ibaPDA v8.3.0

New Features

05.05.2023
iba AG

Table of contents

1 ibaQPanel

1.1 Numeric Up/Down

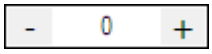
A new numeric up/down control is available in the ibaQPanel.
It contains a numeric signal value that can be incremented or decremented by clicking the buttons or by scrolling with the mouse wheel. The value can also be typed in manually.
This control allows you to change a numeric value quickly, safely and easily.

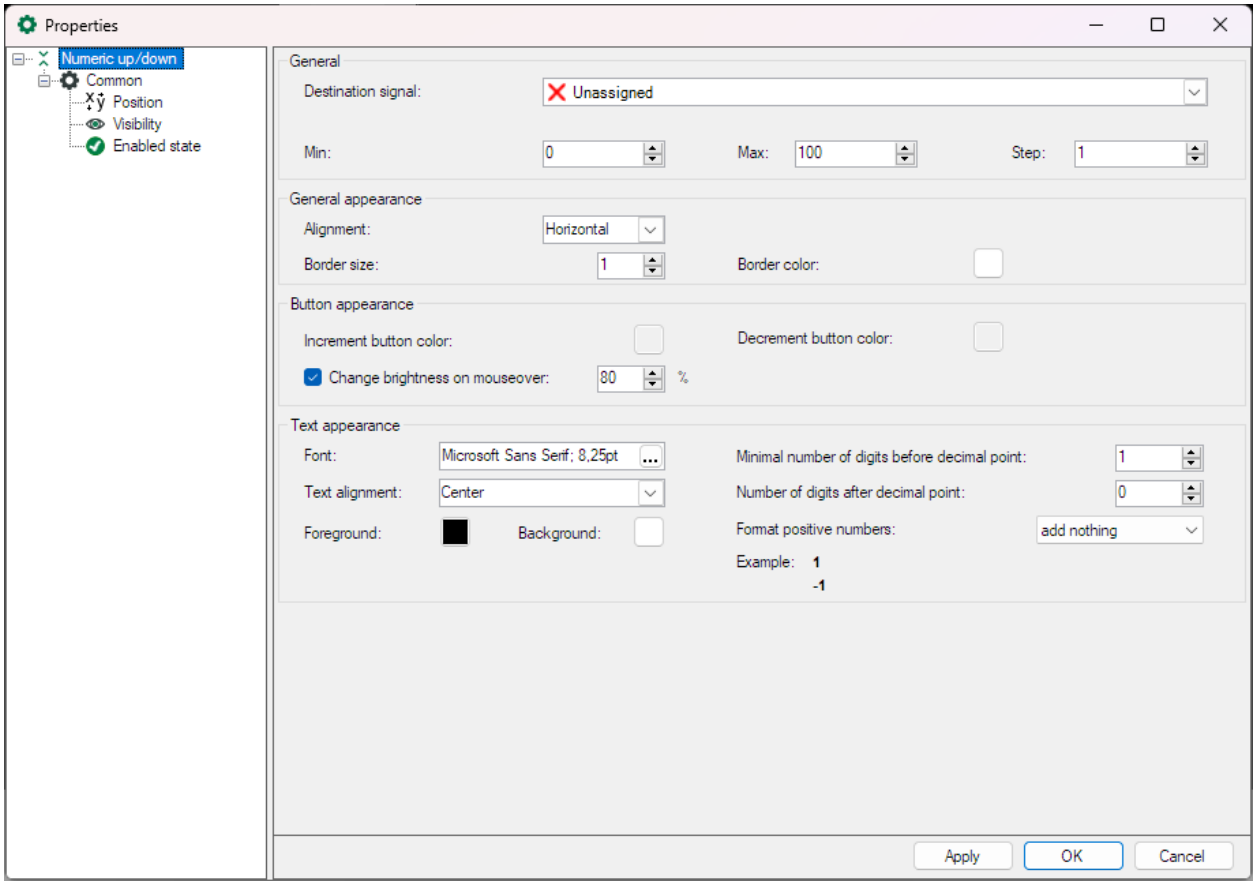
Notes:

- The signal is updated immediately when changing the value.
- The longer a button is pressed, the faster the value increases or decreases

1.1.1 Settings

By default, the settings from the client preferences are applied.





Destination signal
The destination signal contains the value that is used in the control. Only ibaQPanel input signals can be selected.

Min, Max and Step
Min and *Max* define the limits of the signal value, *Step* defines the value used to increment or decrement the signal.

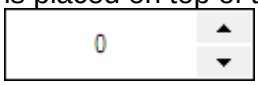
Notes:
- *Step* cannot be greater than the difference between *Min* and *Max*.
- *Min*, *Max* and *Step* always have the same number of decimal places.
- The value cannot exceed the limits. If it does, it will be reset to the respective bound.

Adjusting the precision of the *Min*, *Max* or *Step* property will not adjust the *displayed number of decimal places*. Use the *Number of digits after decimal point* option for this.

Alignment
The alignment defines the position of the increment and decrement button.

Horizontal places the decrement button on the left side and the increment button on the right side.

Vertical places both the decrement and increment button on the right side. The increment button is placed on top of the decrement button.



Border
If *0* is defined as the *border size*, the border will be removed. Otherwise, the specified size is applied to the border.

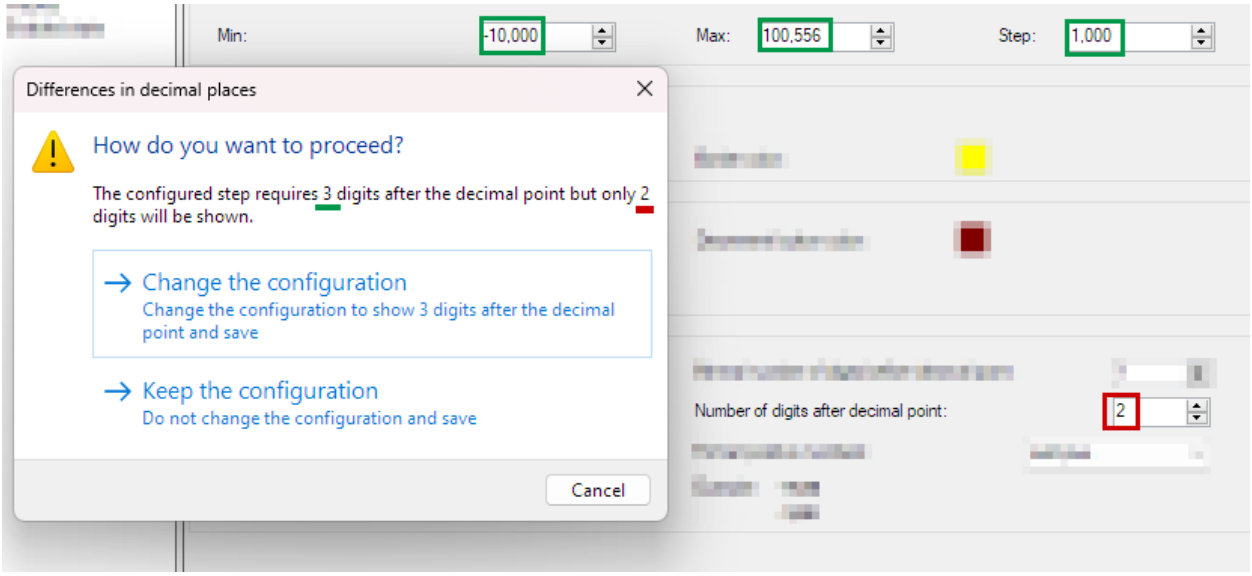
Font
The selected font is used to display the value.

Number of digits after decimal point
This setting defines the *displayed* number of digits after the decimal point. It may or may not match the number of digits after the decimal point of the Min, Max and Step property (see *Precision*).

1.1.2 Precision

The displayed number of digits after the decimal point may or may not match the precision of the Min, Max and Step properties.

If they do not match, a message box appears that can either adjust the difference or – if intended – keep it.



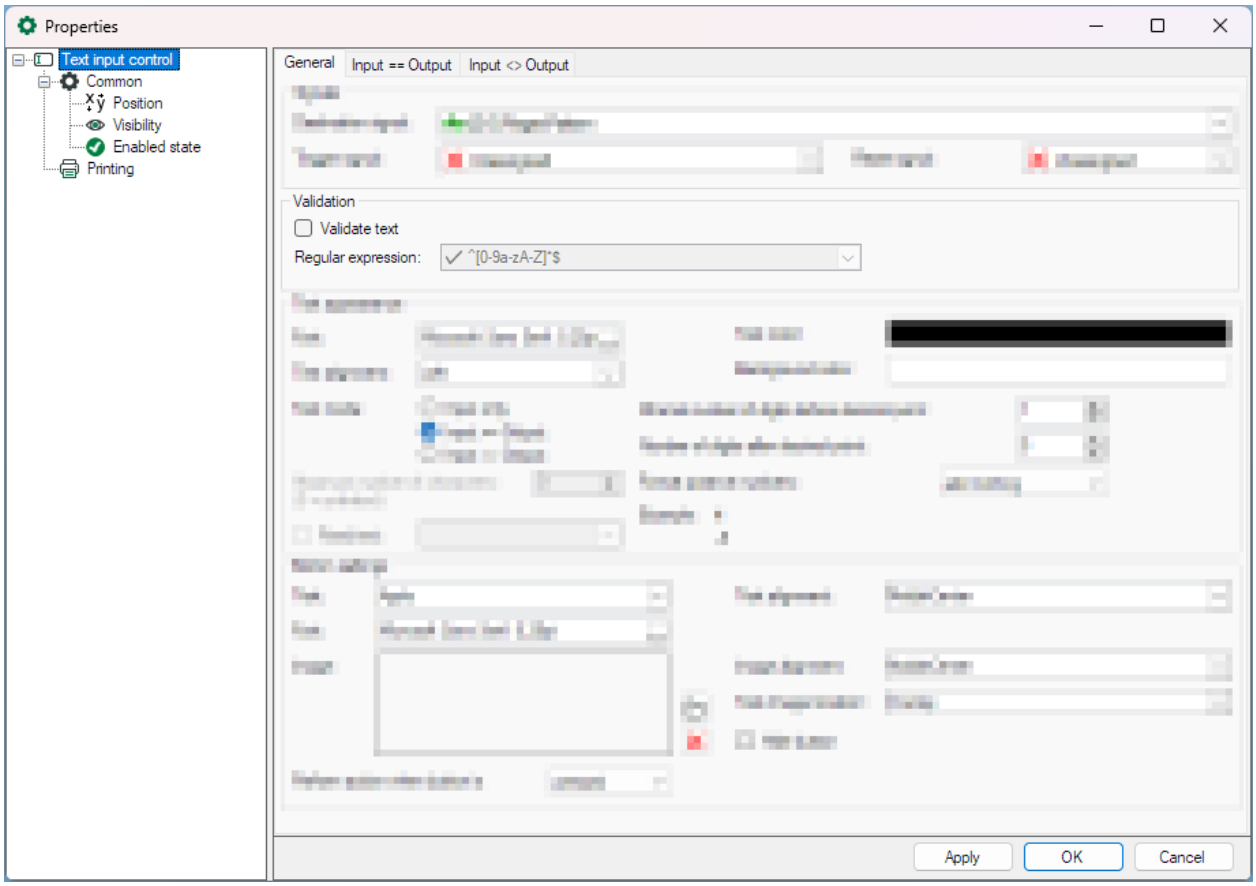
Change the configuration
This option will adjust the *number of digits after decimal point to* 2 and save it.

Keep the configuration
With this option, no changes are made.
Also, the warning will be suppressed until a setting related to the number of digits after the decimal point has been changed.

1.2 Text input validation

The text of a text input control can be validated with a regular expression. This is mainly used to ensure that the text only contains defined, valid characters.

To activate the validation, the new *Validate text* option can be selected in the properties.



By default, this option is based on the client preferences.

1.2.1 Static text as pattern

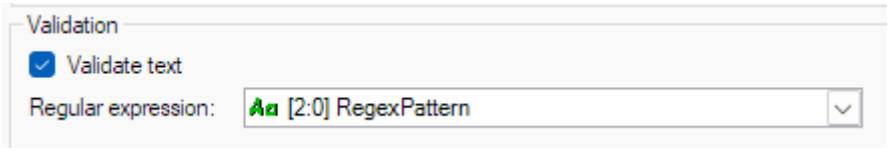
A static regular expression can be entered in the *Regular expression* entry. The default pattern **^[0-9a-zA-Z]*$** allows any characters from a-z, A-Z and 0-9, special characters are not allowed.

When the pattern is changed, an icon and the tooltip at the end indicate whether the entered pattern is valid.

If the pattern is invalid, the icon changes and the tooltip informs about the error.

1.2.2 Text signal as pattern

Besides using a static text as the pattern, a signal can be used as well.



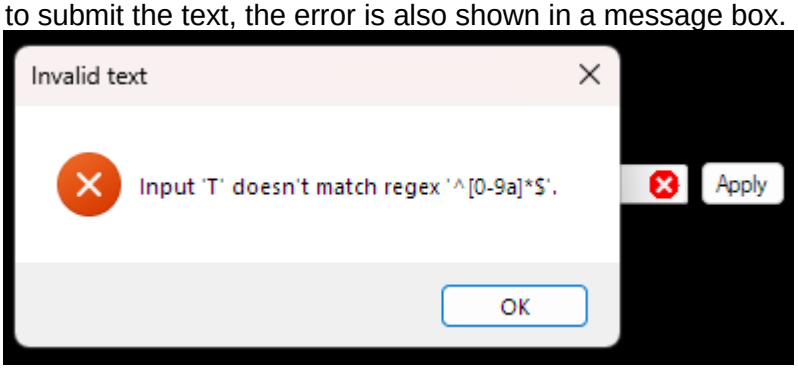
The signal contains the pattern that is used to validate the text. This is especially useful if the pattern has to be changed dynamically.

1.2.3 Text input behavior

If the validation is activated and a text is entered that doesn't match the pattern, the text cannot be submitted.

This is indicated by an error icon with a tooltip.

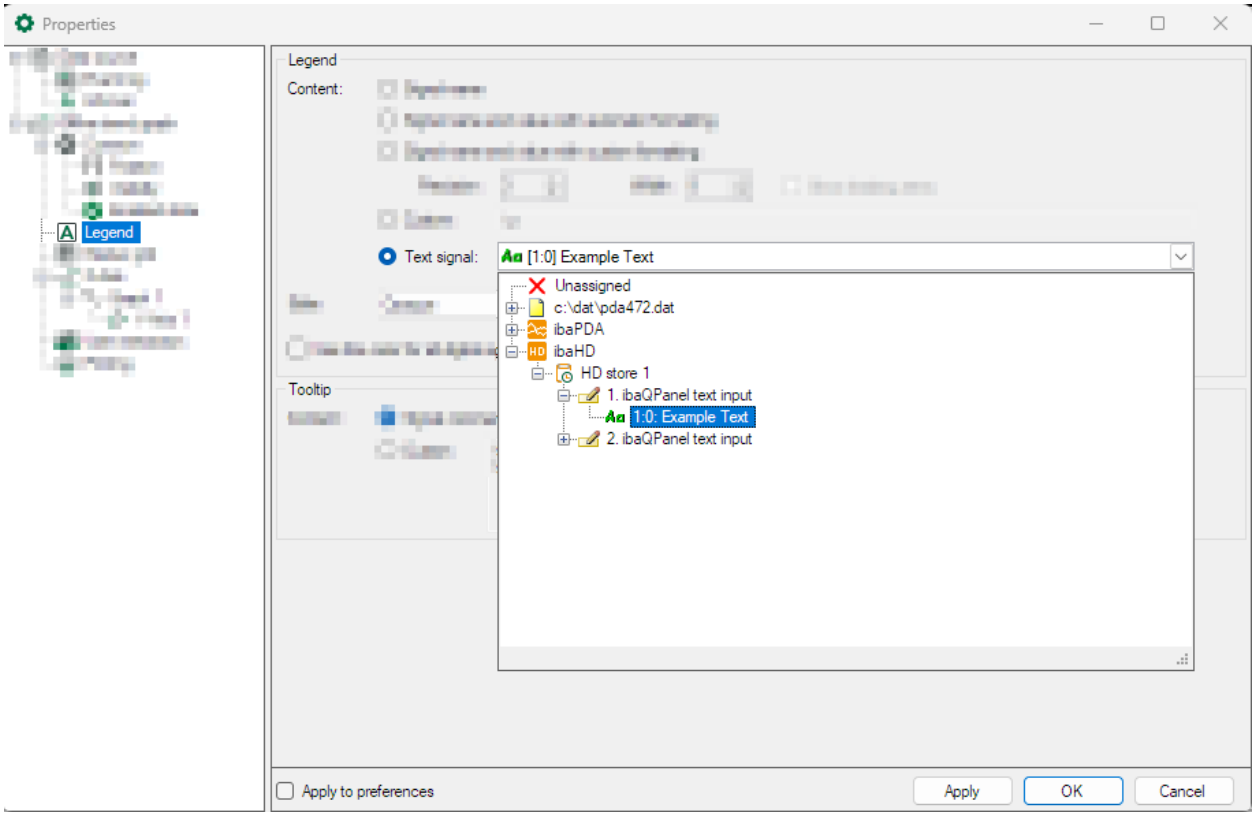
When attempting to submit the text, the error is also shown in a message box.



When a signal is selected as the source of the pattern, the signal may contain an invalid pattern. In this case, a more detailed error is shown.

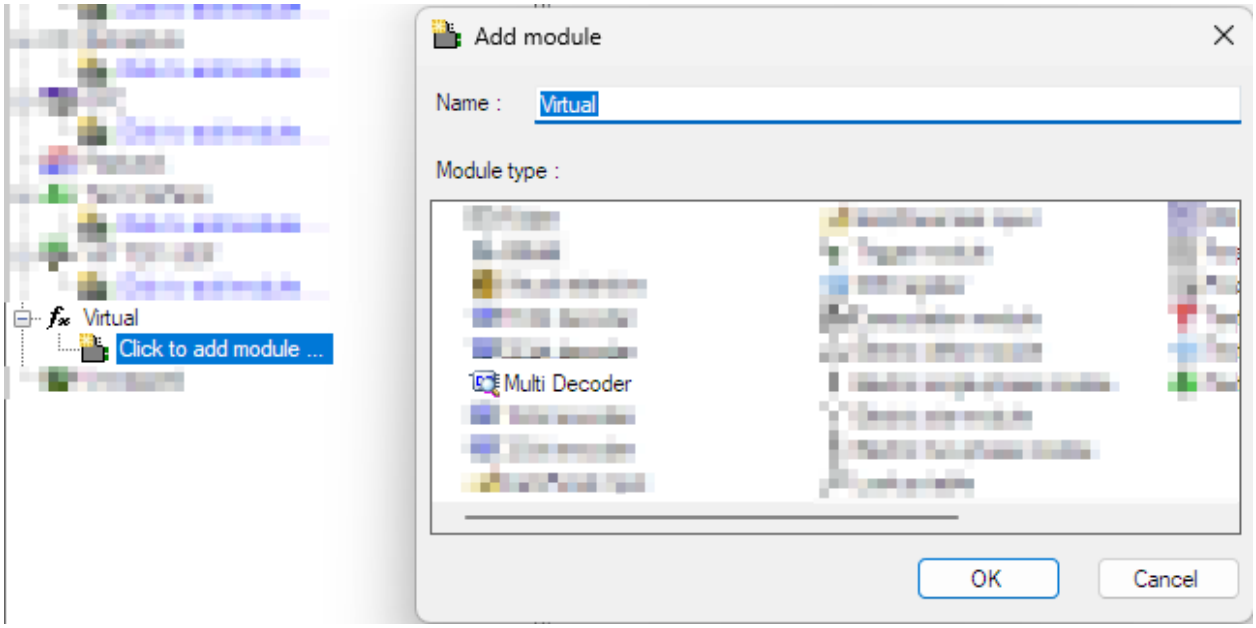
2 Offline trend graph – Legend

Previously the offline trend graph only supported file signals as the legend text signal. Starting with this version, ibaPDA and ibaHD signals can be selected as well.



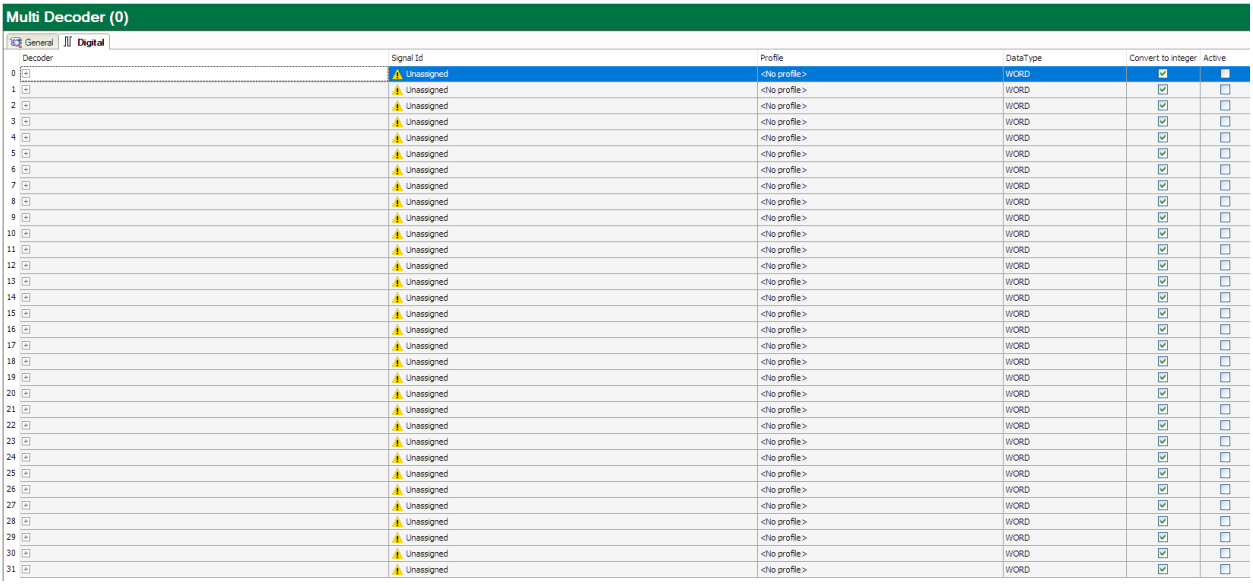
3 Multi decoder module

The new Multi decoder module allows defining multiple decoders in a single module. This greatly improves configuring decoders and provides a more user-friendly experience.



3.1 Settings

The individual decoders are listed in the *Digital* tab of the module settings.
Each row represents a decoder. By default, 32 decoders are available. You can change the number of decoders in the property grid on the *General* tab.

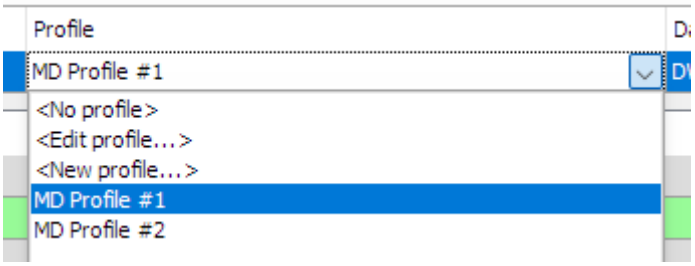


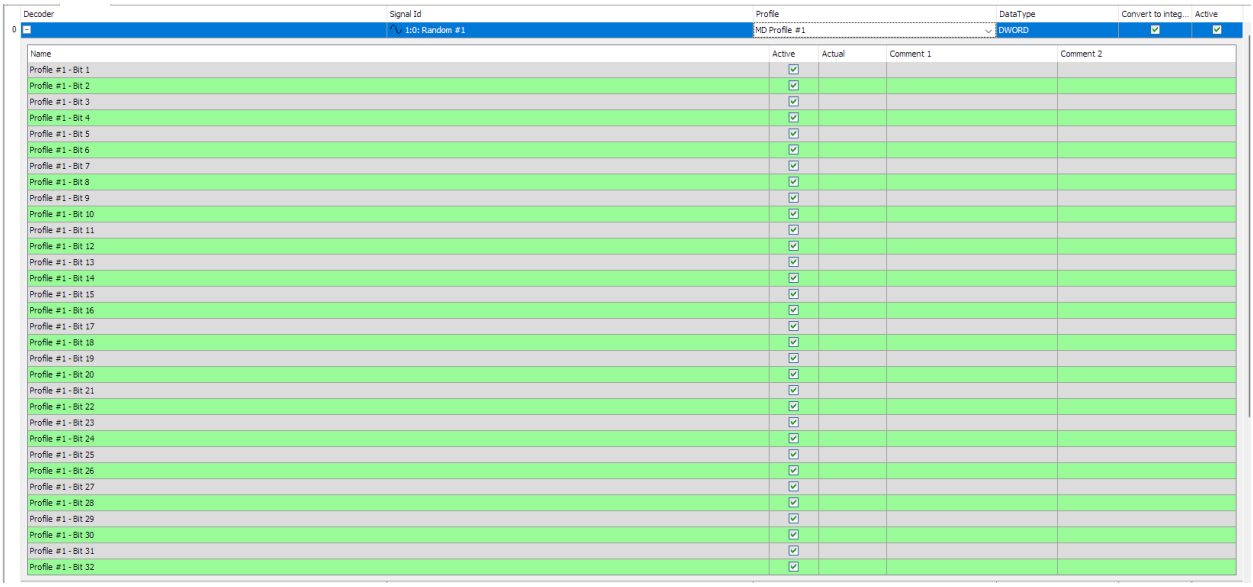
Signal Id

The signal of a decoder can be set using the *Signal Id* column. It is set to *Unassigned* by default. *Unassigned* signals are deactivated and cannot be activated.

Profile

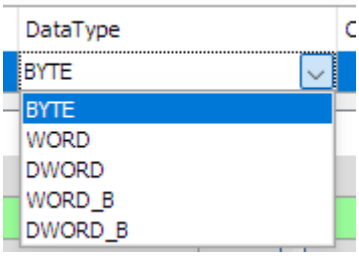
The *Profile* column defines the profile that is applied to the decoder. A profile contains predefined names and comments for the decoder signals (see Profile). The profile is applied when selecting it via the dropdown.





Datatype

The datatype defines the endianness and the number of bits available in the current decoder. Selecting a *Signal Id* automatically changes the *datatype* to the datatype of the selected signal.



BYTE

BYTE provides 8 bits.

WORD

WORD provides 16 bits. *WORD*_B is the big-endian variant.

DWORD

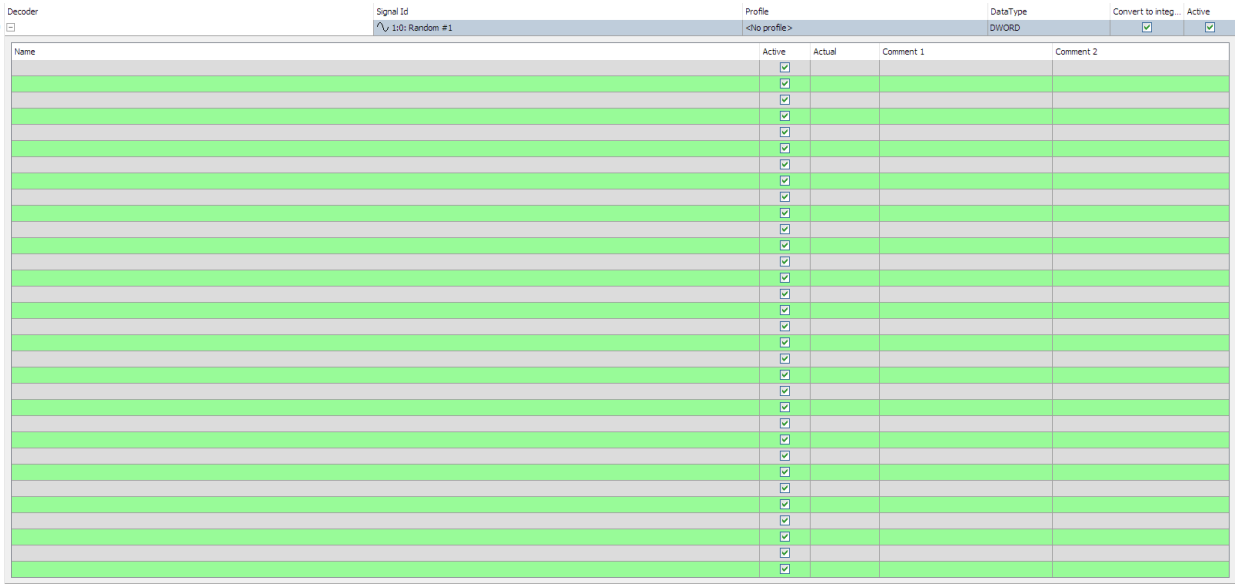
DWORD provides 32 bits. *DWORD*_B is the big-endian variant.

Convert to integer
Using this option, the selected signal is converted to an integer before it's processed. If this is unchecked then

Active
A decoder can be activated or deactivated using the *Active* column.
The decoder cannot be activated if the *Signal Id* is set to *Unassigned*.

3.2 Decoder

Expanding a decoder displays the contained bits.
For each bit, the *Name*, *Comment1*, *Comment2* and an *Active* flag can be set.

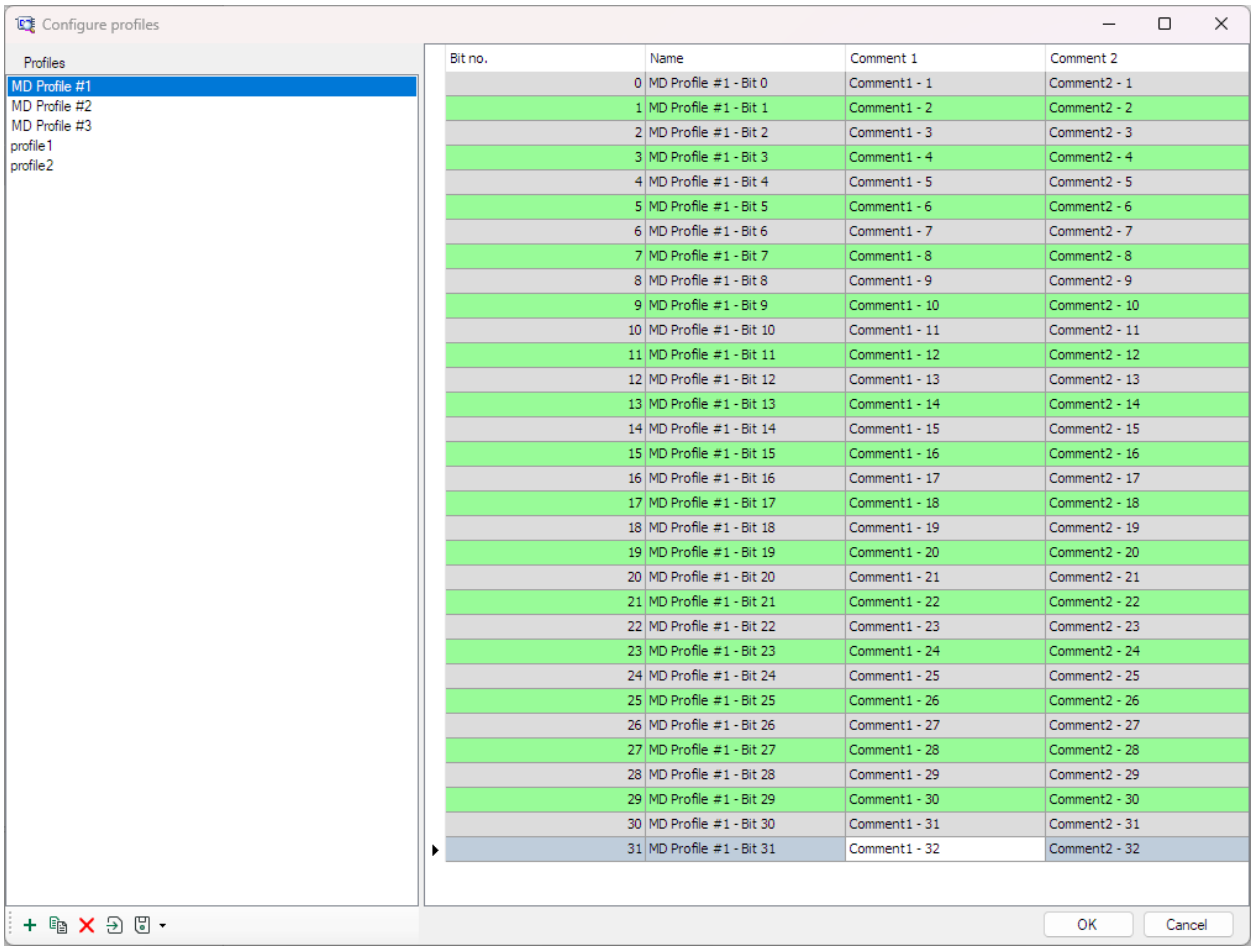


Selecting a *Profile* automatically fills the *Name*, *Comment1* and *Comment2*. The bits are read-only and can only be changed if no profile is selected.

3.3 Profiles

A Multi decoder profile contains predefined values that can be assigned to the bits of a decoder.

Profiles can be managed using the *Configure profiles* link in the *general* tab, or using *the <Edit profile…>* and *<New profile…>* option in the *Profiles* column in the grid.



Bit no.	Name	Comment 1	Comment 2
0	MD Profile #1 - Bit 0	Comment1 - 1	Comment2 - 1
1	MD Profile #1 - Bit 1	Comment1 - 2	Comment2 - 2
2	MD Profile #1 - Bit 2	Comment1 - 3	Comment2 - 3
3	MD Profile #1 - Bit 3	Comment1 - 4	Comment2 - 4
4	MD Profile #1 - Bit 4	Comment1 - 5	Comment2 - 5
5	MD Profile #1 - Bit 5	Comment1 - 6	Comment2 - 6
6	MD Profile #1 - Bit 6	Comment1 - 7	Comment2 - 7
7	MD Profile #1 - Bit 7	Comment1 - 8	Comment2 - 8
8	MD Profile #1 - Bit 8	Comment1 - 9	Comment2 - 9
9	MD Profile #1 - Bit 9	Comment1 - 10	Comment2 - 10
10	MD Profile #1 - Bit 10	Comment1 - 11	Comment2 - 11
11	MD Profile #1 - Bit 11	Comment1 - 12	Comment2 - 12
12	MD Profile #1 - Bit 12	Comment1 - 13	Comment2 - 13
13	MD Profile #1 - Bit 13	Comment1 - 14	Comment2 - 14
14	MD Profile #1 - Bit 14	Comment1 - 15	Comment2 - 15
15	MD Profile #1 - Bit 15	Comment1 - 16	Comment2 - 16
16	MD Profile #1 - Bit 16	Comment1 - 17	Comment2 - 17
17	MD Profile #1 - Bit 17	Comment1 - 18	Comment2 - 18
18	MD Profile #1 - Bit 18	Comment1 - 19	Comment2 - 19
19	MD Profile #1 - Bit 19	Comment1 - 20	Comment2 - 20
20	MD Profile #1 - Bit 20	Comment1 - 21	Comment2 - 21
21	MD Profile #1 - Bit 21	Comment1 - 22	Comment2 - 22
22	MD Profile #1 - Bit 22	Comment1 - 23	Comment2 - 23
23	MD Profile #1 - Bit 23	Comment1 - 24	Comment2 - 24
24	MD Profile #1 - Bit 24	Comment1 - 25	Comment2 - 25
25	MD Profile #1 - Bit 25	Comment1 - 26	Comment2 - 26
26	MD Profile #1 - Bit 26	Comment1 - 27	Comment2 - 27
27	MD Profile #1 - Bit 27	Comment1 - 28	Comment2 - 28
28	MD Profile #1 - Bit 28	Comment1 - 29	Comment2 - 29
29	MD Profile #1 - Bit 29	Comment1 - 30	Comment2 - 30
30	MD Profile #1 - Bit 30	Comment1 - 31	Comment2 - 31
31	MD Profile #1 - Bit 31	Comment1 - 32	Comment2 - 32

Each profile contains 32 bits. If the selected decoder contains less than 32 bits, only the bits up to the number of bits in the profile are applied.

For each bit, the *Name*, *Comment 1* and *Comment 2* can be set. *Bit no.* is only a visual help and doesn't provide any functionality.

Changes in a profile (e.g renaming a bit) are also adapated to the related decoders.

If a profile is deleted, the profile of the related decoders is set to *<No profile…>*.

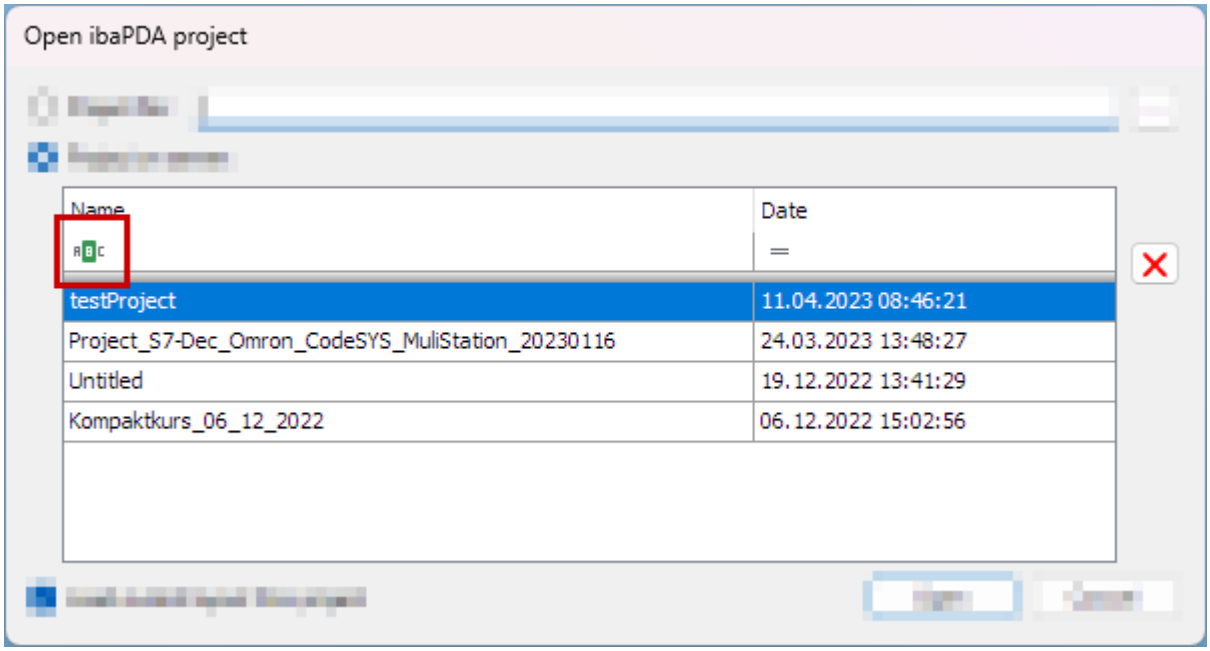
4 Advanced filter in grids

From this version, most grids provide a powerful and advanced filter.

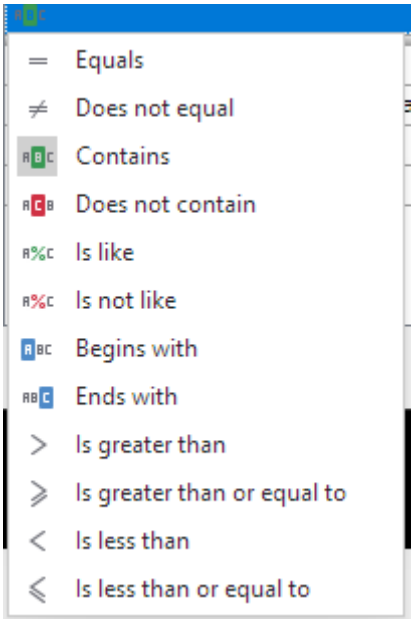
4.1 Simple filter

A simplified version of the filter can be configured in the designated filter row, using the available methods and the filter pattern.

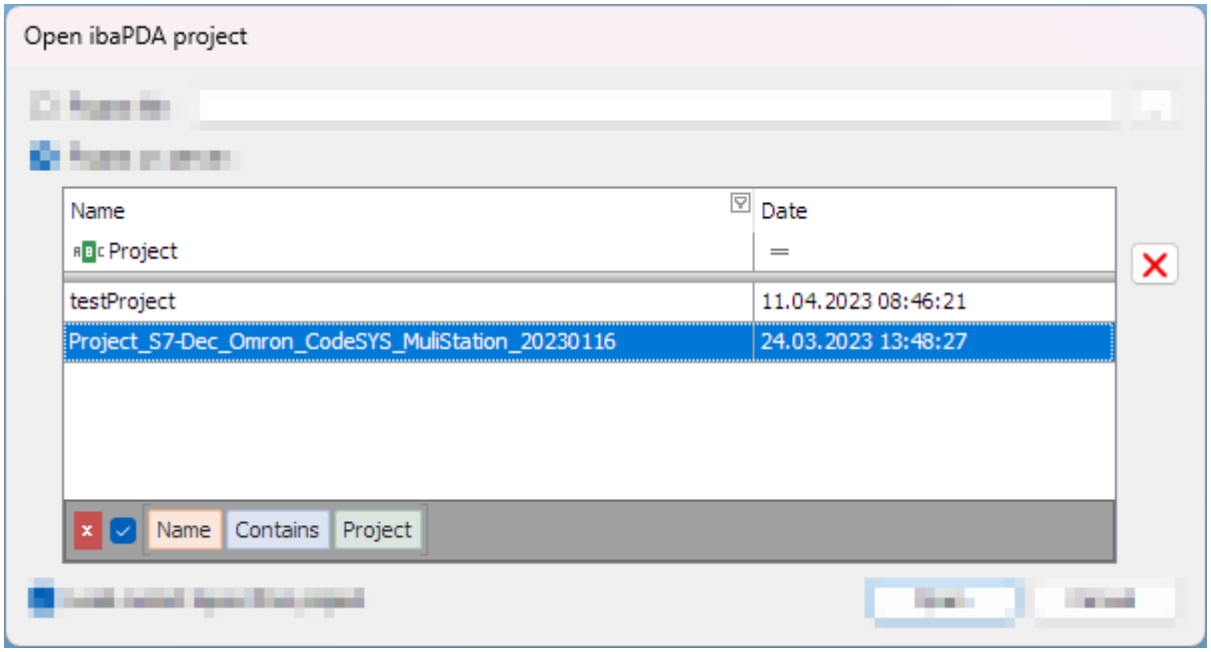
The designated filter row can be identified by an icon representing the currently active filter method.



In this example, the default method is selected for the *Name* column. Clicking on the icon displays a list of available methods. Each method leads to a different filter behavior.



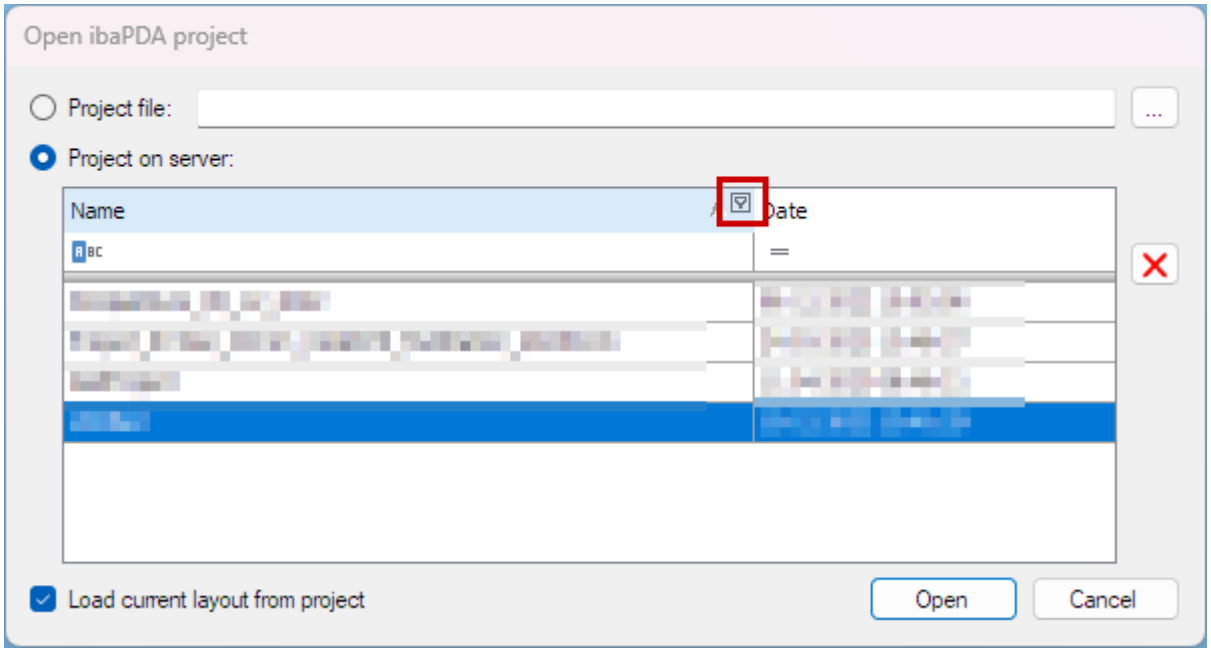
With *Contains* selected as the filtering method and *Project* as the pattern, the available rows in the grid are reduced to those that match the configured filter.

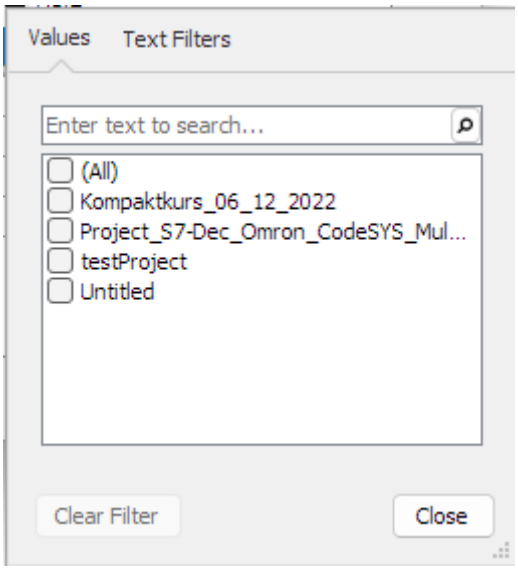


The currently active filter is also visualized at the bottom of the grid in a human-readable way, where the first fields represents the column (*Name*), the second the filter method (*Contains*) and the third the filter pattern (*Project*).

4.2 Advanced filter

The advanced filter can be activated by moving the mouse cursor over a column header and clicking the filter icon.



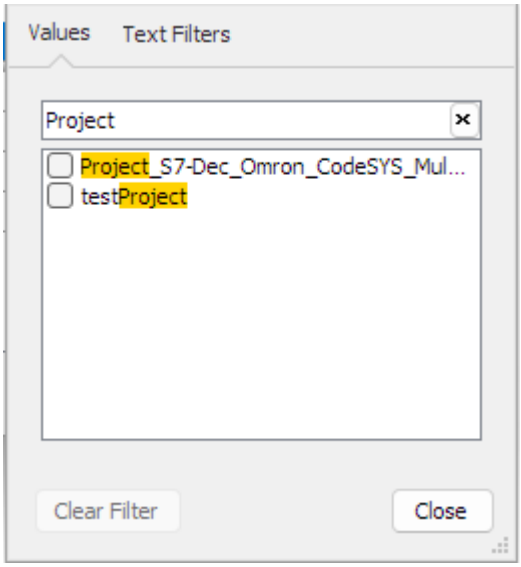


The advanced filter offers two tabs. *Values* for simply searching for a value and *Advanced Filters* for configuring a filter based on the value type.

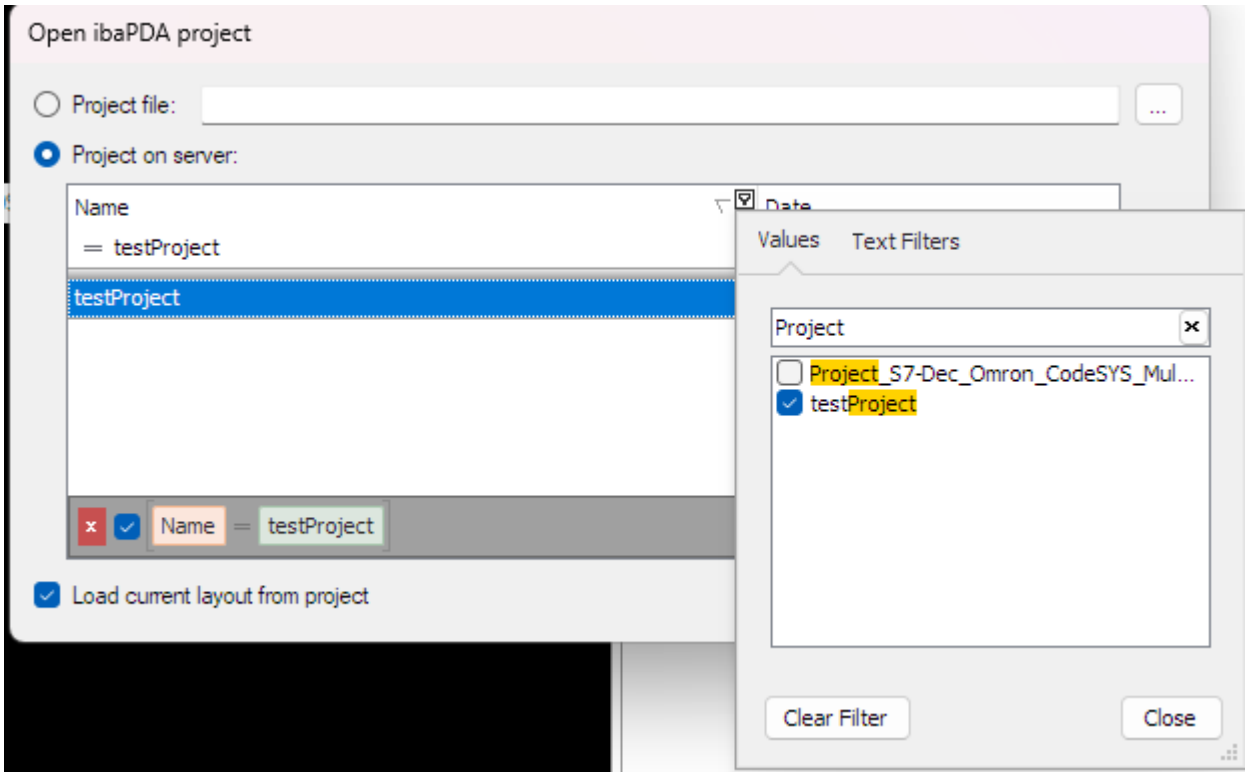
In this example the value is a text, so a *Text Filters* tab is available. For dates, a *Date Filters* tab would be visible instead, and for numeric values, the *Numeric Filters* tab (see Advanced filters).

4.2.1 Values

The *Values* tab contains a list with all values of the selected column.
Above the list, a search pattern can be entered. Rows that don't match the pattern are removed from the list and the pattern is highlighted in each matching row.



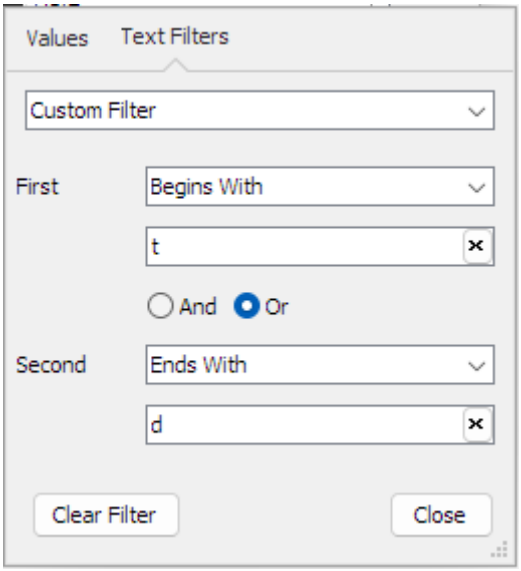
The selected matches are automatically shown in the grid.

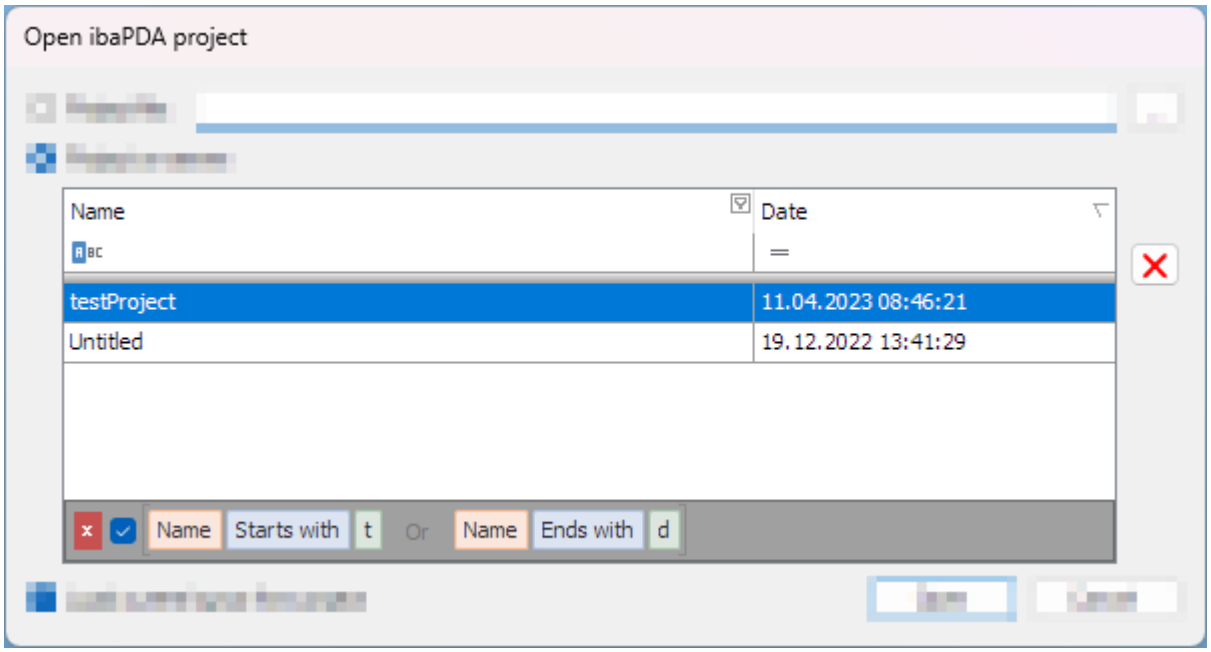


4.2.2 Advanced filters

The content of the advanced filters depends on the type of column value. For example, a column with numeric values has different filter options than one with date values.

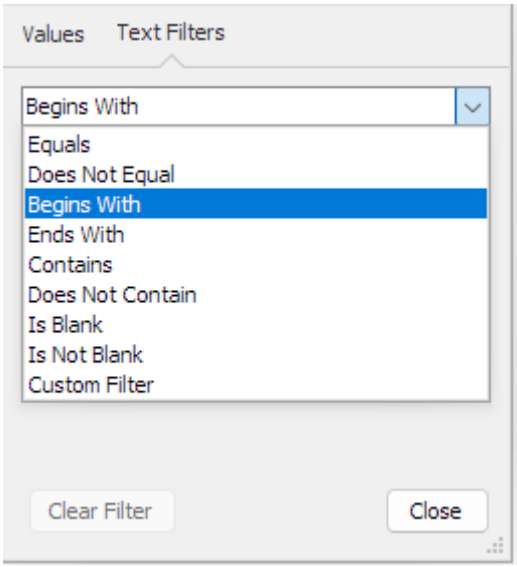
However, each filter type also provides a *Custom Filter* option.
Custom Filter can be used to create additional filters when the existing ones are not enough.





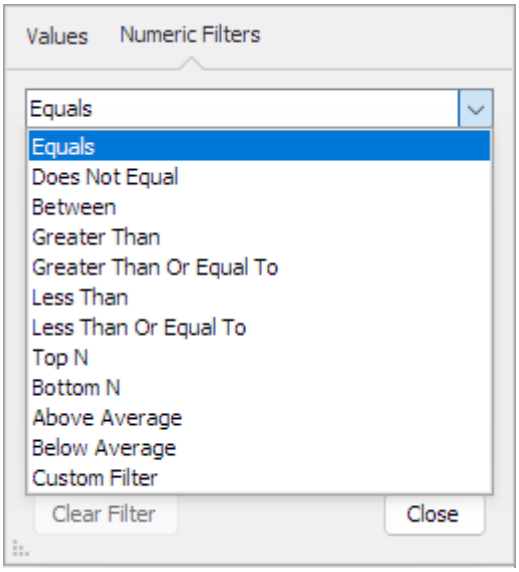
4.2.2.1 Text filters

For columns with text, *Text Filters* can be used.



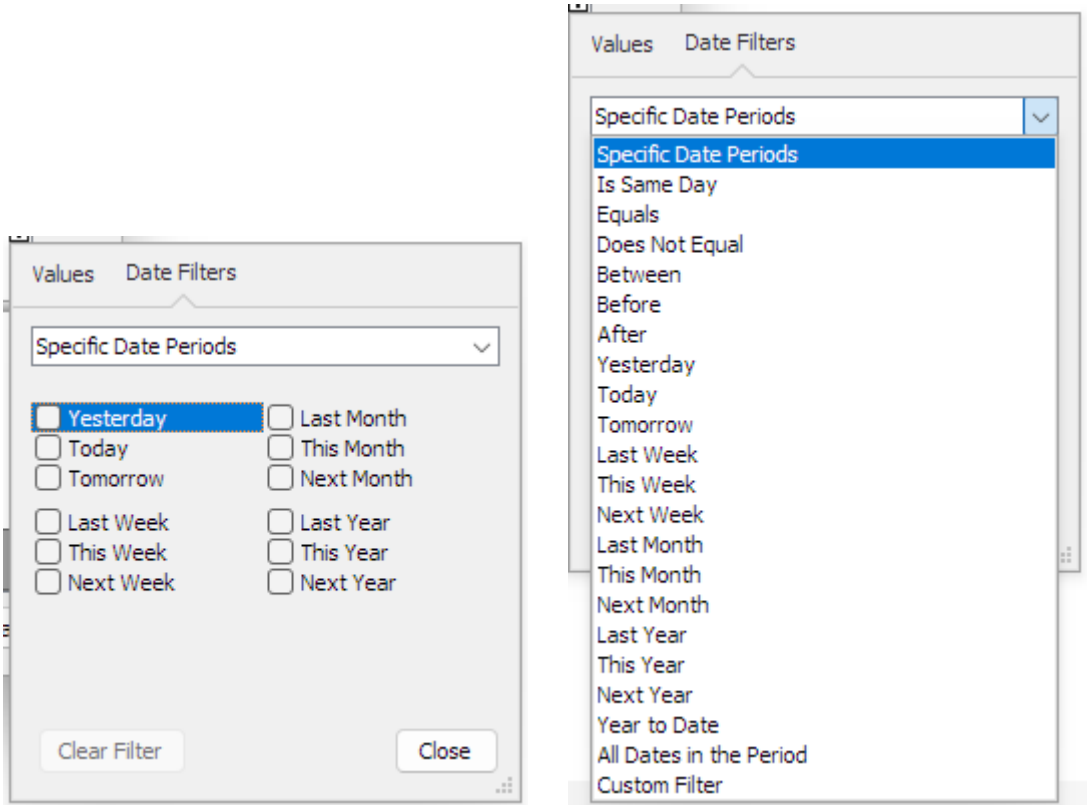
4.2.2.2 Numeric filters

Numeric Filters are only available for numeric columns.



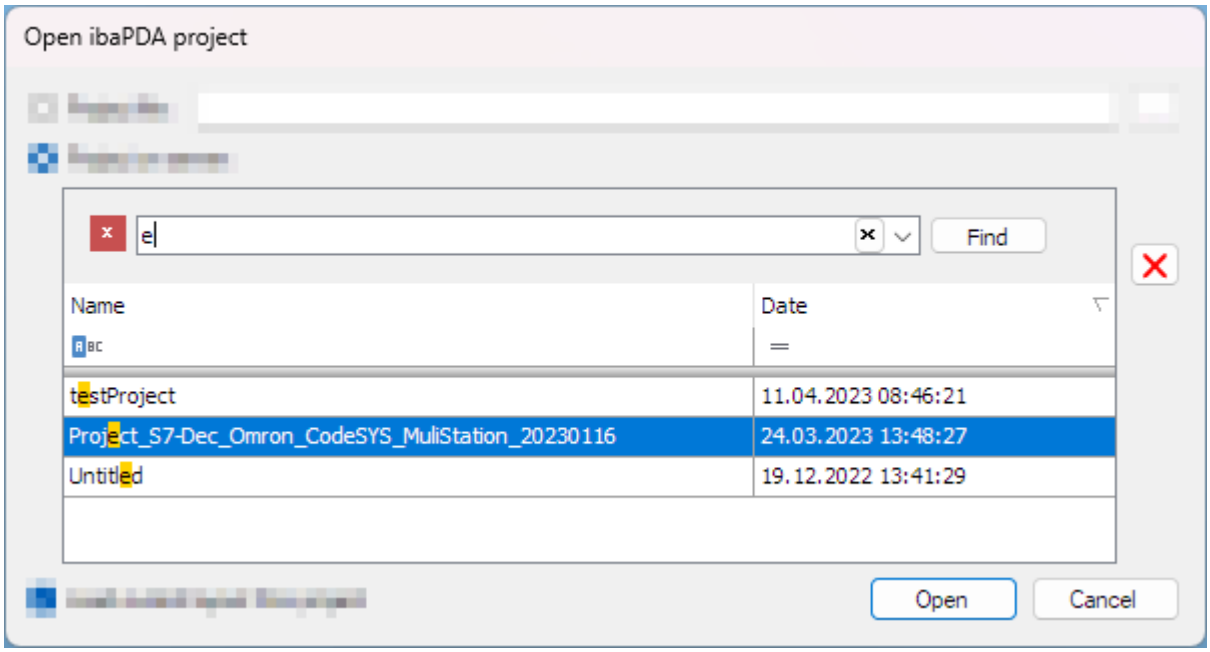
4.2.2.3 Date filters

Columns containing date values can use the *Date Filters* tab.



4.3 Search

A normal search is available via CTRL + F.



5 MQTT Sparkplug B data store

The new *MQTT Sparkplug B data* store allows you to write Sparkplug B formatted data to an MQTT server.

5.1 Licensing

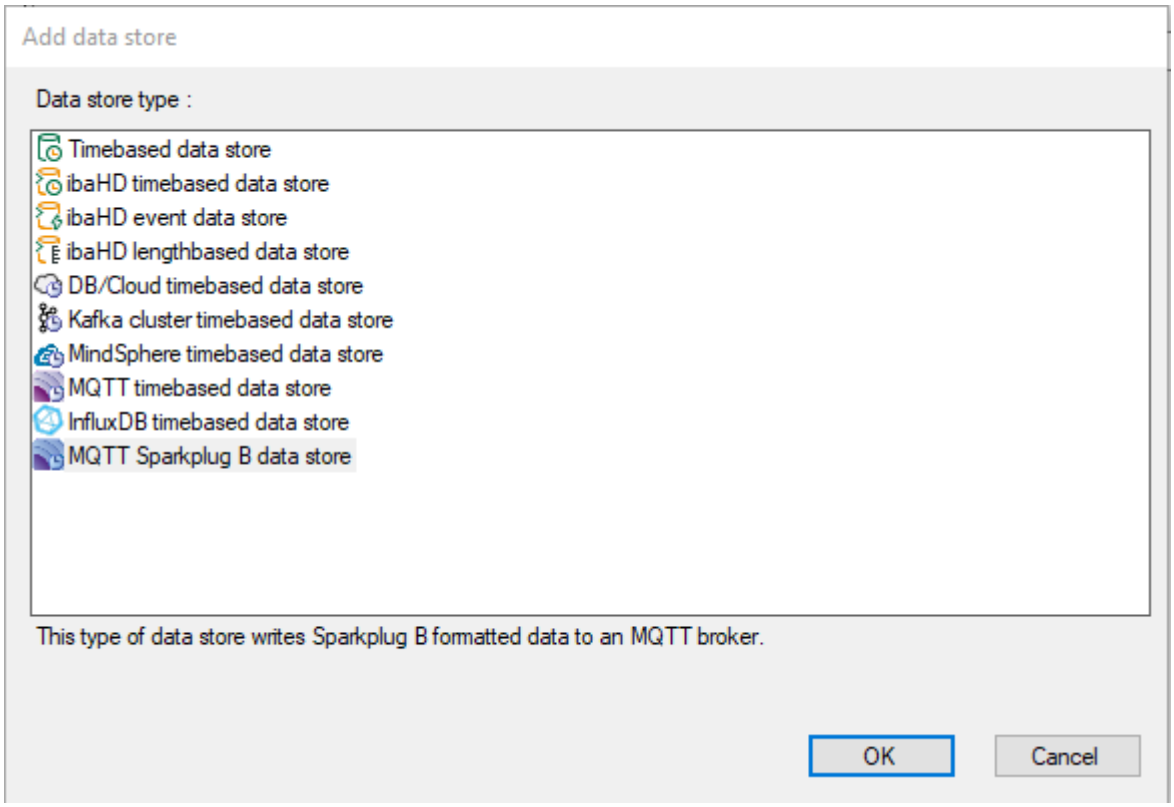
The licensing uses the existing MQTT data store licenses.

- *ibaPDA-Data-Store-MQTT-16* (30.671000), allows you to write 16 signals in total
- *ibaPDA-Data-Store-MQTT-64* (30.671001), allows you to write 64 signals in total
- *ibaPDA-Data-Store-MQTT-256* (30.671002), allows you to write 256 signals in total
- *ibaPDA-Data-Store-MQTT-1024* (30.671003), allows you to write 1024 signals in total

Licensing is possible on MARX and WIBU dongle systems.

5.2 Configuration in ibaPDA

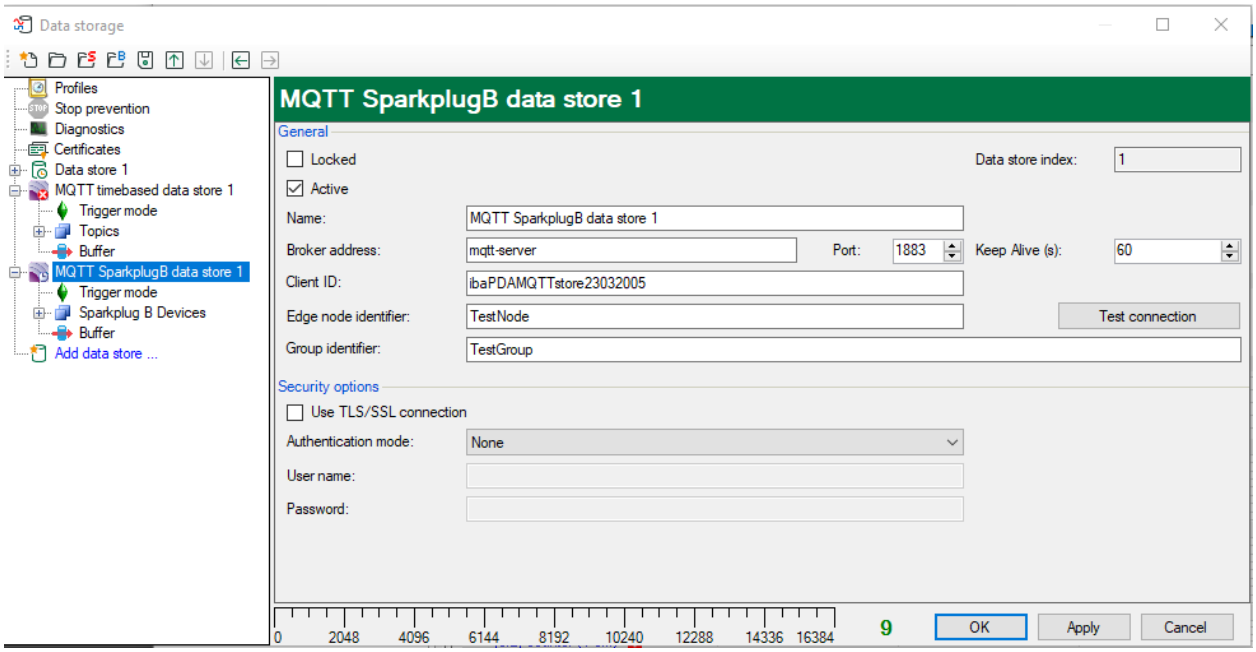
To add a *MQTT Sparkplug B data store* click on *Add data store…* in the Data storage configuration, select the *MQTT Sparkplug B data store*, and click <OK>.



5.2.1 Sparkplug B specific configuration

The basic connection configuration is like the MQTT timebased data store, except for options predefined by Sparkplug B, like the *Last Will* or *Connect* messages.

ibaPDA currently supports the *EdgeNode Rebirth* command, when a valid Sparkplug B message containing the metric “Node Control/Rebirth” is written into the MQTT topic “spBv1.0/<Group identifier>/NCMD/<Edge node identifier>. When a client application issues this command, ibaPDA will generate new NBIRTH/DBIRTH messages. IbaPDA uses the Sparkplug B protobuf definition found in https://github.com/eclipse/tahu/tree/master/sparkplug_b/



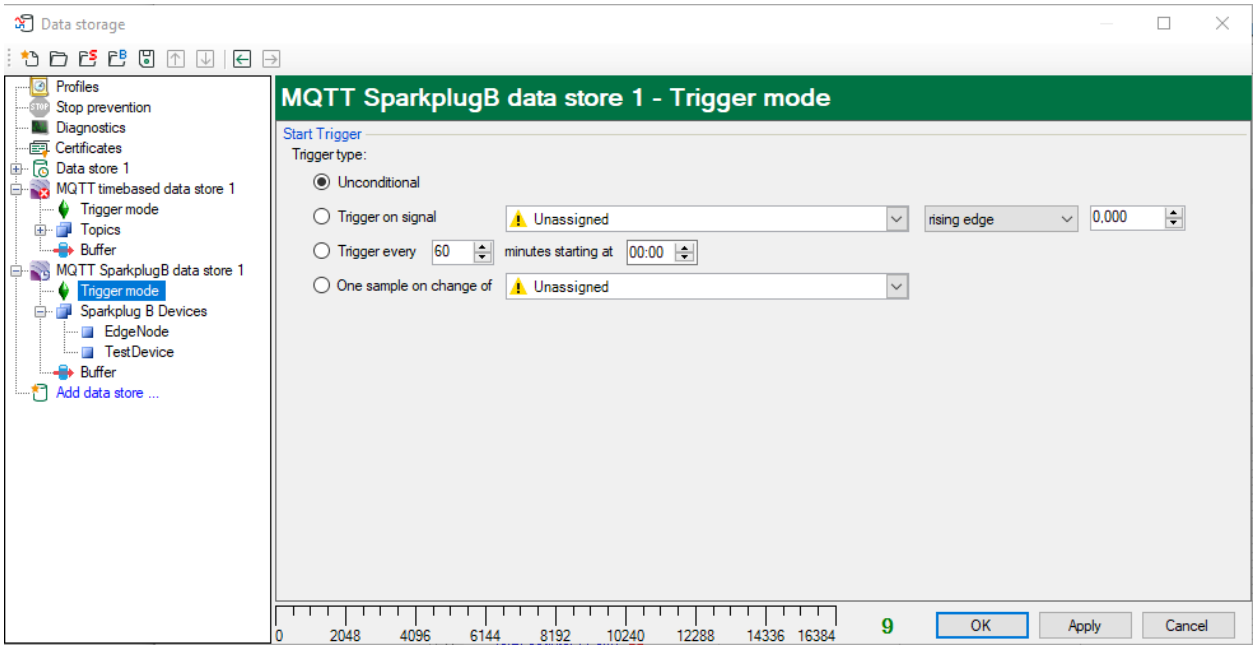
- **Locked**: data store can be locked in order to prevent an accidental or unauthorized change of settings.
- **Active**: A data store must be enabled in order to work. However, you can configure various data stores and disable data stores that are not required.
- **Data store index**: Unique index of all existing MQTT data stores. You need to reference this index e.g. in the virtual function *DataStoreInfoMqtt()* for generating diagnostic data for a specific MQTT data store.
- **Name**: You can enter a name for the data store here. Data store names have to be unique.
- **Server address**: The IP address or hostname of the MQTT server
- **Port**: Port to use for the connection. The standard port is 1883 for unsecured communication.

  Click on the button <Test connection> to verify that ibaPDA can establish a basic connection to the MQTT server without generating a *Last Will* or *Connect* message.
- **Client ID**: Unique client ID used by the MQTT server to track connectivity, and to publish the *Last Will* message
- **Edge node identifier**: ibaPDA acts as an Edge node, as defined in the Sparkplug B specification. The required edge node name has to be entered here.
- **Group identifier**: The group identifier according to the Sparkplug B specification.

  The resulting data path in MQTT will be “spBv1.0/<Group identifier>/<message type>/<Edge node identifier>/
- **Use TLS/SSL connection**: To secure the communication with the MQTT server, either a login using user name/password or a certificate can be used. Select the appropriate setting here, and change the port according to your MQTT server configuration.

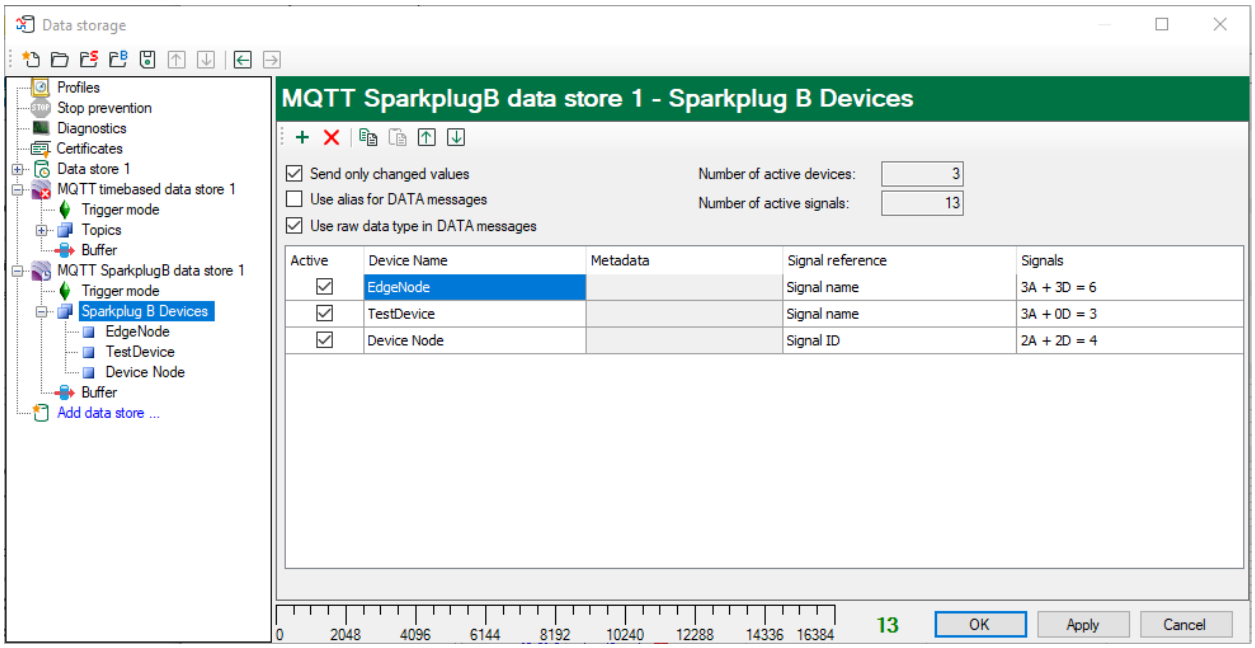
5.2.2 Trigger Mode

In the node *Trigger Mode* you adjust if the data is written continuously, or only on certain conditions. The available options are the same as for other data store types.



5.2.3 Sparkplug B Devices

In the *Sparkplug B Devices* node, the signals transferred in the *DATA* messages can be selected for the basic edge node and the additional devices. The predefined *EdgeNode* must be always present and cannot be deleted or deactivated. As usual, the *Signals* selection uses the defined profiles of the *Time* profile type to achieve subsampling or min/max/average signal aggregation.



- **Send only changed values**: Each NDATA/DDATA message contains only the signals that have changed.

- **Use alias for DATA messages**: Instead of the *signal reference*, the alias number is used in NDATA/DDATA messages. The NBIRTH/DBIRTH messages contain the signal reference together with the alias.
- **Use raw data type in DATA messages**: Usually ibaPDA writes signal values using float or double value formatting. By checking this option, the original value format is used (e.g. INT16 or DWORD). This can be useful for bitmapped values. But by using this option, the signal values will disregard any scale/offset option selected.

The complete signal definition set, together with any metadata selected, is contained in the NBIRTH/DBIRTH messages. The NDATA/DDATA messages will not contain the metadata or additional information about the edge node or the device. As signal reference can be used:

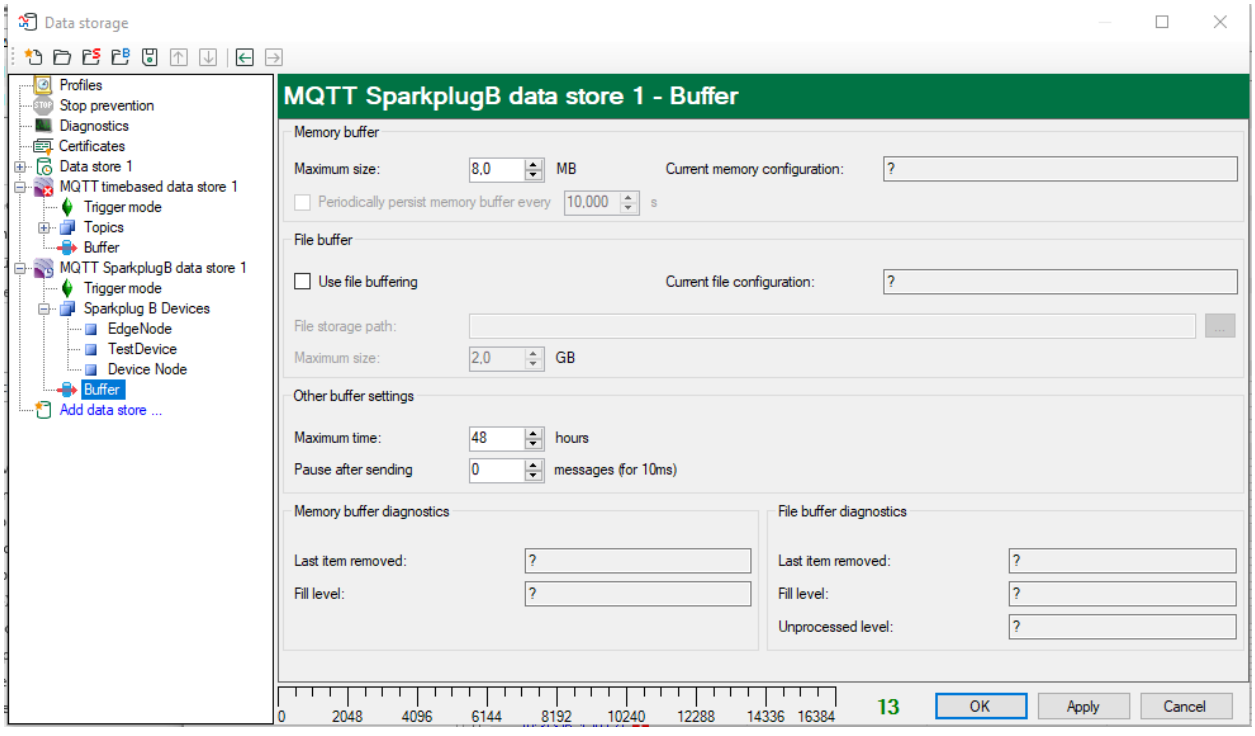
- Signal name
- Signal ID (e.g. [1:3])
- Signal comment 1
- Signal comment 2

Metadata selection includes:

- Signal name
- Signal ID
- Signal comment 1
- Signal comment 2
- Unit

5.2.4 Buffer

Like for other data store types used for streaming data to external systems a buffer functionality is available to overcome connection losses and to balance the network overloads. The functionality is identical to other data store types.

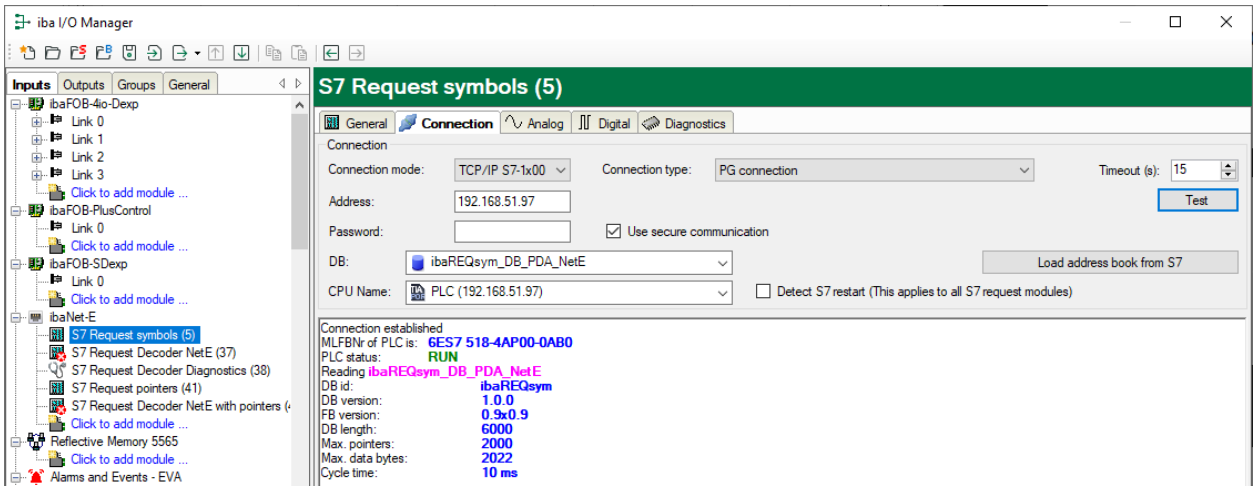


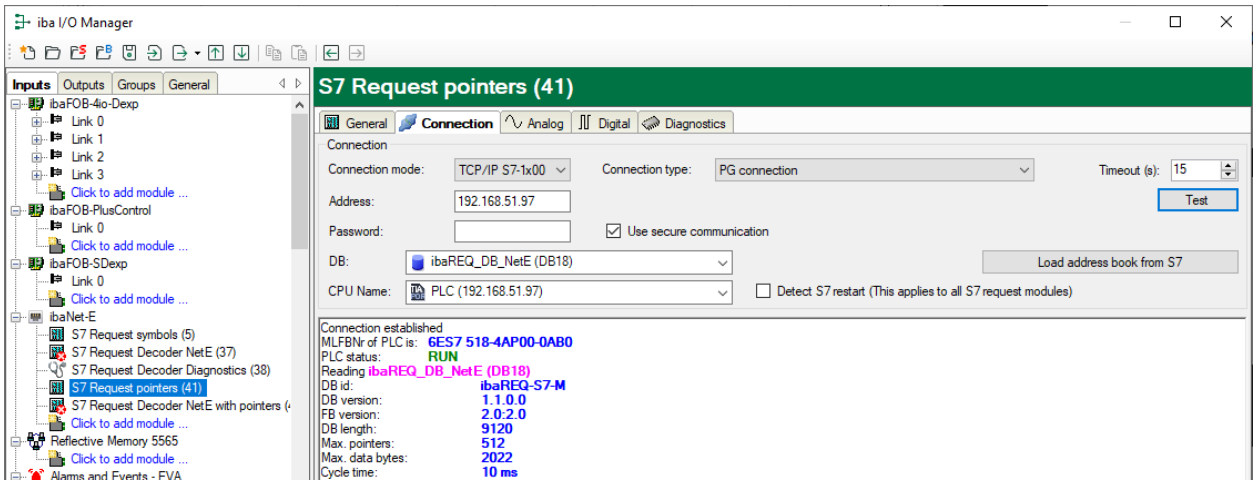
6 S7 request

6.1 Support for optimized data blocks

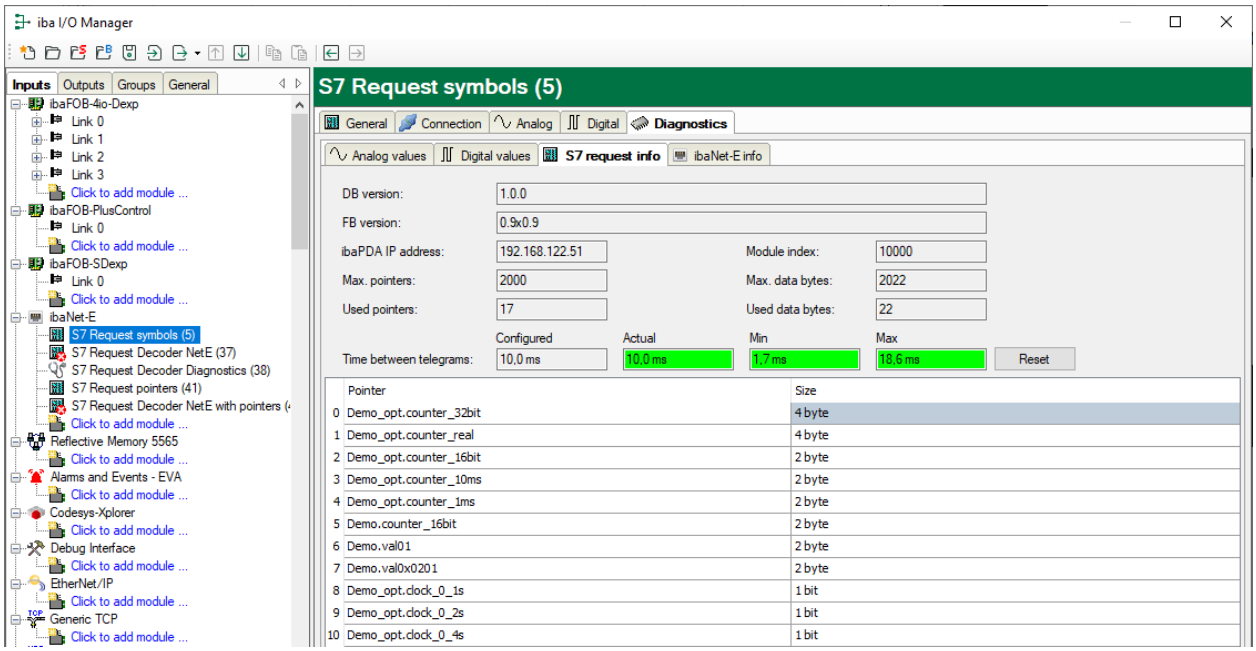
In S7-1500 PLCs you can use optimized data blocks. Signals inside these data blocks can only be addressed through their symbolic names and not by address or operand. In firmware version 3 of the S7-1500 PLCs Siemens has added system calls to read the values of members of such optimized data blocks using their names. Iba has created new S7 request function blocks that support these optimized data blocks. IbaPDA supports both request functions blocks: those that support optimized data blocks and those that do not.

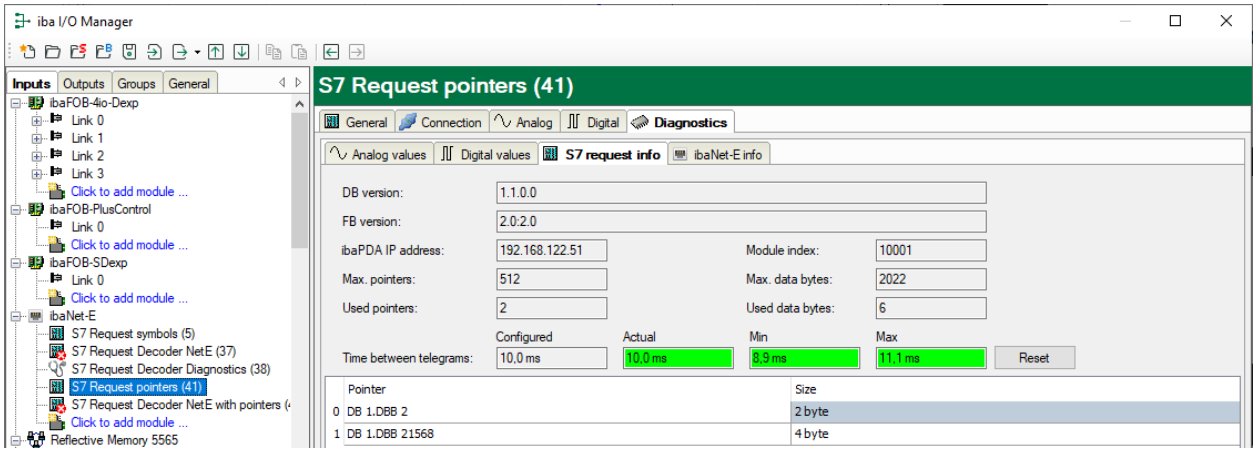
In ibaPDA you can click the *Test* button to connect to the S7 and read out some information from the request function block. IbaPDA uses the DB id to determine if the request function block supports optimized data blocks or not. If the DB id is *ibaREQsym* then optimized data blocks are supported. The S7 symbol browser will allow optimized data blocks if the connected request function block supports them.





On the S7 request info tab on the diagnostics tab you will see symbols or S7 operands in the table depending on which request function block is connected.





6.2 Support for ibaNet-E

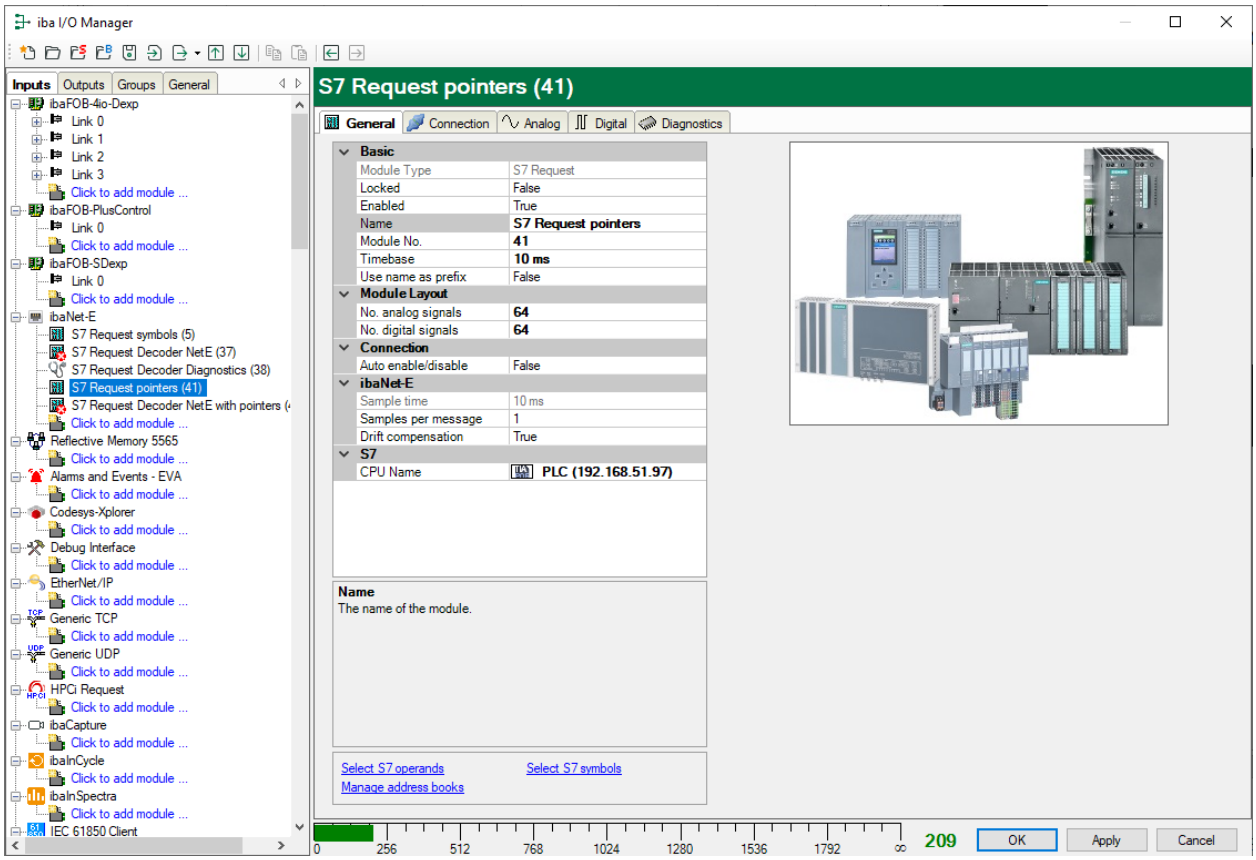
The ibaNet-E protocol is a network protocol developed by iba to transfer data between measurement devices and ibaPDA. It has 2 variants:

- UDP-based
- Ethernet-based

Iba has created S7 request function blocks that can send the requested data using the UDP-based ibaNet-E protocol. The ibaNet-E protocol has the following advantages compared to UDP:

- Better handling of jitter. There are a lot less samples skipped or duplicated compared to UDP.
- Multiple samples can be transferred in 1 message. This means that the sending cycle can be slower than the measurement cycle. This reduces the load on the communications processor.

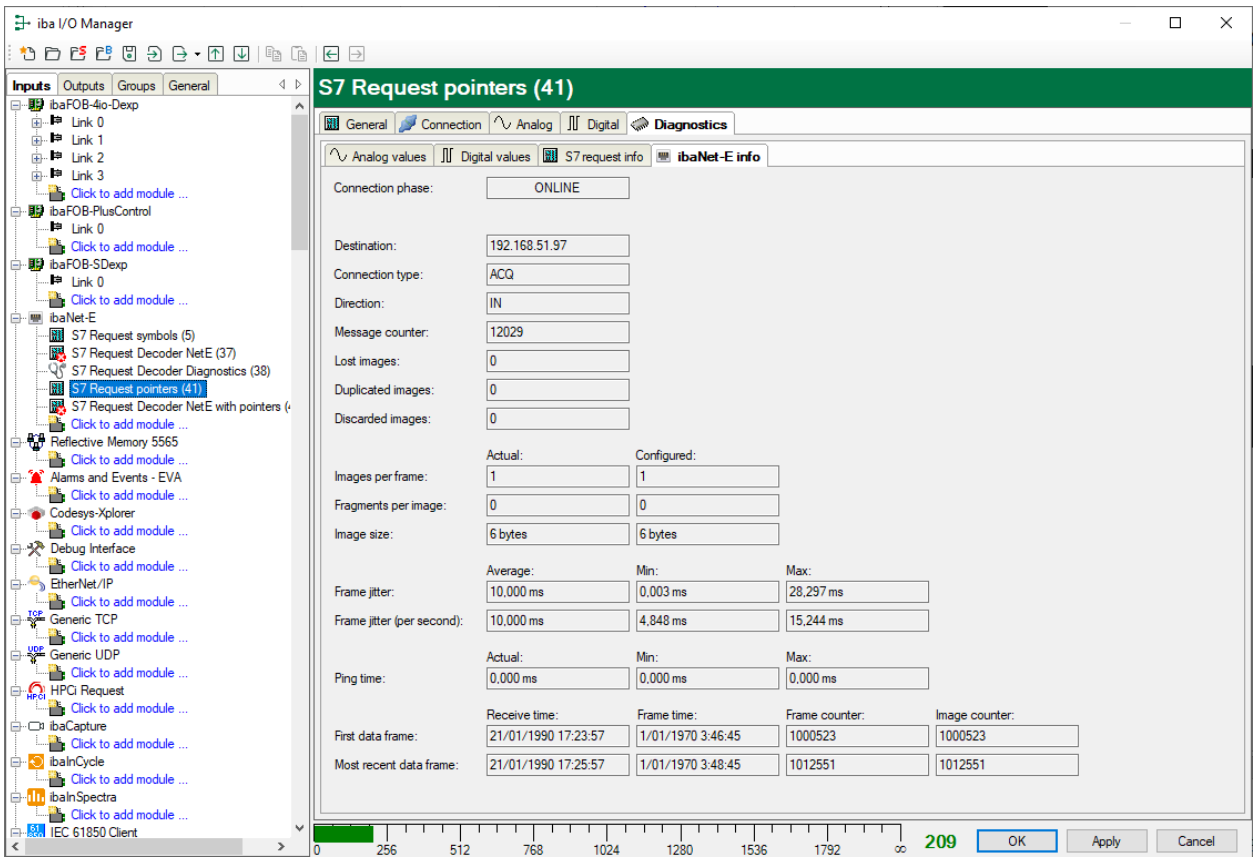
On the ibaNet-E interface in the I/O manager you can now add *S7 request* and *S7 request decoder* modules. The configuration is almost the same as other S7 request modules on ibaBM-DP, ibaBM-PN and S7 TCP/UDP.



On the *General* tab there is an ibaNet-E section with the following properties:

- **Sample time**: This read-only property shows how fast the data is sampled on the S7 side. IbaPDA receives this from the S7 request function block when you test the connection.
- **Samples per message**: This property determines how many samples will be sent in 1 ibaNet-E message. By default, this is set to 1. This means that each message contains 1 sample of every requested signal. If you increase this to 2 then each message will contain 2 samples of every requested signal. If the sample time is 10 ms then a new message will be sent every 20 ms. The message will be 2 times as long. An ibaNet-E message can only be 2047 bytes long. If the configured number of samples requires a message larger than the maximum message size then ibaPDA will automatically reduce the number of samples per message.
- **Drift compensation**: The clock of the S7 and the clock of ibaPDA are not synchronized. So, after some time the S7 will have sent either too many or too few samples to ibaPDA. IbaPDA handles this by skipping or duplicating samples. If drift compensation is enabled then a PLL is used in ibaPDA to measure the difference between the sender frequency and the receiver frequency over a longer time. This measured difference is then used to spread the skipping or duplicating of samples over a long time. It takes some time for drift compensation to get a good measurement of the frequency difference. So, right after the start of the acquisition more resampling can be seen then after some time. If drift compensation is disabled then the skipping or duplicating depends more on the jitter

when receiving samples. It is recommended to use drift compensation if the sampling time is 10 ms or faster. For slower sampling times drift compensation can be disabled.



There is also an extra *ibaNet-E info* tab. It contains some diagnostics information about the ibaNet-E protocol.

7 InSpectra Expert

7.1 Multiple preprocessing steps

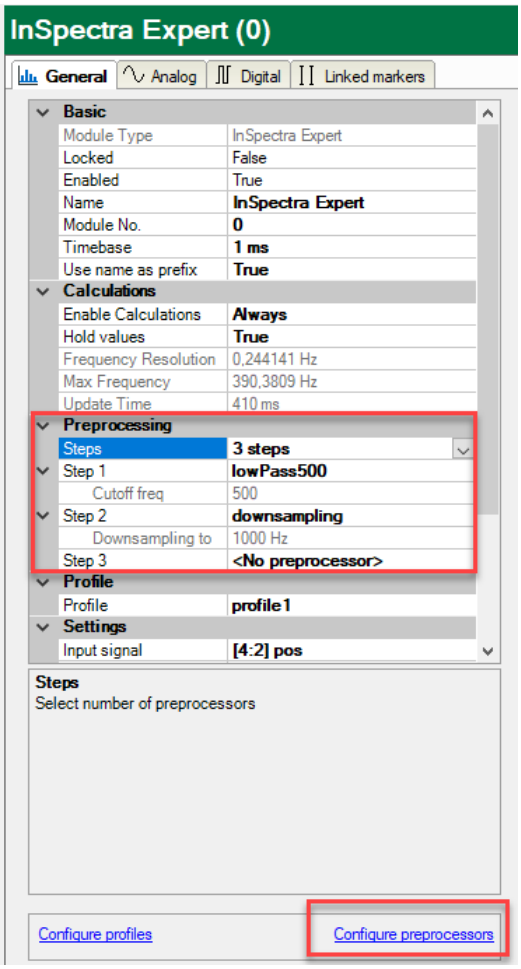
In earlier versions, to prepare the input signal for the FFT calculation, one could use one preprocessor at maximum.

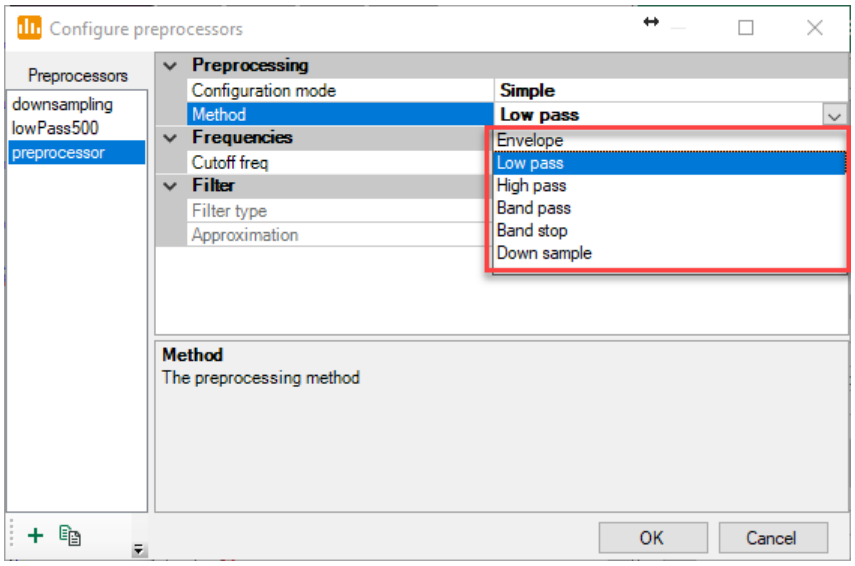
Now, it is possible to combine multiple preprocessors which are applied to the signal sequentially. The following preprocessor types are available:

- Envelope
- Low pass
- High pass
- Band pass
- Band stop
- Down sampling

The number of preprocessing steps can be configured in the *General* tab. One can choose between *None* or *1-2-3-4-5* preprocessors.

The preprocessors can be configured using the *Configure preprocessors* link at the bottom.





7.2 Anti-aliasing filtering for order resampling

To avoid aliasing when using order resampling, the *Anti-aliasing* filtering can be enabled in the spectrum profile. This setting requires configuring the maximum speed as well.

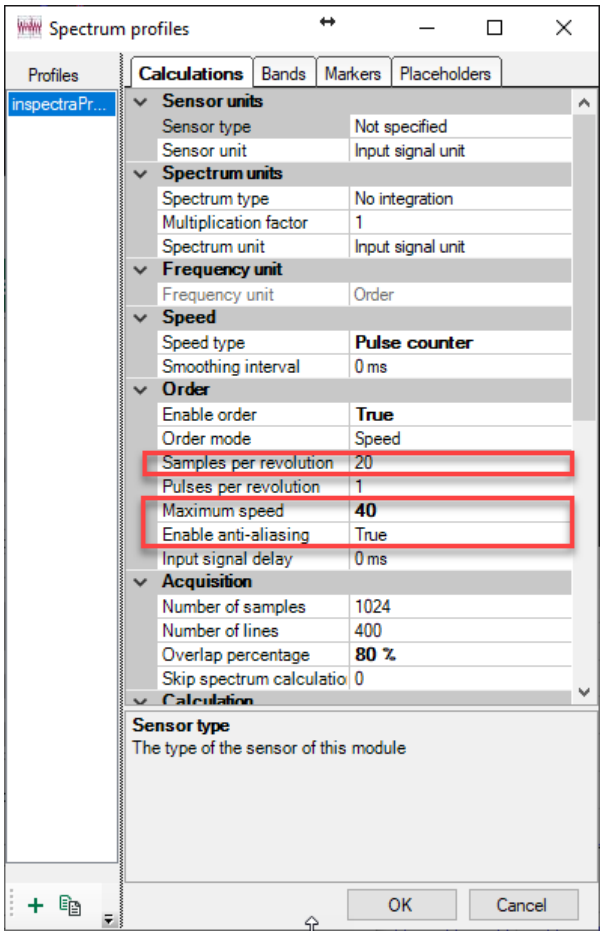
In case the *Anti-aliasing* is enabled, the following steps are performed to the input signal:

1) The custom preprocessors (as configured by the user in the *General* tab) are applied to the input signal first.
2) An automatic low pass filter is applied. It is a Butterworth 4th order filter. The cutoff frequency is the number of samples per revolution configured in the profile, multiplied by the configured maximum speed, divided by 2.
3) The order resampling is done using two times the configured number of samples per revolution.
4) A fixed low pass filter is applied. This is a Butterworth 4th order filter that suppresses all frequencies higher than the Nyquist frequency divided by 2.
5) The result is down sampled with a factor of 2.

If the actual speed is higher than 120% of the configured *Maximum speed*, then the sampling of the input signal is halted temporarily.

If the actual speed is less than 20% of the M*aximum speed*, aliasing effects can still occur.

In new spectrum profiles with order resampling, the *Anti-aliasing* filtering is enabled by default. Do not forget to configure the *Maximum speed* in this case.



If the cutoff frequency of the low pass filter in step 2 is near or higher than the Nyquist frequency, this filter is omitted from the calculation. This filter is also displayed in the *General* tab below the custom preprocessors (if any):