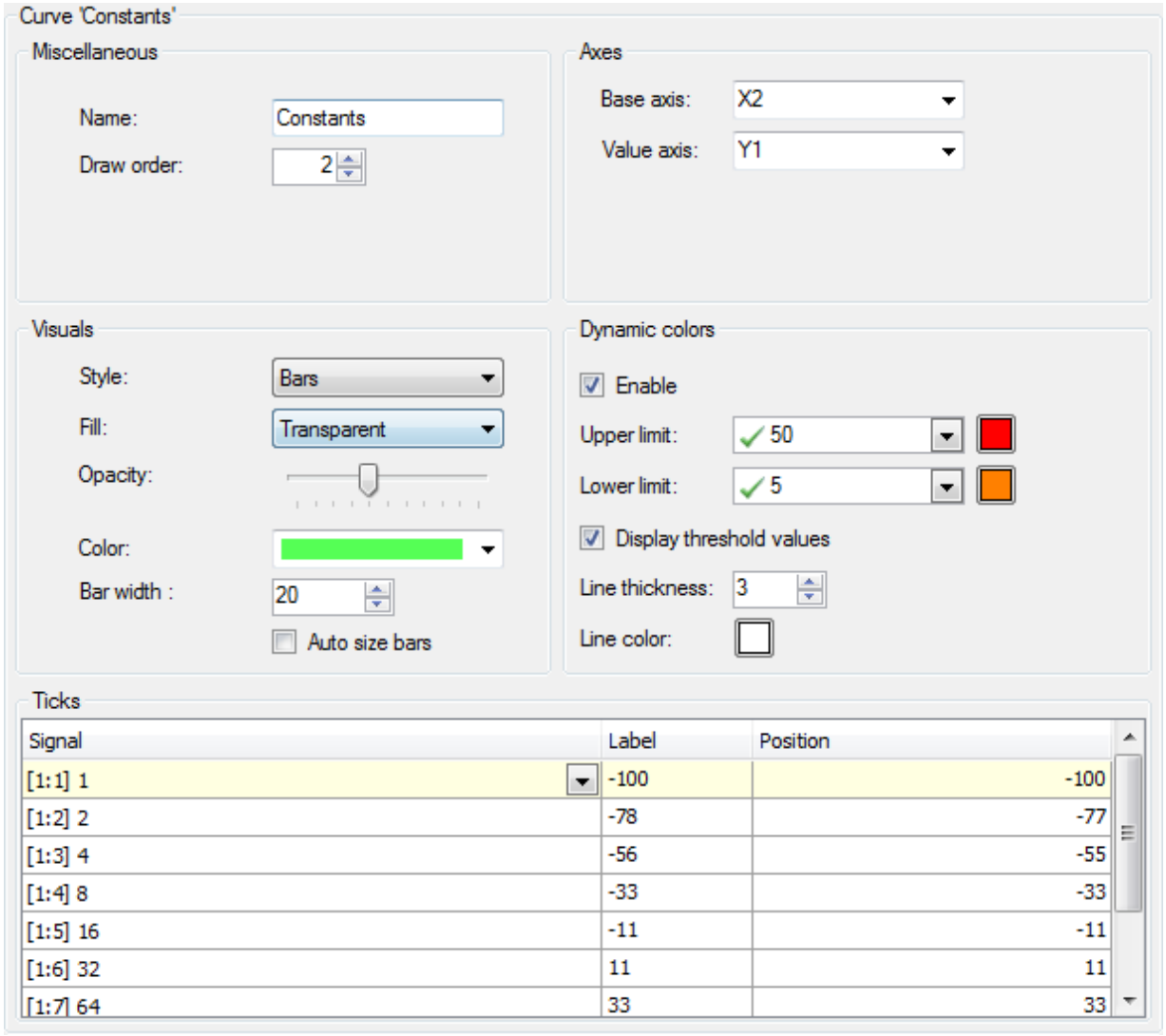


Figure 3 - The properties dialog of a curve item

Every curve must be linked to both a base axis and a value axis, which can be selected through the drop-down menus in the *Axes* section.

A curve can be displayed in four possible ways: as discrete bars, lines or points or as a continuous curve with a user-defined pen/bar width. For each display style, it is also possible to configure whether the resulting drawing should be empty, opaque or transparent. In the latter case, it is possible to define an opacity ranging from 0% to 100% where 0% corresponds to full transparency and 100% to opaqueness.

Apart from a default color, the user can also define fixed or dynamic threshold values with corresponding colors. In the example of Figure 3, the portion of the curve above 50 will be colored in red while values below 5 will be colored in orange. Values in between the thresholds will be drawn using the color set in the *Visuals* section.
The current threshold values can also be displayed as dashed lines with a user-defined color and line thickness.

The actual signals, linked to the ticks can be defined in the *Ticks* section.

The user is able to drop signals into the view. If the amount of dropped signals is less than five, the signals will be dropped on a newly created base and value axis automatically.

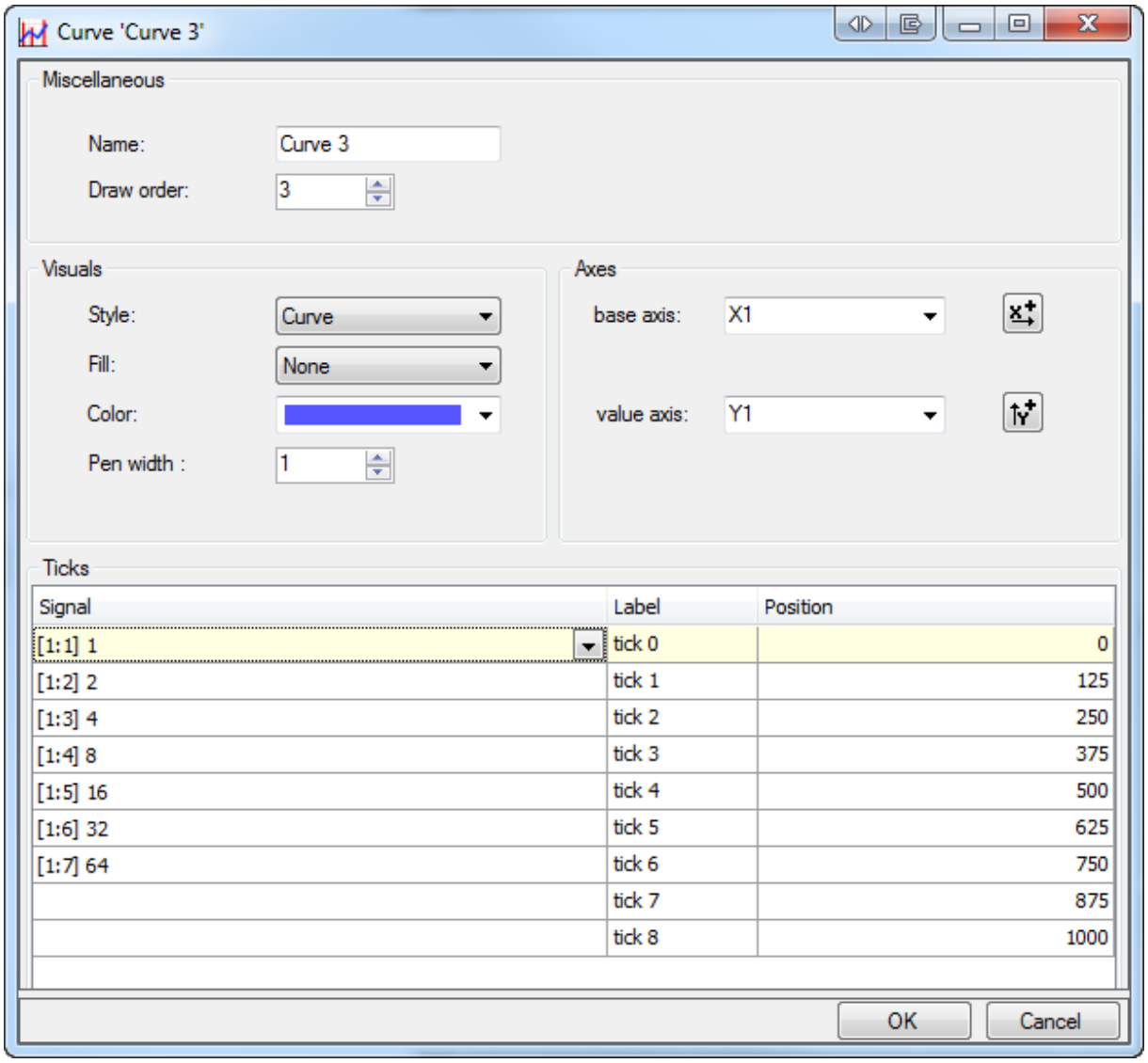
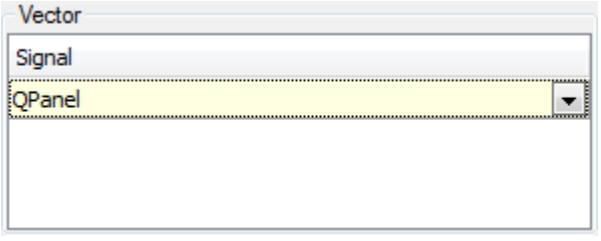


Figure 4 - Dropping more than 5 signals at a time produces this dialog to manually configure a base and value axis

When more than five signals are dropped into the view a new dialog will appear (see Figure 4) where a base and value axis can be selected. In addition, the user has the opportunity to add new base and value axes by clicking the buttons  and  .

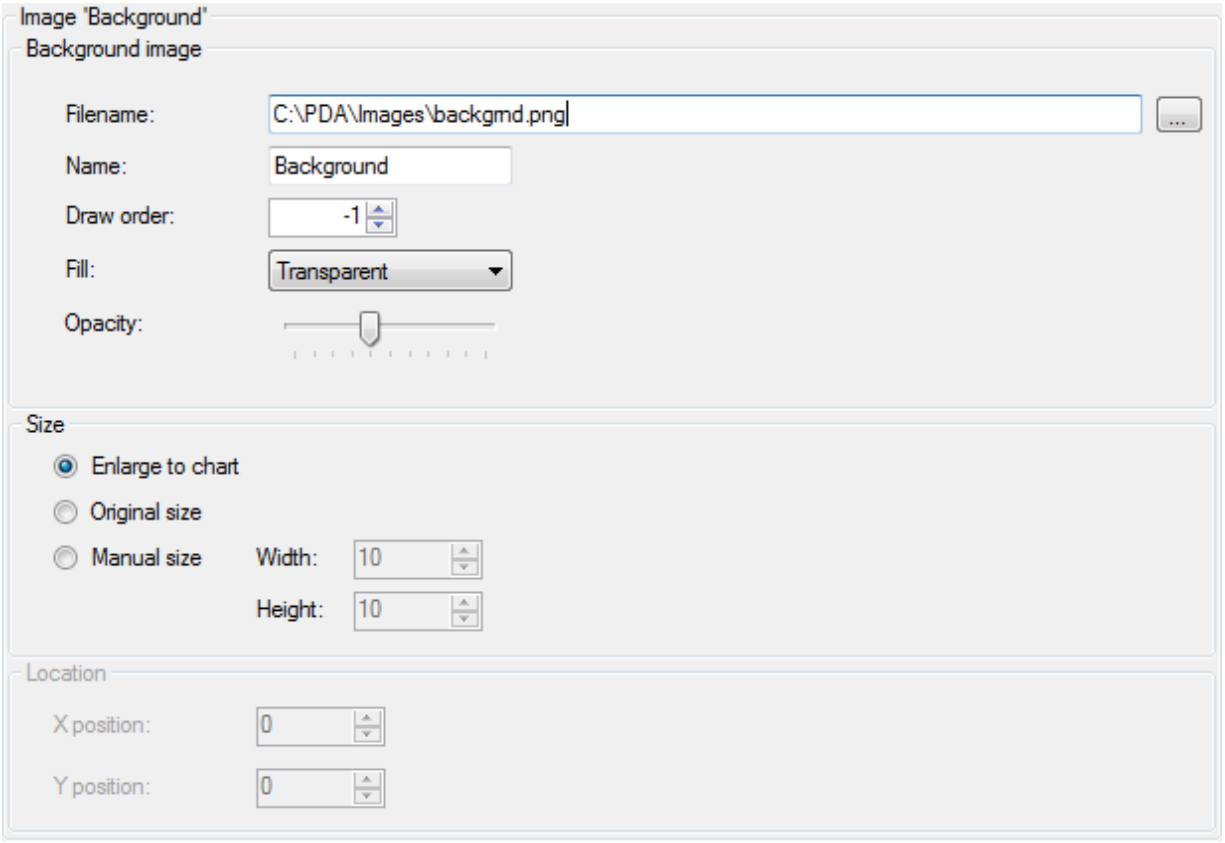
Vectors can also be dropped in the Chart.



The tick list will be different in this case: the label and position column are no longer visible and only the vector signal is displayed. It is possible to select another vector but it is not possible to switch back to the original mode where individual signals can be selected in the tick list. When a different vector is selected, it is checked whether the selected base axis contains enough ticks to properly display the new vector. If this is not the case, a warning will be displayed in the *Axes* section and changes will not be applied.

14.4.2 Image

Static images can be incorporated in the Chart through means of the image item.



Apart from the standard item name and draw position, the user has to supply the full path of the image. If the path is invalid or does not exist, a default image (the iba logo) will be used.
There is also an option to set the transparency of the image by selecting the *Transparent* fill mode and adjusting the opacity slider.

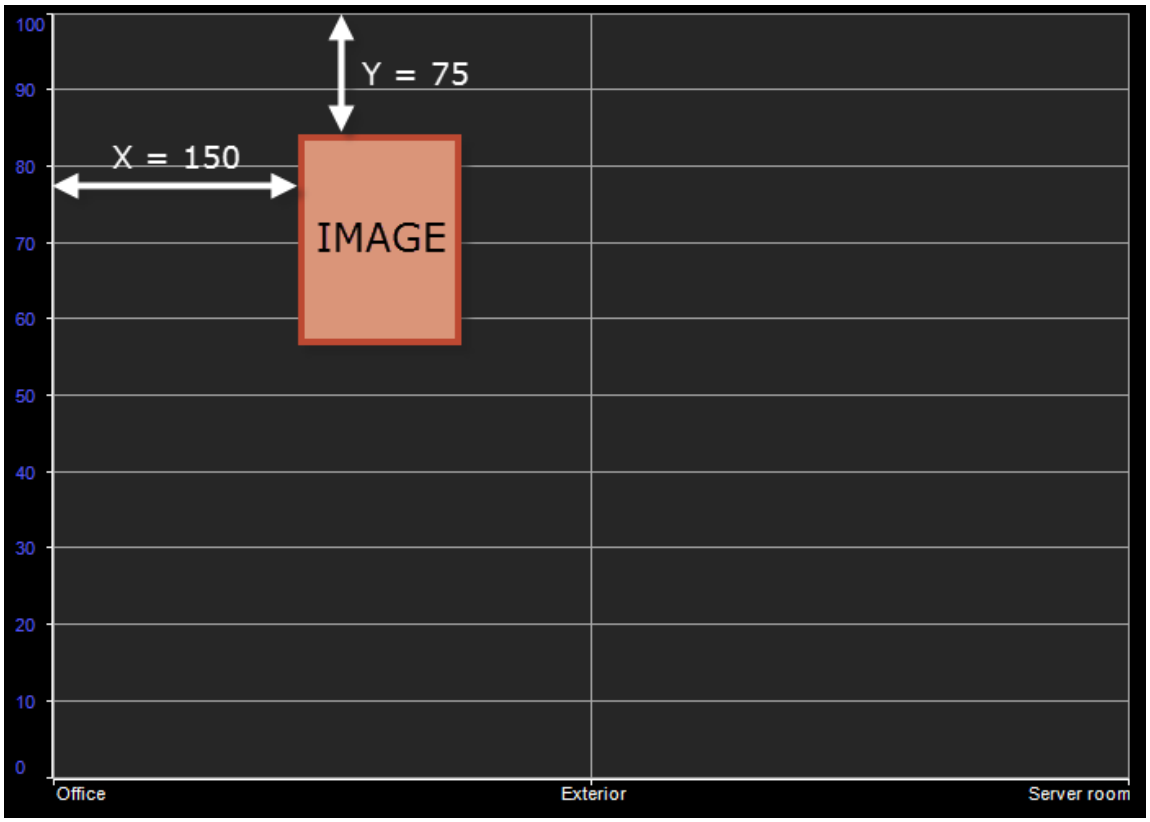


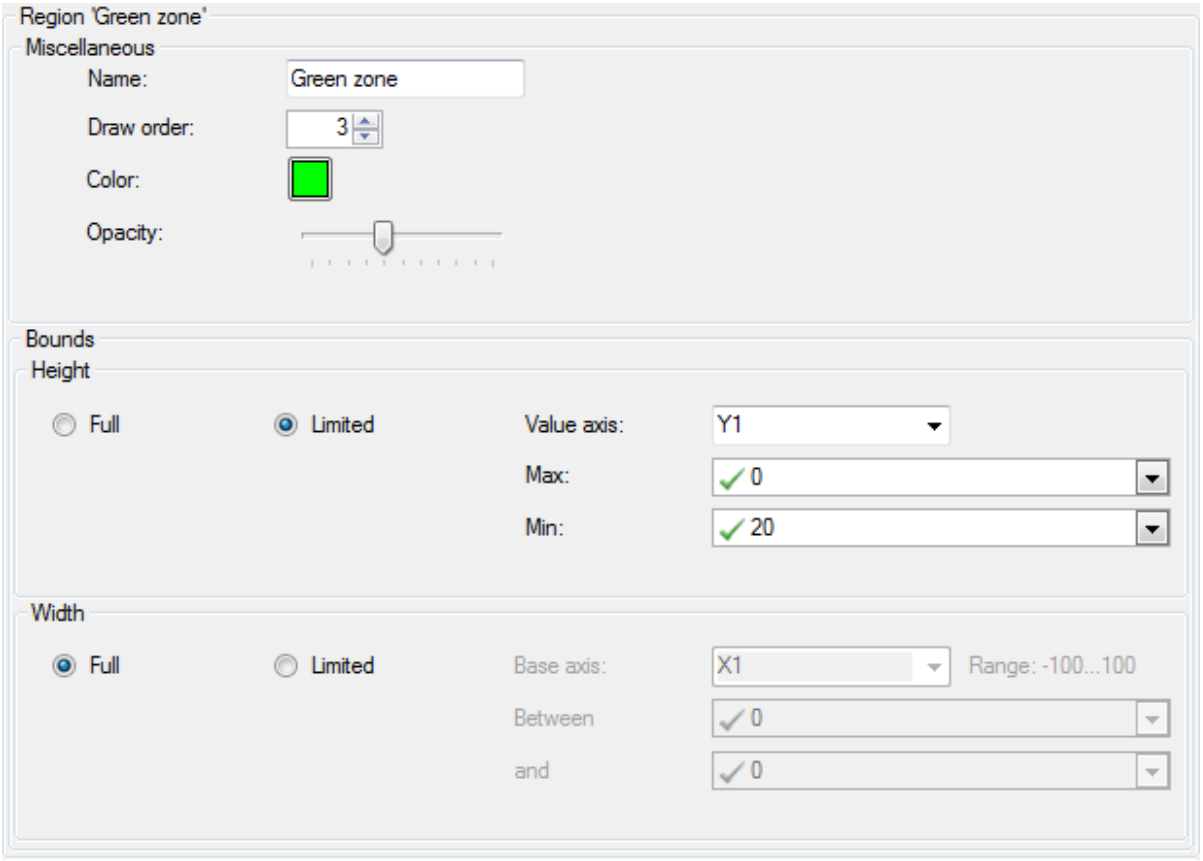
Figure 5 - Relative position of on image item

The image can be stretched to fit the Chart, resized to a desired width and height or kept at its original size. When the *Original size* or *Manual size* mode is selected, the top-left corner (relative to the Chart's top-left corner; see Figure 5) of the image can be configured.

Images can also be dropped from Windows Explorer into the Chart. By default, the image will be drawn transparent, stretched to fit the Chart and assigned to the first base and value axes.

14.4.3 Region

A region is a defined, colored area within the Chart and can only be adjusted in the properties dialog.

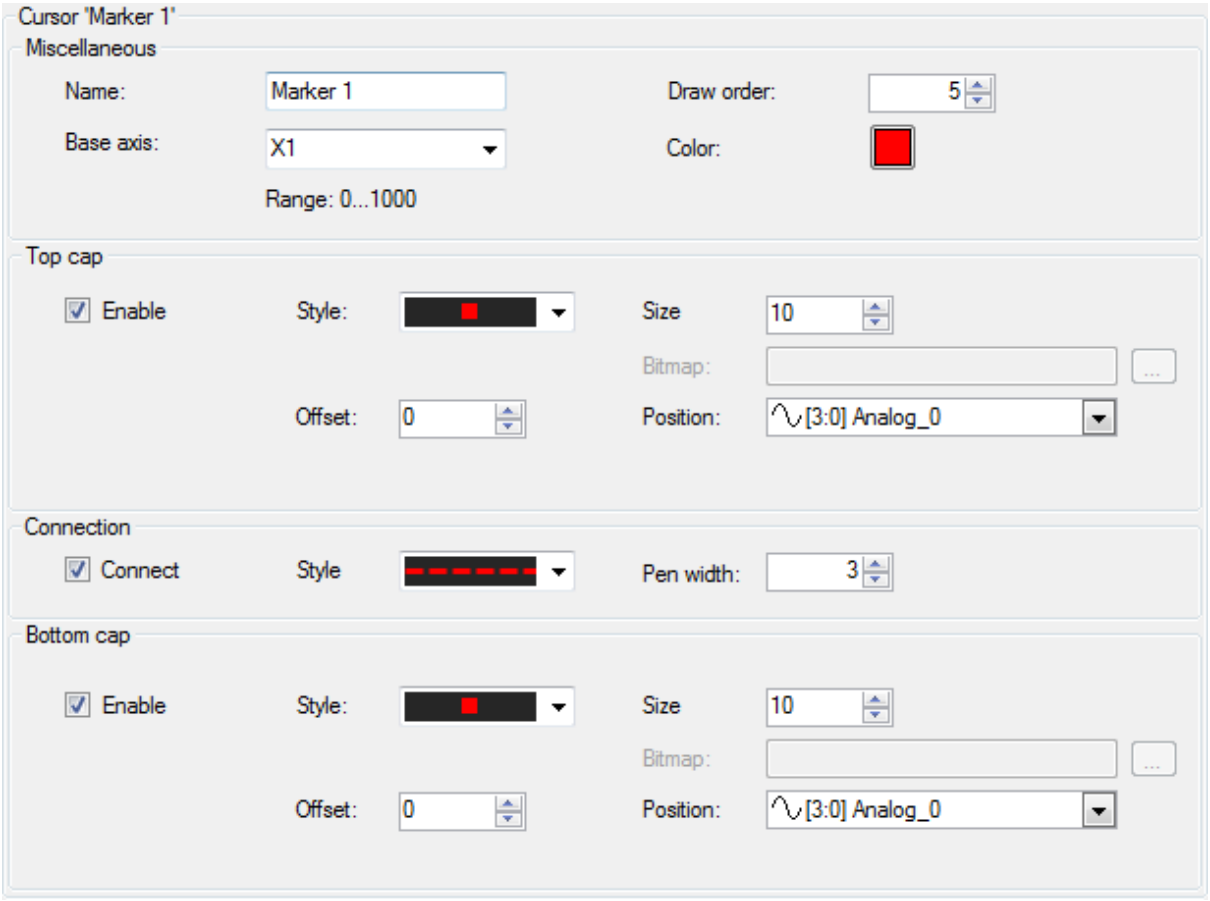


The appearance of a region is defined by its color and opacity which can be set in the *Miscellaneous* section.

For both the height and the width, the user can choose whether to stretch the region or to have it bound by fixed or dynamic coordinates. Values exceeding the axis' range will be clipped (e.g., if the base axis ranges from 0 to 10, a value of -5 will be interpreted as 0).

14.4.4 Cursor

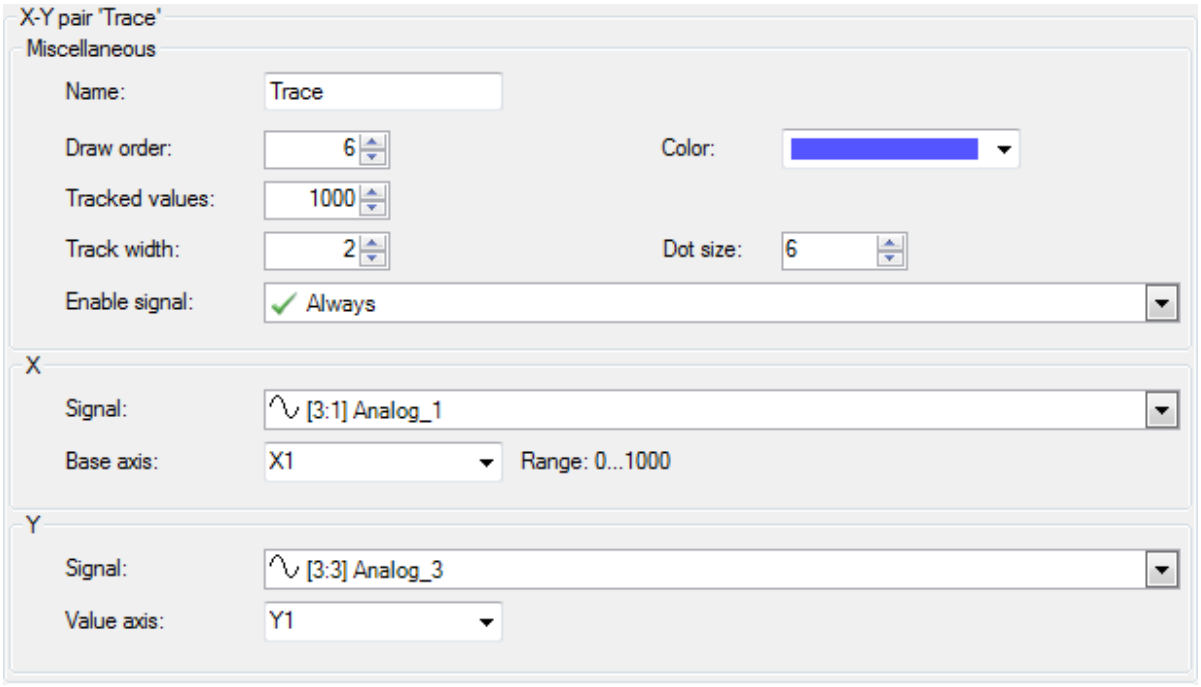
A cursor is a visual item that is able to move over the Chart. It can be used for example to indicate position of certain devices and can be displayed in several visual styles.



A general cursor consists of a top cap, a bottom cap and a connecting line, all of which can be made visible or invisible. Apart from the standard triangle and rectangle, it is also possible to supply an image file (which will be drawn to its original size). The position of both caps can be configured independently by a fixed or dynamic value.

14.4.5 X-Y pair

An X-Y pair is a dot in the Chart of which the coordinate is defined by the current value of two signals. It is also possible to plot a defined number of previous coordinate values which allows for drawing e.g. Lissajous-figures.



The X- and Y-coordinate that define the point are linked to a base and value axis respectively.

In the configuration dialog, the number of previous or tracked values can be configured as well as the size of the previous and current point.

As long as the digital *Enable signal* is high (or set to *Always*), a new point will be drawn; in case the signal is low, the X-Y pair will remain at its current position until the signal becomes high again.

Note that applying the isometric perspective (see section 14.6) in combination with displaying a large amount of tracked values can become very costly in terms of available memory resources.

14.5 Legend

Depending on which setting is configured in the *Visuals* node (see section 14.2), the legend displays the names of all items that have been added to the view or only those of curves and X-Y pairs. As is the case for e.g. a Trend Graph, the items are grouped per value axis.

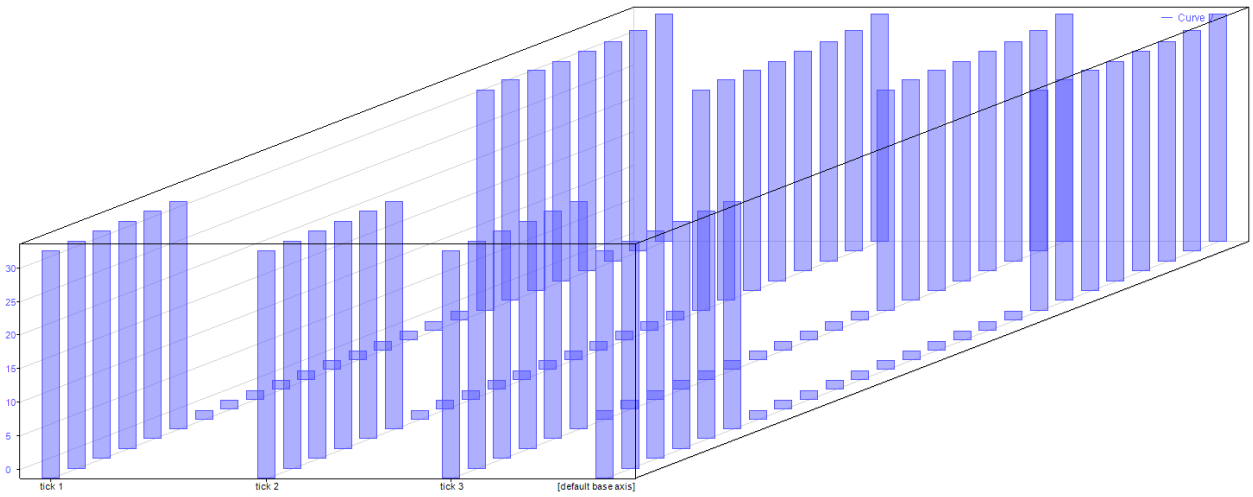
If a signal of a curve is not valid an exclamation mark is shown at the end of the signal row. The legend also provides drag and drop functionality: an item in the legend can be dragged to another, already existing, value axis by pointing to the short line segment preceding the item's name and waiting until a small arrow appears. It is also possible to create a new value axis for the item by dragging it elsewhere in the legend.

Note that, in contrast to some other views, the legend does not indicate the order of drawing, which is handled by the *Draw position*.

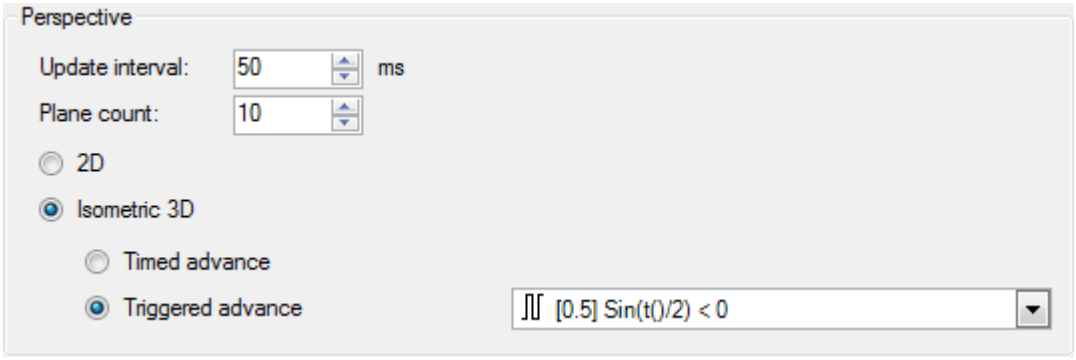
The context menu of the legend allows the removal of items.

14.6 Isometric perspective

The Chart can be configured to use an isometric 3D perspective instead of a 2D representation, as it is done in the FFT view.



Configuration of the isometric view is done in the main node of the Chart's properties dialog



When selecting *Isometric 3D*, additional options will become available to indicate whether the Chart should advance to a next plane at every time step or based on a trigger signal.

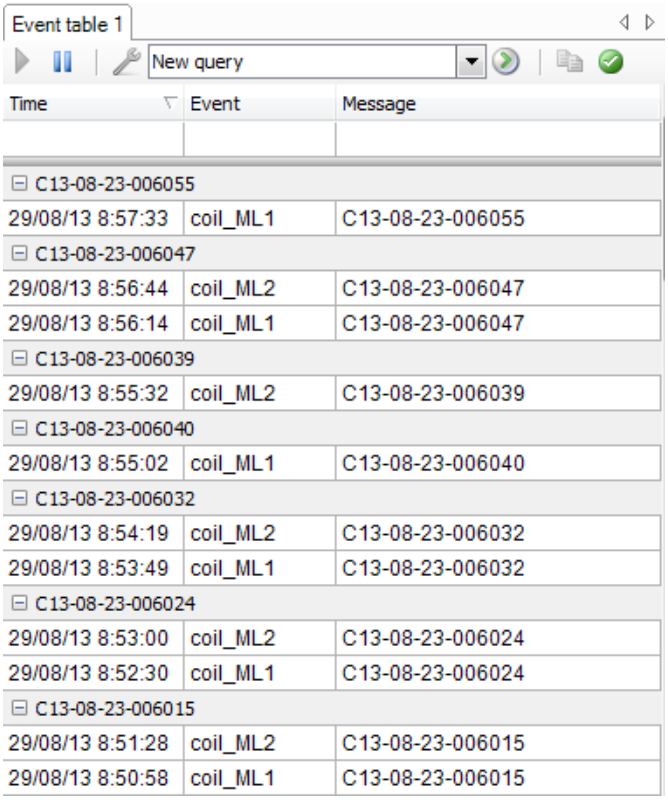
The item data will be drawn on a z-axis with the newest data closest to the axis origin to create this specific waterfall effect. Images and regions will only be drawn to the back plane; text items are drawn on the front plane.

Note that using the isometric perspective is more resource demanding than using the 2D representation.

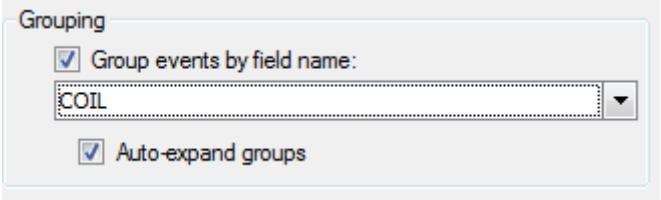
The axis layout settings are overridden in this mode. Axes are always displayed at the side of the Chart that does not overlap with the perspective flow direction. While using the isometric perspective, the zoom function is restricted to the front plane.

When the isometric perspective is enabled, the perspective itself is configurable by dragging the back plane center point. The back plane center point is marked with a small cross, and the cursor will change once the mouse cursor hovers over it. During dragging, the items disappear and the perspective is adjusted.

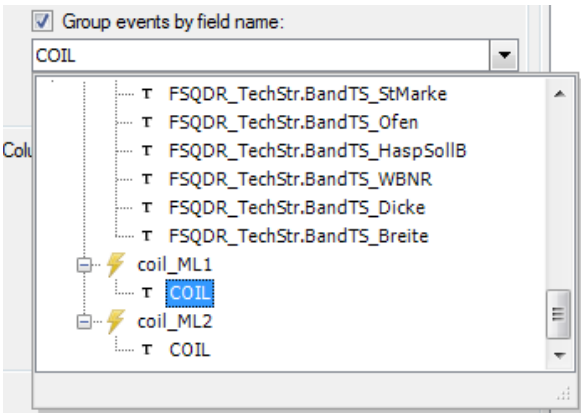
15 Event table row grouping



It is now possible to group event table rows. To group multiple events they need to have an equal field name. This can be configured in the event table properties:



When you click the dropdown button a list of all events on the connected HD server and their fields is displayed:



As you can see, the events coil_ML1 and coil_ML2 both have a field called ‘COIL’ that is used to group the events. The table will create a group row for each of the field values and place their parent event occurrences inside the group.

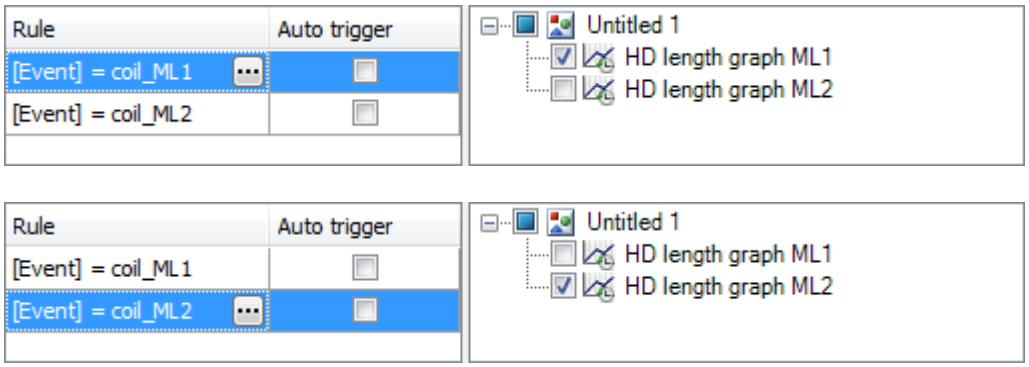
When grouping is activated, it is applied both in live mode, pause mode and query mode.

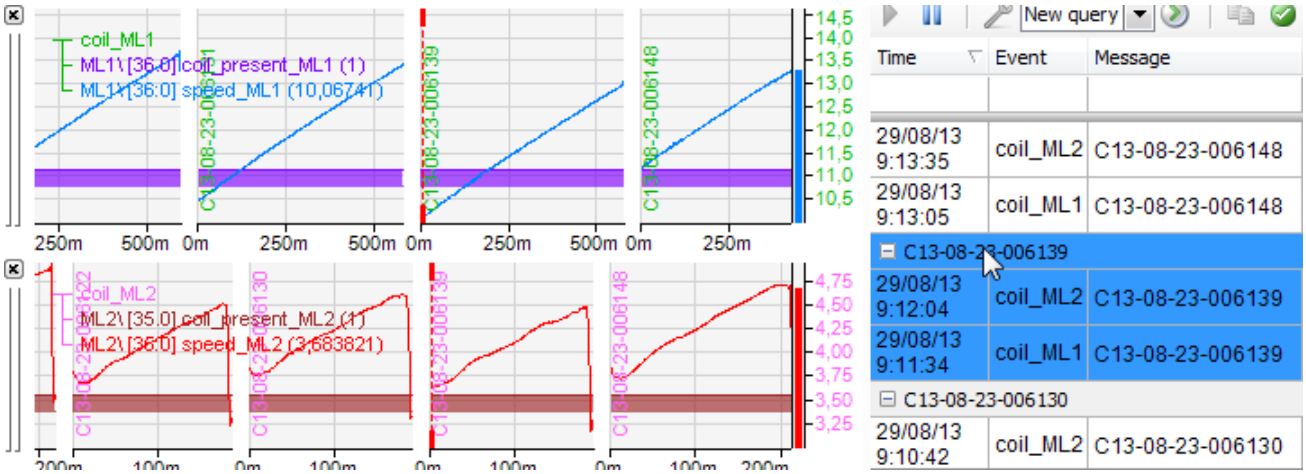
Rows inside a group are sorted as expected. Groups themselves are sorted by comparing their first child row’s sort column value. Not that the first row inside a group always contains the maximal/minimal sort column value when sort order is ascending/descending.

The user can manually expand or collapse groups. Using the table context menu, he can expand or collapse all rows. In the properties dialog he can also choose to auto-expand groups.

It is possible to double click a group row. If you do this, all events inside the group will be fired. Using the command rules in the settings dialog, the user can send certain commands to certain HD trend graphs in the layout. If a HD trend graph receives multiple commands it will execute only one of them.

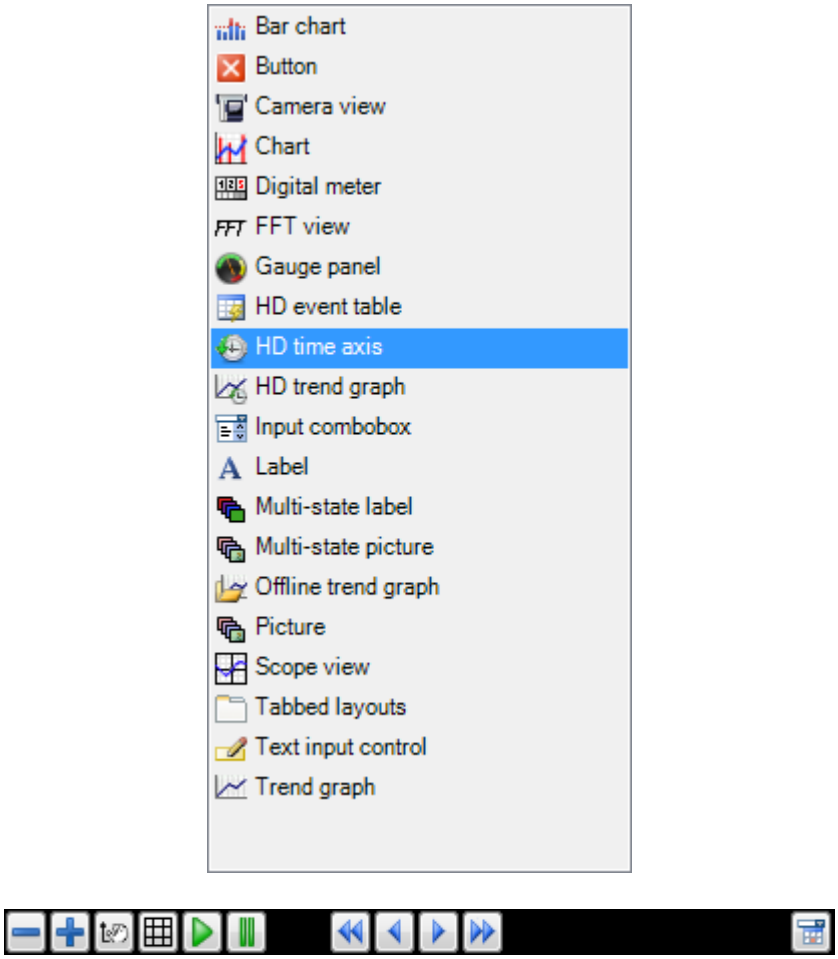
A possible scenario when using group double clicks is to display a certain product from a HD length store on multiple measuring locations:



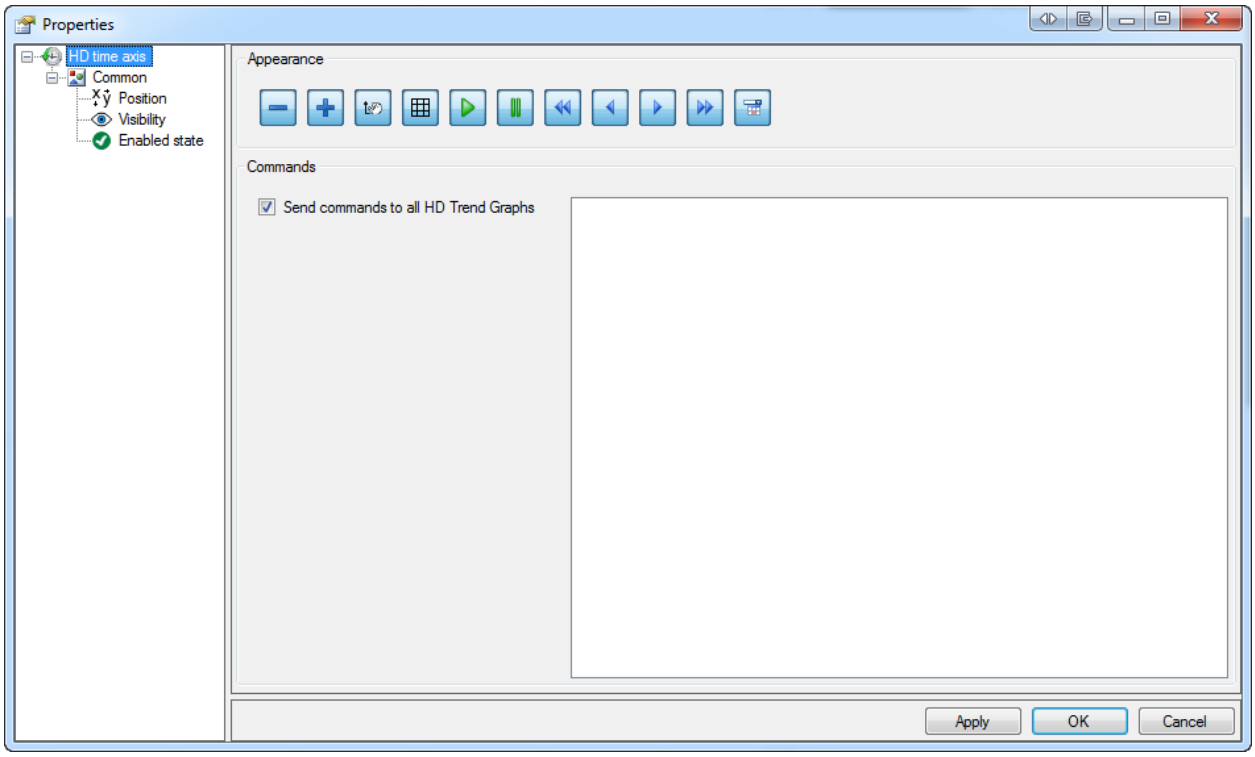


16 QPanel – Dynamic object properties

16.1 HD Time Axis control

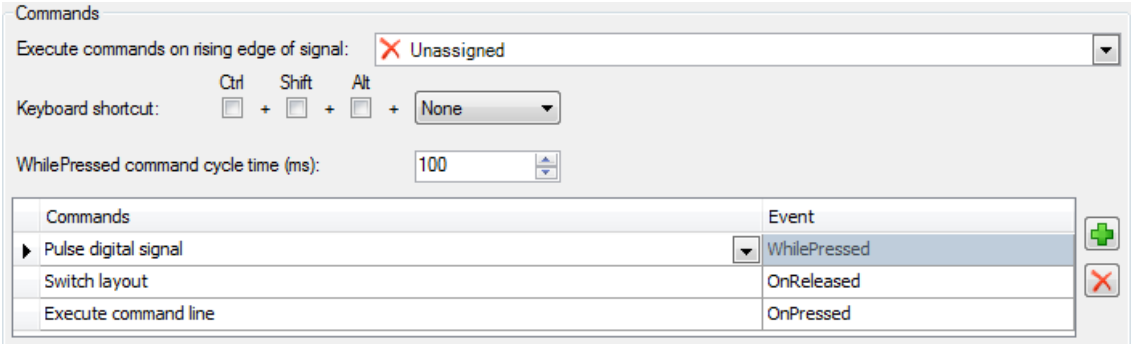


A new control has been added to the QPanel called HD time axis. Using the HD time axis, it is possible to control several HD trend graphs simultaneously. The zoom and pan buttons, found in individual HD trend graphs are available, as well as a play and pause button and a zoom reset button to restore all linked HD trend graphs to their original state after zoom operations have been performed.



In the properties window of the HD time axis it is possible to configure which buttons are displayed: e.g., it is possible to only offer play or pause functionality. Since it is possible to have several HD time axes in one QPanel, the user can customize the position of the individual buttons: in the most extreme case this means that there are 10 HD time axis controls, each displaying only one button. That way, the position of the buttons can be fully customized.
To configure which HD trend graphs are addressed by the HD time axis, the user has to select the desired HD trend graphs in the tree structure on the right. In case all HD trend graphs need to be addressed, checking “Send commands to all HD Trend Graphs” is sufficient.

16.2 Command button event can be specified



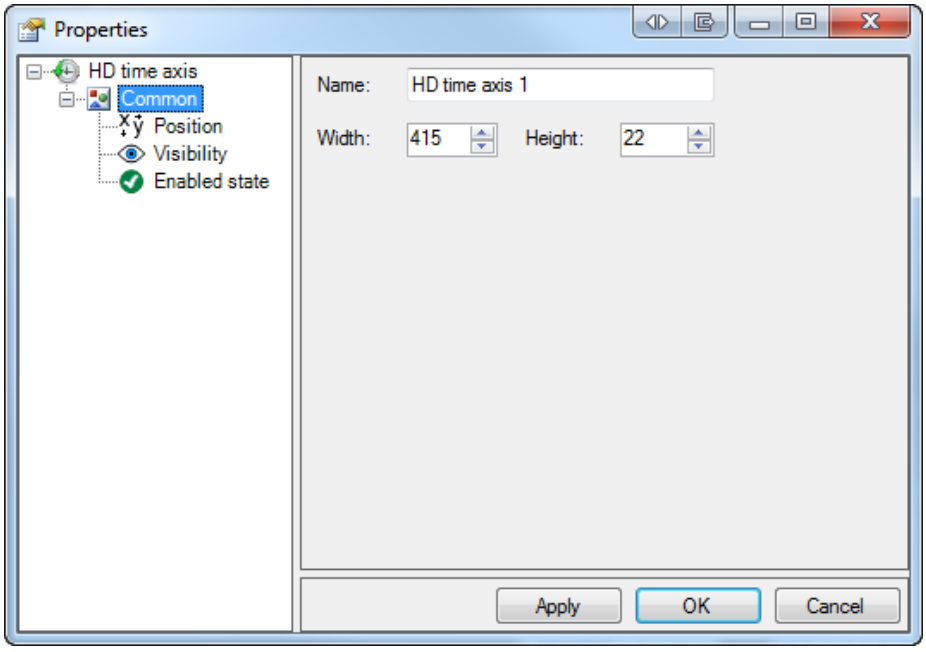
For command buttons, it is now possible to define when a command is executed. The three possible options are: when the button is pressed, while the button is pressed or when the button is released. The event is linked to individual commands so it is possible to define complex behavior for a command button.
The WhilePressed event is however a special case: first of all, the user has to define a repetition rate with which the configured command is repeatedly executed while the button is being pressed. Second, only a few relevant commands are allowed to be used in combination with WhilePressed behavior:

- Execute command line
- Pulse digital channel
  - Upon pressing the button, the digital channel is set to 1; upon releasing the button, the digital is set to 0 (regardless of the original value)
- Set signal value
  - The selected value is written to the selected channel upon pressing the button; when releasing the button, the original channel value is restored
- Pause
  - When pressing the button, the related views are paused; upon release the related views are resumed
- Resume
  - Inverse behavior of pause command

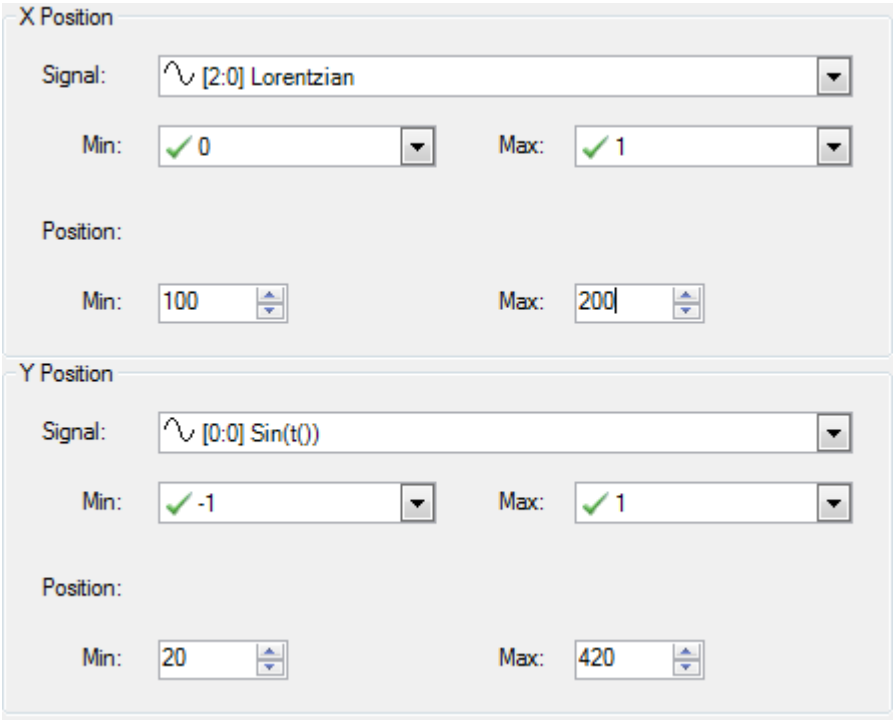
16.3 Auto-hide toolbar

The QPanel toolbar has an auto-hide feature: by right-clicking the toolbar and selecting Auto-hide, the toolbar will disappear. When hovering the mouse over the top of the QPanel, the toolbar will appear again. This setting can also be configured in the general QPanel properties.

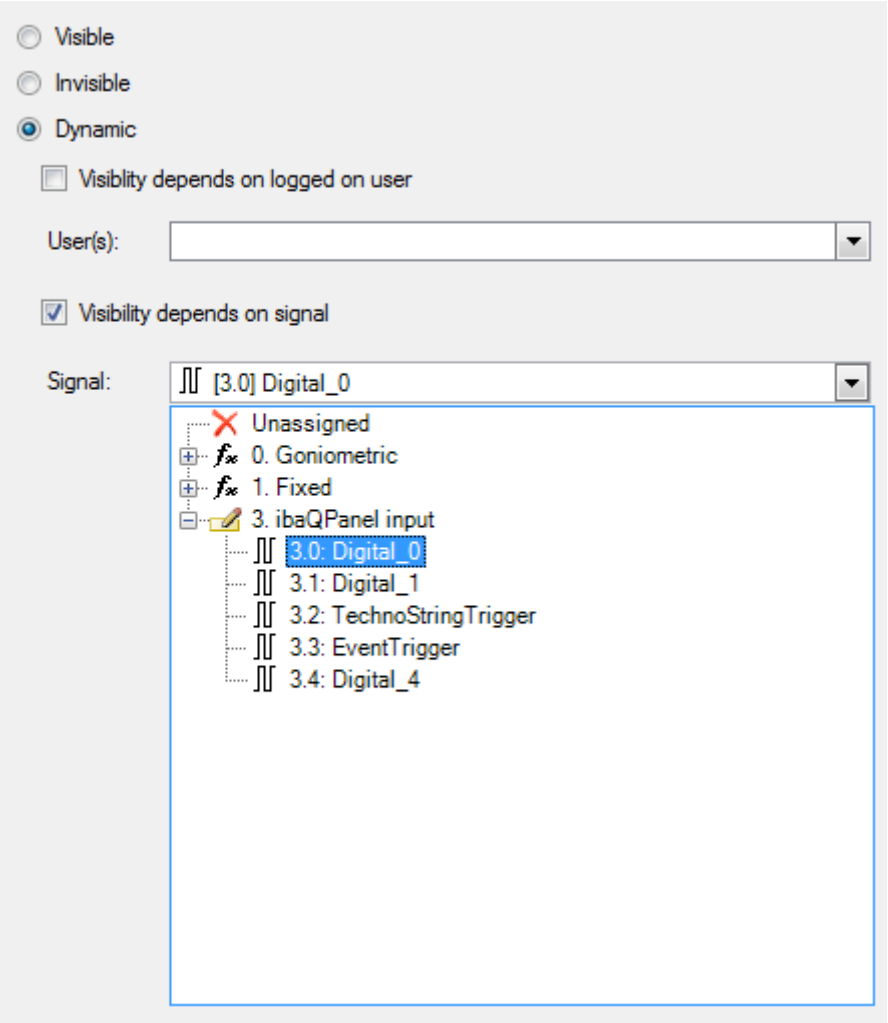
16.4 Dynamic properties of all objects



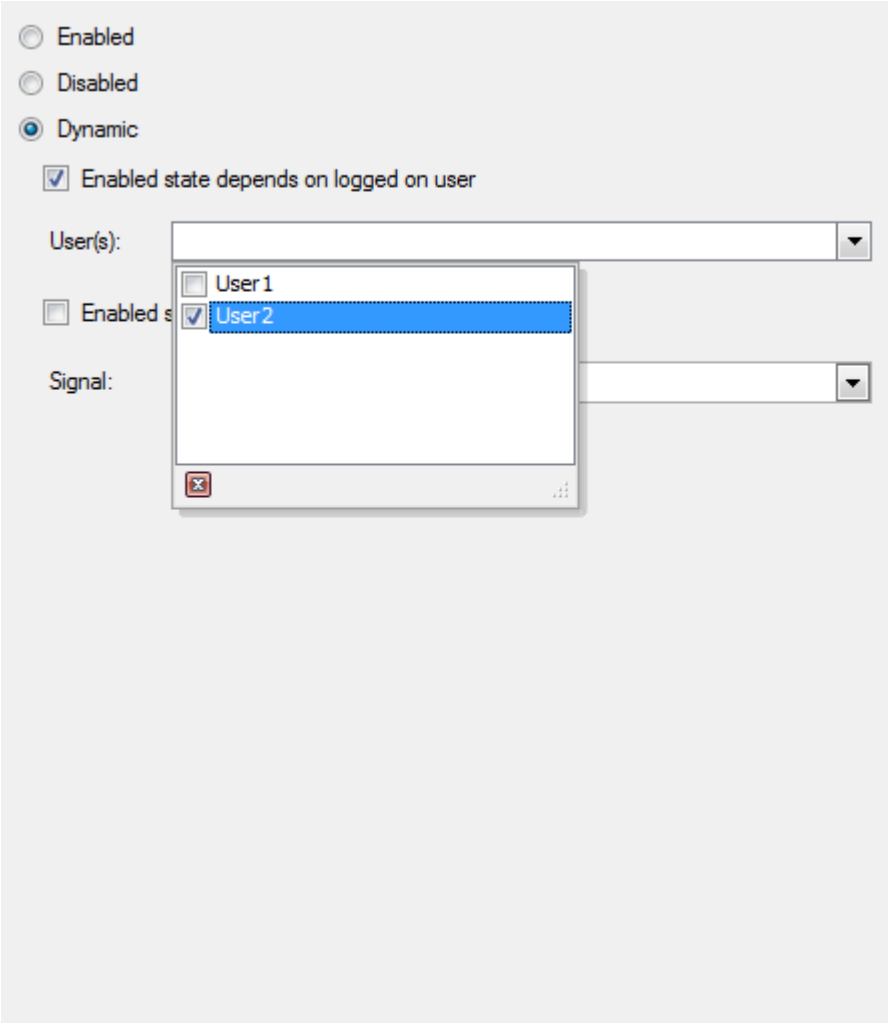
All QPanel objects now have an extra Common node in the tree in the properties window. Here, the name of the object can easily be modified as well as its width and height. Three subnodes allow the user to configure the position, visibility and enabled state of objects in a dynamic fashion.



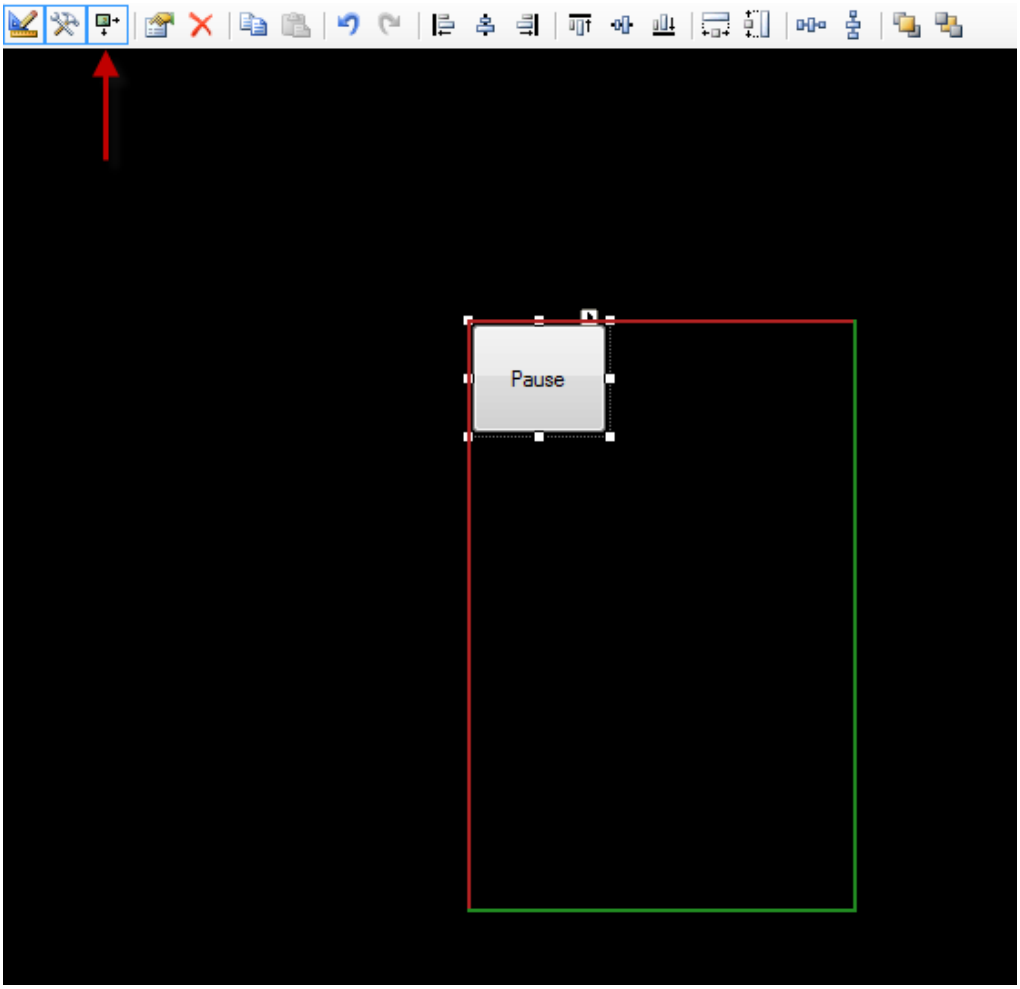
In the dialog corresponding to the Position node, the user can link the X- and Y-position to ibaPDA signals. After a signal is selected, the user has to define the range in which the signal operates. Then, the value of the minimum and maximum X- or Y-position has to be configured. In the above example, when signal [2:0] has a value of 0 or lower, the X-position of the object will be 100. When the signal has a value of 1 or higher, the X-position will be 200. For values in between, the X-position will change linearly with the signal value.



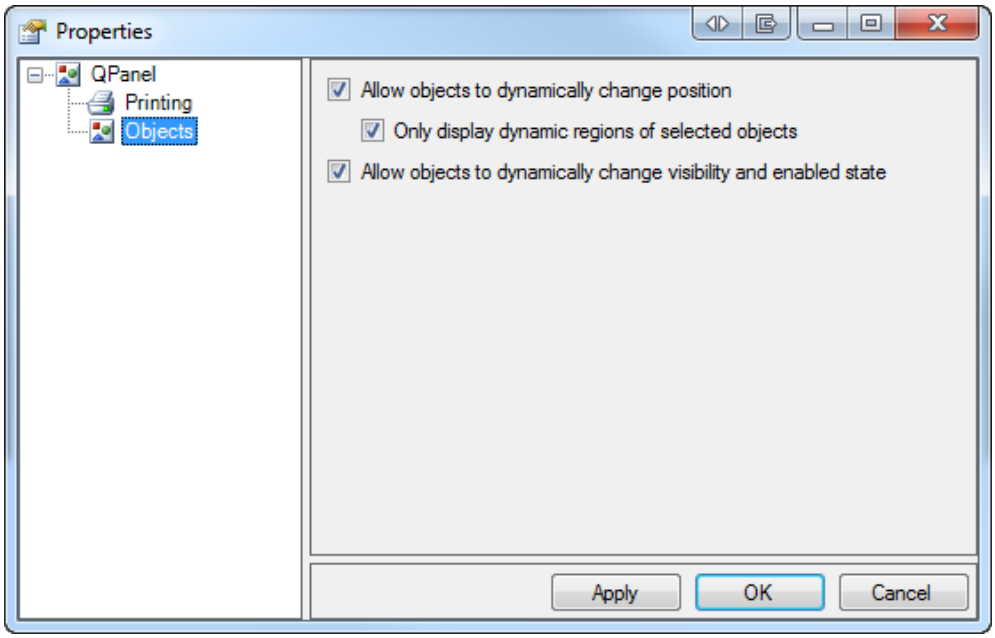
For the visibility of each object, the user can define whether an object is always visible or invisible, or whether this property depends on the logged on user or on a selected signal. This can be done by checking the corresponding boxes and defining corresponding users or a signal. When both options are checked, both conditions should be satisfied in order to display the object. Note that the user condition is always fulfilled for the administrator.



The enabled state of an object can be configured in a similar way.

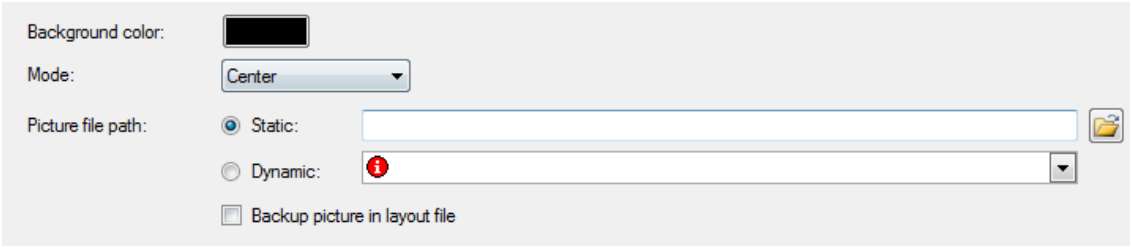


In order to help the user visualize in which region an object can dynamically move, the concept of dynamic regions was introduced. By pressing the third button from the left in the QPanel toolbar these regions can be shown or not. It is also possible to toggle the visibility of dynamic regions using the context menu.
When selecting an object, apart from the usual resize handles, there will also be a rectangle consisting of red and green lines. The enclosed region defines the boundaries in which the object can move. The top and left lines of the rectangle are marked in red since they cannot be resized or moved: the top left corner of an object is always anchored to the top left corner of the dynamic region. The bottom and right lines of the rectangle can be resized, either individually or simultaneously by positioning the mouse pointer over the bottom right corner and dragging the appearing handle.
Note that in design mode, an object is always positioned at the minimum coordinate.



In the properties window of the QPanel, dynamic behavior of all objects can be enabled or disabled. The user can also decide whether the dynamic regions are only displayed for the selected objects or for all objects on the QPanel. This can be useful for aligning several regions with respect to eachother.

16.5 Backup picture in layout file

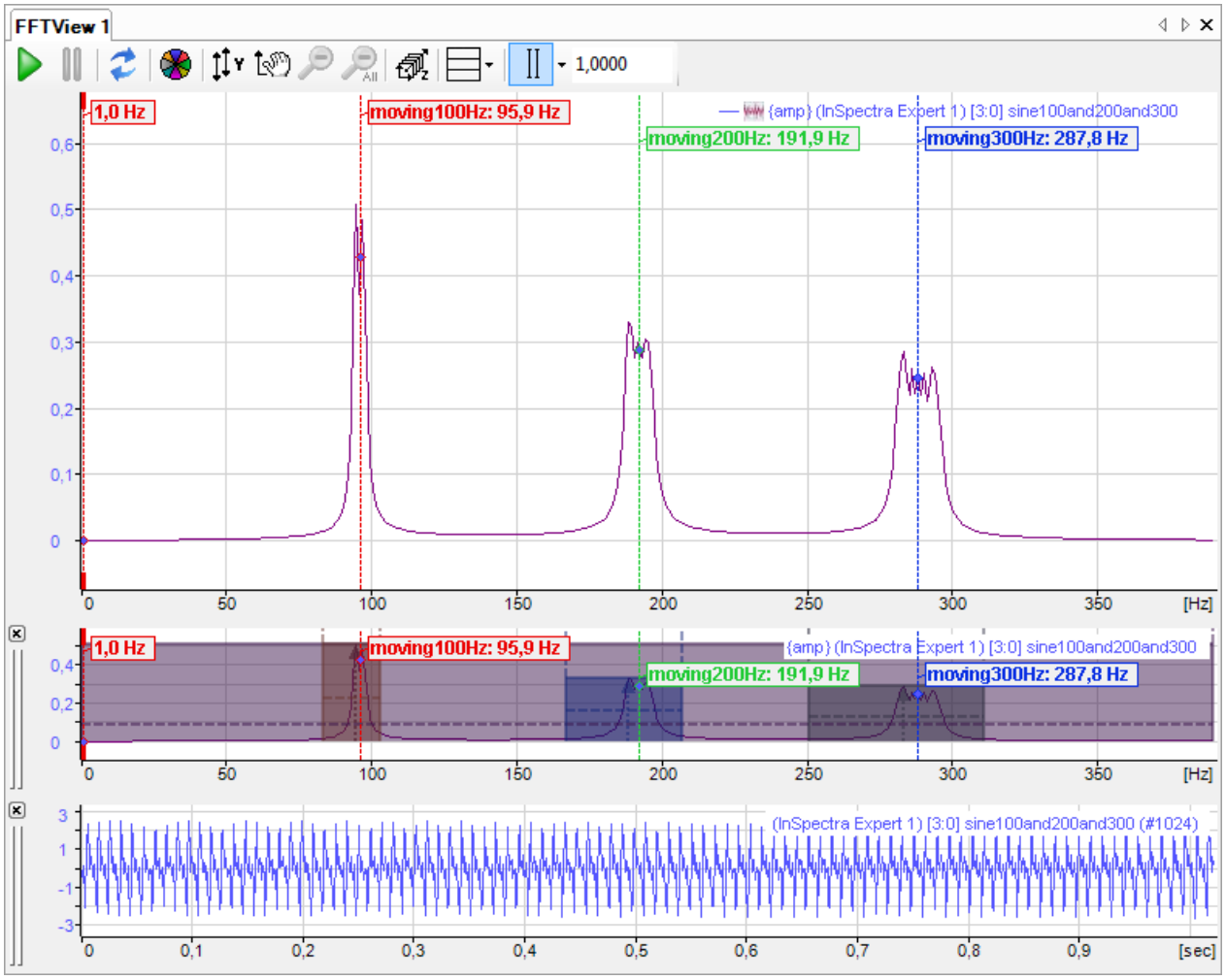


Before, in a single-state picture, only the path to an external picture file was saved. When porting an ibaPDA layout file to another computer, the path might become invalid. Now, there is an option to backup the last-known picture at the configured path in the layout file. QPanel will then display this backup picture instead of the default iba logo in case the path is invalid.

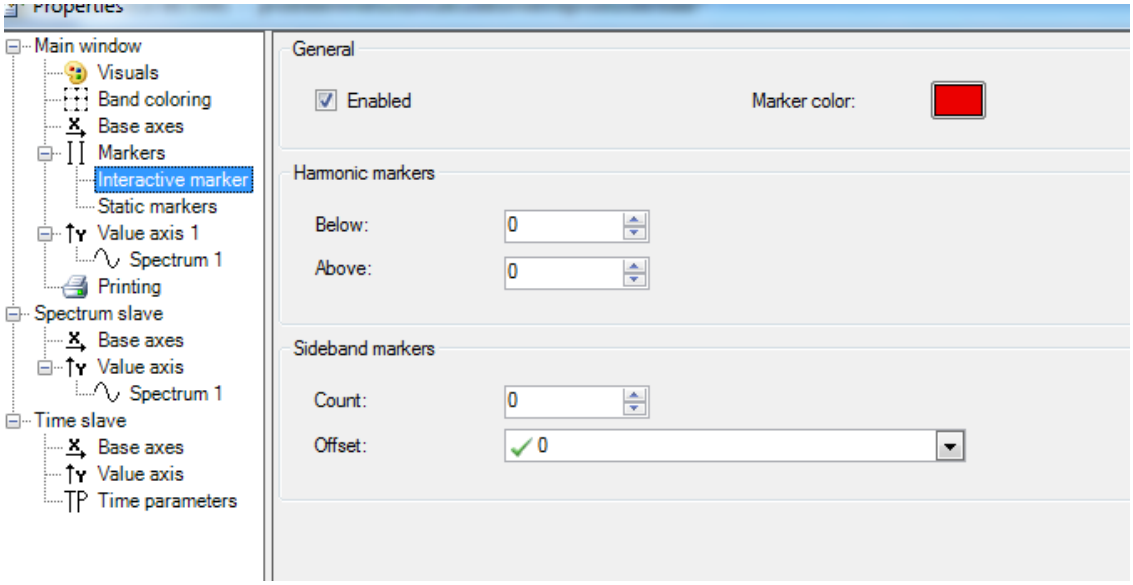
17 FFT view

17.1 Interactive and static markers

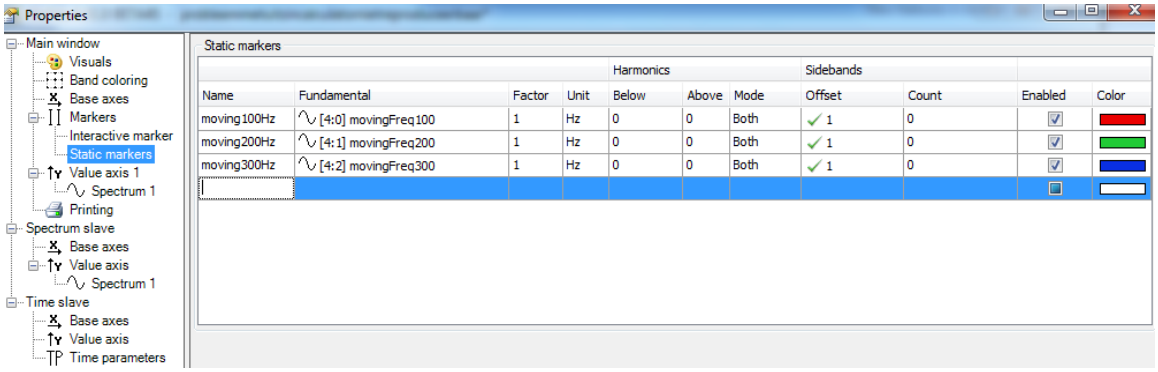
We introduced a distinction between static markers and the interactive marker. In previous versions of ibaPDA, there was an interactive marker only. You can move the interactive with the mouse. The positions of the static markers are determined automatically. For both types of markers, you can define harmonic and sideband markers. Below, you see an example of an FFT view having three static markers next to the interactive marker which is on the left.



You can customize the interactive marker in the properties dialog of the FFT view:



You can change or add static markers in the following table which is also in the properties dialog of the FFT view:



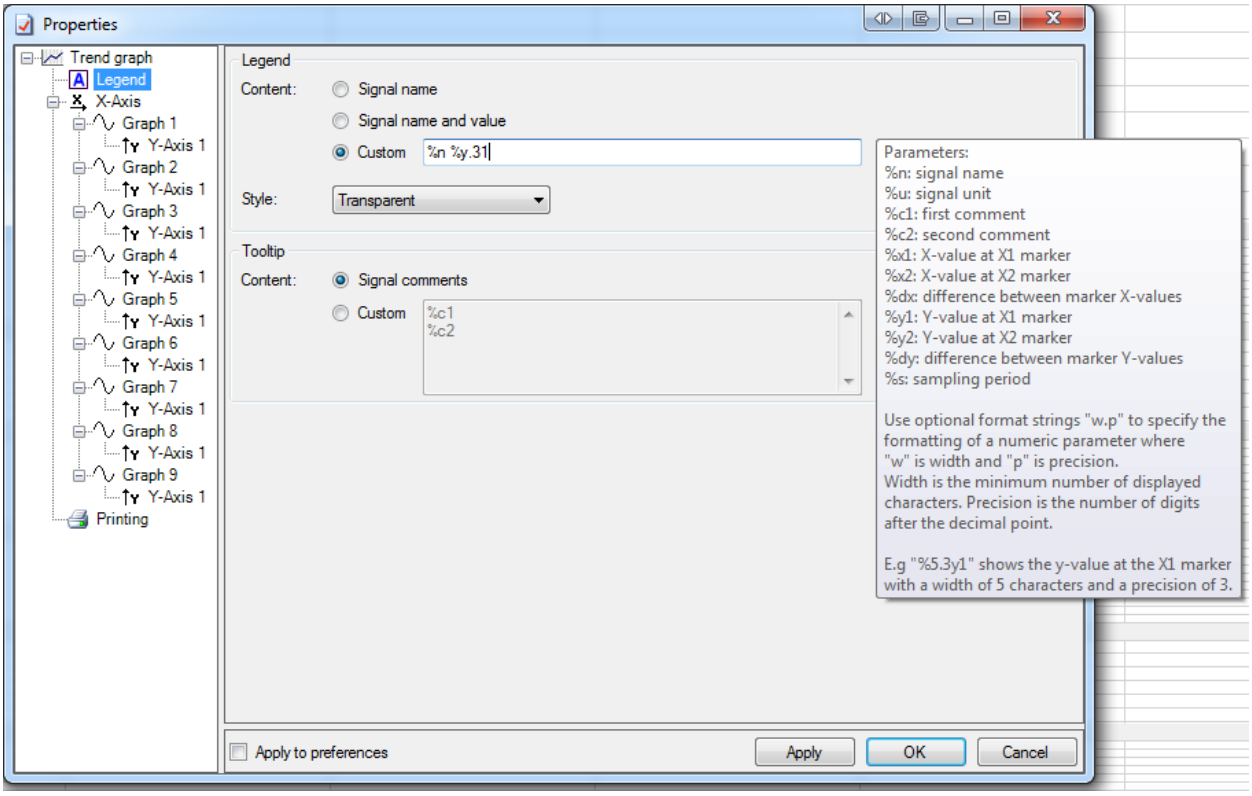
Note that the fundamental as well as the sideband offset can be a signal. In the factor column, you can specify a constant by which the fundamental will be multiplied to get the marker's position.

In case a static marker depends on a signal, the position of the marker is calculated as an average of the signal. The time range for the average corresponds to the time range of the FFT of the synchronization curve. The synchronization curve is a setting in the general properties. In case there is only one curve on the FFT view, then the synchronization curve is always equal to this curve. This property is named synchronization curve because the marker positions are synchronized to the FFT of that curve.

17.2 Bands and labels

From version 6.31 on, also the band and label positions depending on a signal are calculated as an average based on the synchronization curve (as explained in the previous section).

18 Custom legend and tooltip in trend graphs



Similar to ibaAnalyzer, it is now possible to configure a parameterized string for the legend and its tooltip in normal and HD trend graphs. A tooltip that appears once the user starts editing either the custom text field of the legend or the tooltip explains which parameters can be used. Some parameters have no significance in live mode (e.g. parameters related to the second marker which is then not displayed); these values are then replaced by a -. Note that the current signal value corresponds to %y1 (even when no markers are displayed).

The values are formatted in the same way as the marker grid. This means that the number of decimals and the scientific notation is dependent on the axis settings and zoom levels. It is also possible to apply formatting to numeric values. The syntax is as follows: %w.pc where ‚c' is the name of the parameter (e.g. x1, dy, …), w corresponds to the desired width of the resulting string. This is the minimum number of characters that will be displayed. Extra spaces will be added to the left in order to get to this minimum number of characters. If the value requires more characters than the specified width then the width is increased. ‚P' is the precision. It determines how many digits after the decimal point are shown.

The configuration of the legend has been moved from the general Trend graph node to a separate Legend node.