



ibaPDA v7.3.0

New Features

2021-03-22
iba AG

Table of contents

1	General remarks	4
1.1	Post-processing option in data stores	4
1.2	ExecuteCommand function	4
2	ABB-Xplorer Interface	6
2.1	Defining Xplorer module in ibaPDA	6
2.2	Defining access variables in ABB Compact Control Builder	10
3	Hitachi MicroSigma Interface	14
4	Active Directory support in user management	20
5	File buffer for timebased data stores	25
5.1	Configuration options	26
5.1.1	Memory buffer	26
5.1.2	File buffer	27
5.1.3	Other buffer settings	28
5.2	Diagnostics	29
5.2.1	In the buffer configuration control	29
5.2.2	Virtual functions	29
5.2.3	OPC UA Server and SNMP	30
5.3	Updating the data store configuration	31
6	File buffer for SQL output module	32
6.1	Configuration	32
6.2	Diagnostics module	33
7	OPC UA Server module	35
8	Encrypted client-server communication	40
9	Central certificate store	42
9.1	Overview	42
9.2	Managing certificates	42
9.2.1	Toolbar buttons	43
9.2.2	Columns	46
9.3	Using a certificate	47
9.4	Storage and protection of certificates	48
10	Lookup table module	49
10.1	General tab	49
10.2	Signals	50
10.3	Lookup table profiles	51
10.3.1	Profile editor form	52

10.3.2	Editing a profile	53
10.3.3	Knowhow protection	54
11	Computation module	56
12	Trend graph changes.....	60
12.1	User interaction.....	60
12.2	Traversing vector	60
12.3	Zoom between markers	62
13	Display style: Fixed size.....	63
14	ibaQPanel signal tree	65
15	New options for X-Axis in the cycle-view	69
16	InSpectra-Expert new band-indicators: Crest and Phase	71
17	Results as vector for ibalInSpectra- and ibalInCycle-Expert	72

1 General remarks

In ibaPDA v7.3.0 two breaking changes have been implemented affecting

- the post-processing option of a data store
- the function ExecuteCommand in virtual modules.

In case you are using these functions in your configuration and you are updating from a version < v7.3.0 user interaction is required before you can start the acquisition again.

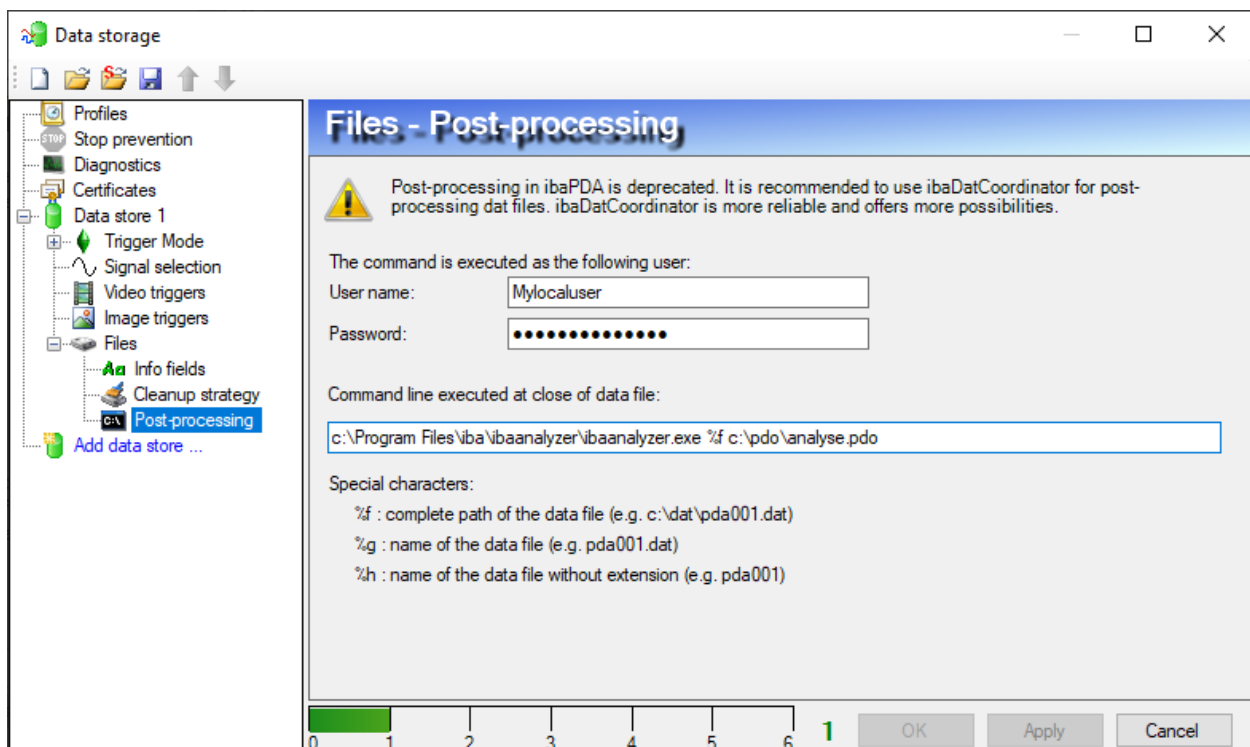
If you need support regarding this contact your regional iba representative listed on our website (<https://www.iba-ag.com/en/contact/>).

1.1 Post-processing option in data stores

The post-processing option is a legacy function since v7.3.0 and it is not available any more for newly configured data stores. It is recommended to use ibaDatCoordinator for post-processing data files. ibaDatCoordinator is more reliable and offers a lot more possibilities.

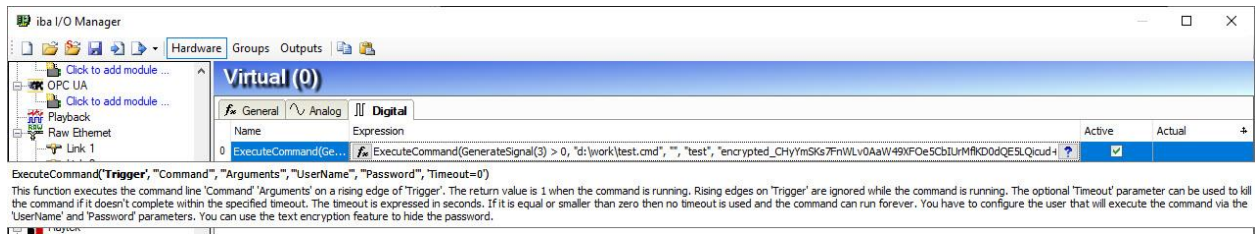
In case you have configured a post-processing command in your data store configuration before updating to v7.3.0 you can continue to use it. However, the post-processing command is not executed anymore with the user the ibaPDA service is running on. Instead, it is required to configure a dedicated user for execution. Unless you have configured a user the data store configuration is invalid and the data store will not work.

Enter the user credentials in the post-processing configuration of your data store:

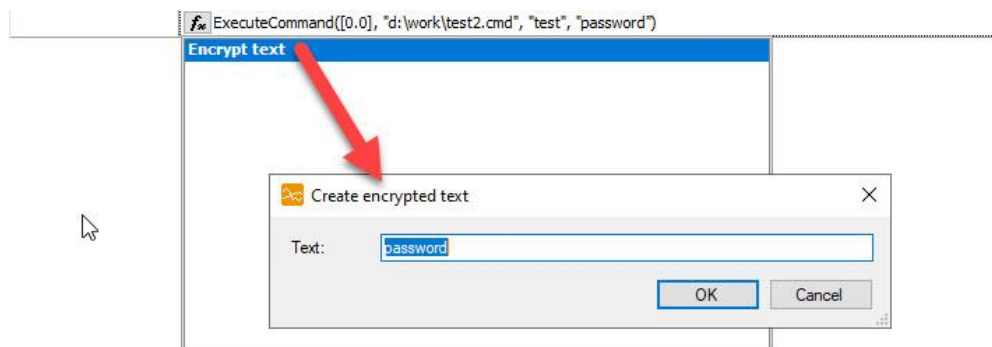


1.2 ExecuteCommand function

The function ExecuteCommand has 2 additional parameters: user name and password. In previous ibaPDA versions the command was executed in the context of the user that was running the service. This was usually LocalSystem. Now this is no longer allowed. You now have to configure a dedicated user for execution.



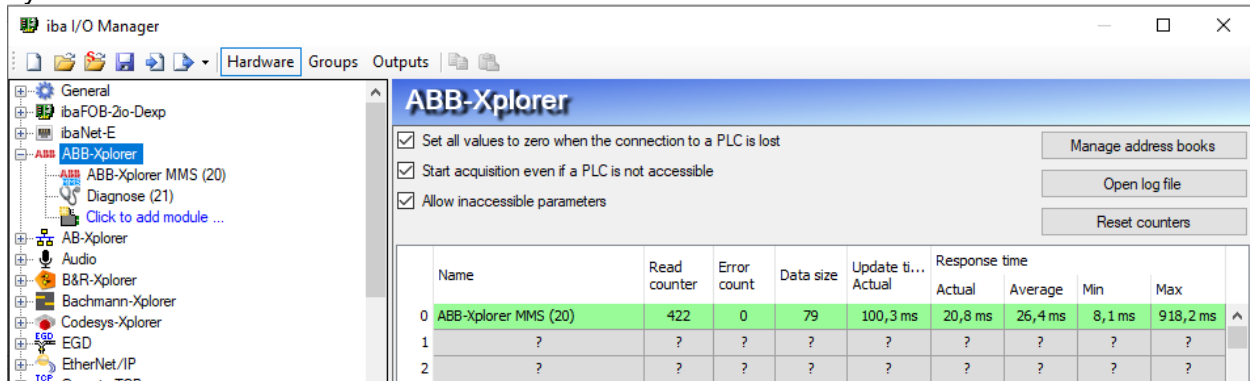
Since a password is sensitive information and shouldn't be visible in a configuration there is a new feature to encrypt text parameters. When you open the quotes and start typing a text then the intellisense window will present you with the option to "Encrypt text". When you select this, a dialog opens where you can enter the text that you want to encrypt. When you click "OK" the text is encrypted and put in the expression.



2 ABB-Xplorer Interface

2.1 Defining Xplorer module in ibaPDA

The ABB-Xplorer interface in ibaPDA is used to measure data from ABB AC800 PLCs. It is an Xplorer interface which means that the data is cyclically read by ibaPDA instead of being sent by the PLC.



The ABB-Xplorer interface shows a table of the available connections. Per ABB-Xplorer license you get 16 connections. A maximum of 240 connections are allowed. This means that maximum 15 licenses can be used. Each connection corresponds to a row in the table. The row is green when the connection is ok and data is being read. The row is orange when the connection is ok but the data is coming slower than the configured update time. The row is red when the connection could not be established. The row is grey when there is no connection configured. The response time is the time it takes to read the data for a connection. The table shows the actual, average, minimum and maximum values of the response time. The update time is the time between 2 read operations. The data size shows how much data is read per read operation; in between brackets is the number of commands used to request the data. You can use the “Reset counters” button to clear the counters for all connections. Clicking “Open log file” opens the most recent log file related to ABB-Xplorer connections.

On the interface you can also decide how to handle some error conditions:

- When the connection to a PLC is lost during the acquisition then you can choose if the values stay at the last read value or if they are set to zero.
- When a PLC is not accessible during the start of the acquisition then you can choose if the acquisition starts without this PLC or if the acquisition is not started. When the acquisition is started without the PLC then ibaPDA will periodically (every 10s) try to connect to the PLC during the acquisition. As long as the PLC is disconnected the values will remain at zero.
- When ibaPDA tries to access an operand that is not available when validating the I/O configuration, the PLC will return an error. If the option “Allow inaccessible operands” is enabled then ibaPDA will ignore this signal and start the acquisition without this signal. If the option is not enabled then the acquisition will not start.

The “General” tab of an ABB-Xplorer module looks as follows:

Basic	
Module Type	ABB-Xplorer MMS
Locked	False
Enabled	True
Name	ABB-Xplorer MMS
Module No.	20
Timebase	10 ms
Use name as prefix	False

Mms	
Update Time	100 ms

Module Layout	
No. analog signals	32
No. digital signals	32

Name
The name of the module.

[Select symbols](#)

Apart from the standard options that can be found on most other modules, the **Update time** has to be configured. This is the time in ms between two read operations.

New signals can be added by clicking **Select MMS variables** at the bottom. This list is only filled after a connection to the target PLC is configured and tested. Only variables defined as MMS access variables in the ABB Compact Control Builder can be used here.

ABB MMS symbol browser

Symbol: PowerFailureInfo.0

Datatype: INT32

Symbols Search

- Access Variables
 - Bool000
 - Bool001
 - Bool002
 - Bool003
 - Bool004
 - Bool005
 - Bool006
 - Bool007
 - DownLoadQuotaExc
 - OccOfRst
 - 0
 - 1
 - PowerFailureInfo
 - 0
 - 1
 - 2
 - Real000
 - Real001
 - Real002
 - Real003
 - Real004

☐ Hide symbols with an unsupported datatype

Update symbols Add Close

Using the MMS symbol browser you can easily add analog or digital signals to the MMS module, by double-clicking any variable, or selecting multiple variables and clicking on “Add”.

The screenshot shows the 'Connection' tab of the software. It contains the following fields and buttons:

- Address:** A text box containing '192.168.50.85'.
- Timeout (s):** A spinner box set to '5'.
- Maximum number of objects to read in a single command:** A spinner box set to '64'.
- Test:** A button to attempt a connection.
- Create address book:** A button to save the current configuration.

In the **Connection** tab, all parameters to establish a proper connection to the PLC have to be configured:

- **Address:** The IP address of the PLC at which the network interface of the PLC is located.
- **Timeout:** the amount of time after which a connection attempt will be aborted.
- **Maximum number of objects to read in a single command:** ibaPDA tries to group the requested items into as few requests as possible. If the load of the PLC-CPU is too high, limiting parameters this will lead to smaller packages which can be handled without causing a timeout.

When clicking the **Test** button, ibaPDA will try to establish a connection to the PLC and update the address book, overwriting any previous address book related to the selected IP address.

Click on the **Create address book** button in order to read the symbols from the PLC and save them locally. Then, the symbols are also offline available via the Symbol Browser, i.e. without a connection to the PLC. Thus, you can configure the signals for the measurement process.

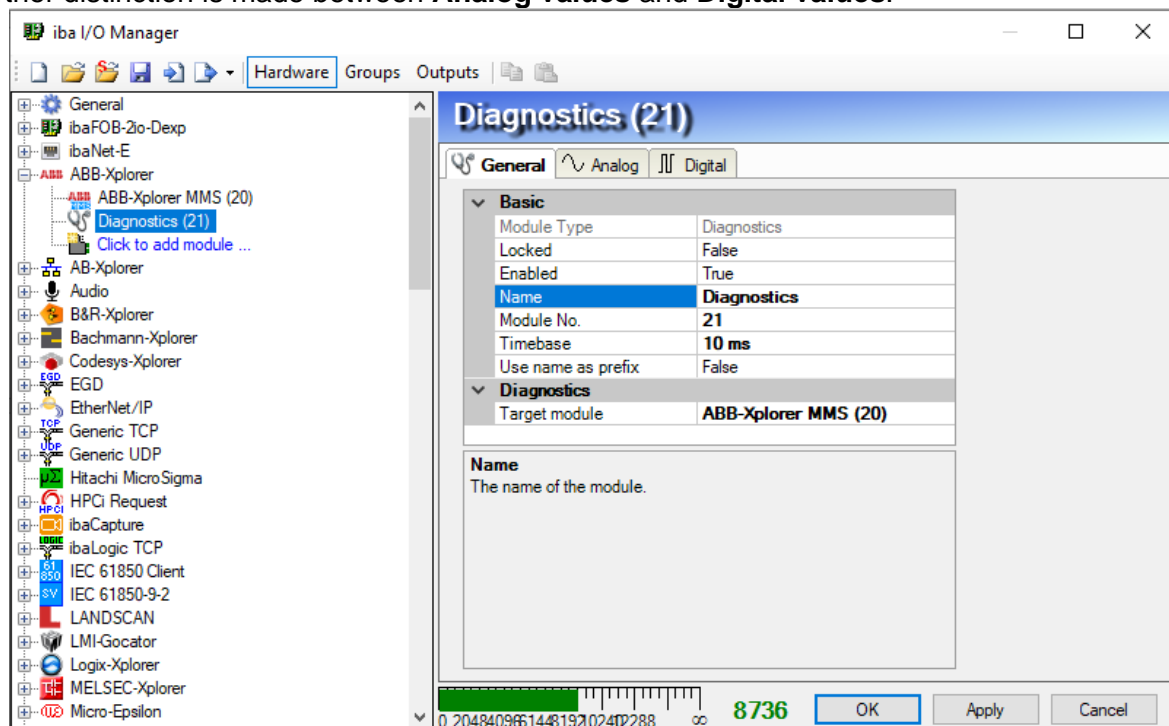
The screenshot shows the 'Analog' tab of the software. It displays a table with the following columns: Name, Unit, Gain, Offset, Symbol, DataType, and Active. The table contains 12 rows of test signals, all of which are active (checked).

	Name	Unit	Gain	Offset	Symbol	DataType	Active
12	TestStringFast		1	0	TestStringFast	STRING[255]	<input checked="" type="checkbox"/>
13	TestStringNormal		1	0	TestStringNormal	STRING[255]	<input checked="" type="checkbox"/>
14	TestStringST		1	0	TestStringST	STRING[255]	<input checked="" type="checkbox"/>
15	TestUIntFast		1	0	TestUIntFast	DINT	<input checked="" type="checkbox"/>
16	TestUIntNormal		1	0	TestUIntNormal	DINT	<input checked="" type="checkbox"/>
17	TestUIntST		1	0	TestUIntST	DINT	<input checked="" type="checkbox"/>
18	TestWordFast		1	0	TestWordFast	DINT	<input checked="" type="checkbox"/>
19	TestWordNormal		1	0	TestWordNormal	DINT	<input checked="" type="checkbox"/>
20	TestWordST		1	0	TestWordST	DINT	<input checked="" type="checkbox"/>
21			1	0		INT	<input type="checkbox"/>
22			1	0		INT	<input type="checkbox"/>
23			1	0		INT	<input type="checkbox"/>
24			1	0		INT	<input type="checkbox"/>

The above figure shows an example of the Analog tab of a MMS module. The datatype is internally synchronized to the current address book.

General Connection Analog Digital Diagnostics					
Analog values Digital values					
	Name	Symbol	Datatype	Value	Unit
0	TestDintFast	TestDintFast	DINT	16777216	
1	TestDintNormal	TestDintNormal	DINT	0	
2	TestDintST	TestDintST	DINT	128	
3	TestDwordFast	TestDwordFast	DINT	1266679808	
4	TestDwordNormal	TestDwordNormal	DINT	0	
5	TestDwordST	TestDwordST	DINT	0	
6	TestIntFast	TestIntFast	DINT	32767	
7	TestIntNormal	TestIntNormal	DINT	0	
8	TestIntST	TestIntST	DINT	505	
9	TestRealFast	TestRealFast	REAL	-0,084107	
10	TestRealNormal	TestRealNormal	REAL	500	
11	TestRealST	TestRealST	RFAI	-0.147861	

The current values of the requested topics can be monitored in the **Diagnostics** tab. There a further distinction is made between **Analog values** and **Digital values**.

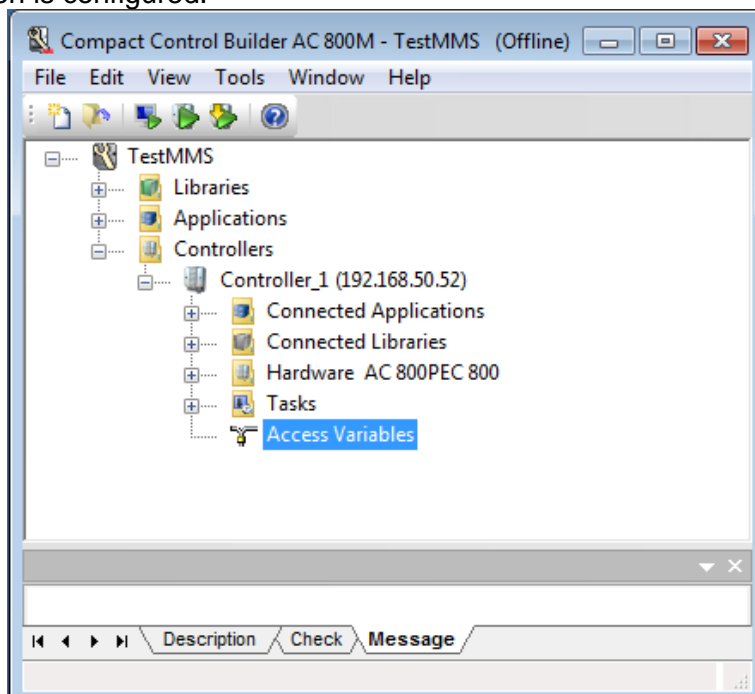


A diagnostic module can be added to record additional information about the PLC connection.

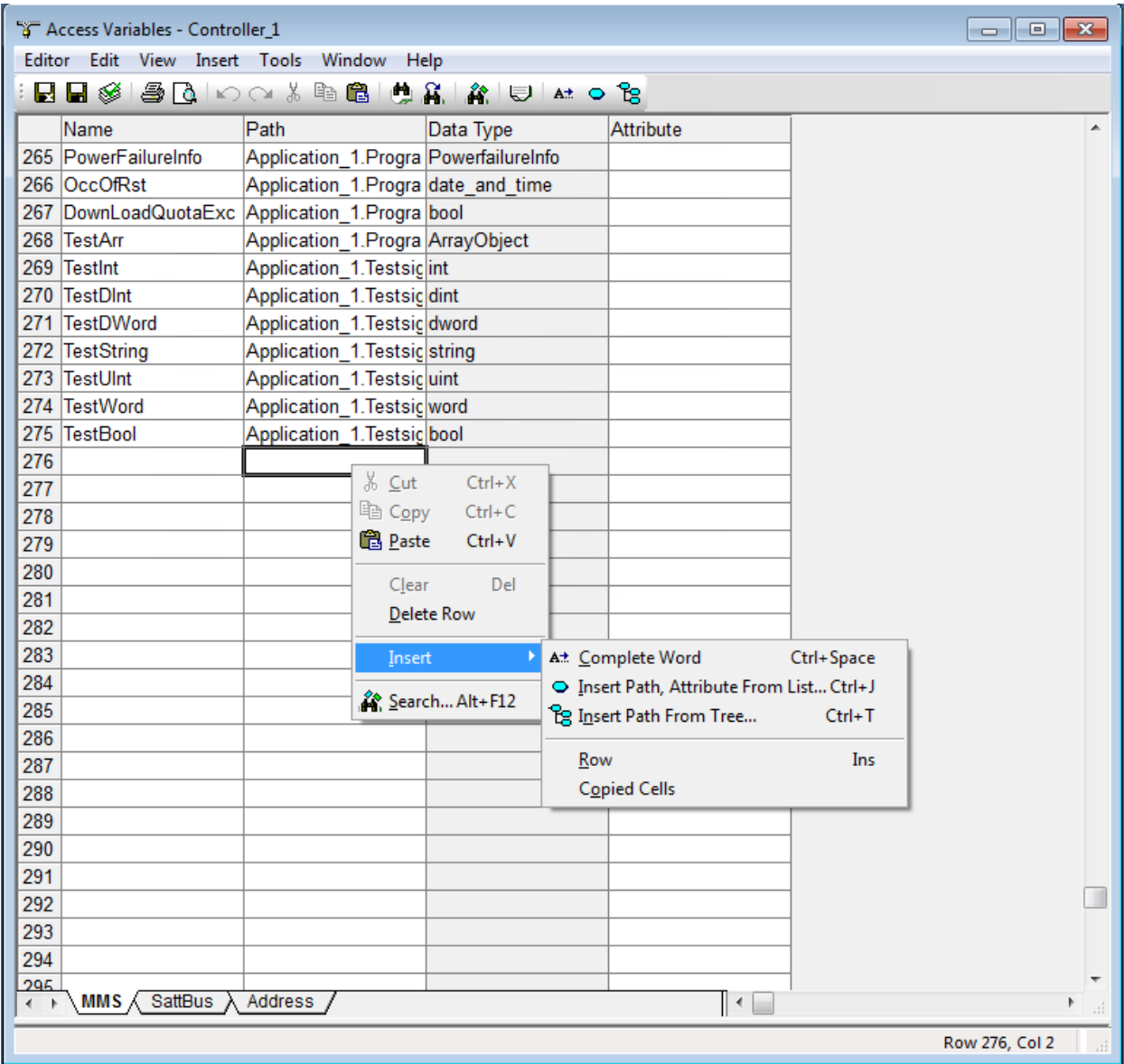
General Analog Digital						
	Name	Unit	Gain	Offset	Active	Actual
0	Message counter		1	0	<input checked="" type="checkbox"/>	3947
1	Data size		1	0	<input checked="" type="checkbox"/>	79
2	Error counter		1	0	<input checked="" type="checkbox"/>	0
3	Response time (actual)	ms	1	0	<input checked="" type="checkbox"/>	10,495 ms
4	Response time (average)	ms	1	0	<input checked="" type="checkbox"/>	10,495 ms
5	Response time (min)	ms	1	0	<input checked="" type="checkbox"/>	7,0319 ms
6	Response time (max)	ms	1	0	<input checked="" type="checkbox"/>	918,181 ms
7	Update time (actual)	ms	1	0	<input checked="" type="checkbox"/>	100,675 ms
8	Update time (configured)	ms	1	0	<input checked="" type="checkbox"/>	100 ms

2.2 Defining access variables in ABB Compact Control Builder

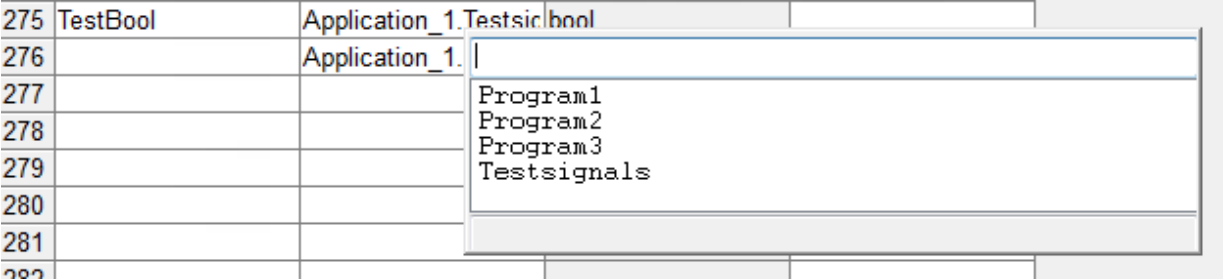
All variables accessible from ibaPDA must be mapped into the MMS Access Variables in the ABB Compact control builder. For activating and configuring the MMS connection for your PLC, please refer to the manufacturer manual. Normally MMS should be activated by default once the Ethernet connection is configured.



Open the used controller and edit the Access Variables. Select the MMS tab here.



You can define a name for the access variable here, and select the appropriate path to the variable inside the program. For this, you can either type it in and use the auto-fill feature (Complete Word, Insert Path From List) or use the variable browser (Insert Path From Tree).



The screenshot shows the 'Access Variables - Controller_1*' window. The main table lists variables with their names, paths, data types, and attributes. A tree view on the right shows the hierarchy of variables under 'Application_1'.

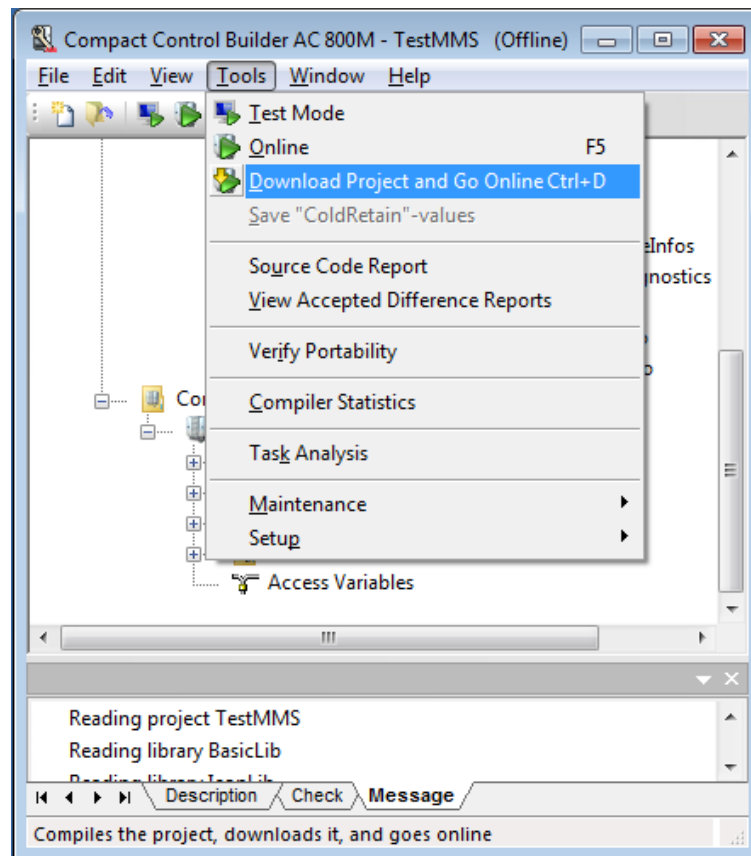
	Name	Path	Data Type	Attribute
265	PowerFailureInfo	Application_1.Progra	PowerfailureInfo	
266	OccOfRst	Application_1.Progra	date_and_time	
267	DownLoadQuotaExc	Application_1.Progra	bool	
268	TestArr	Application_1.Progra	ArrayObject	
269	TestInt	Application_1.Testsig	int	
270	TestDInt	Application_1.Testsig	dint	
271	TestDWord	Application_1.Testsig	dword	
272	TestString	Application_1.Testsig	string	
273	TestUInt	Application_1.Testsig	uint	
274	TestWord	Application_1.Testsig	word	
275	TestBool	Application_1.Testsig	bool	
276		Application_1.	not found	

The tree view on the right shows the following structure:

- Application_1
 - Testsignals
 - CTU_1
 - SinGen_1
 - Bool000
 - Bool001
 - Bool002
 - Bool003
 - Bool004
 - Bool005
 - Bool006
 - Bool007
 - Real000
 - Real001
 - Real002
 - Real003
 - Real004
 - Real005
 - Real006
 - Real007
 - Real008

The status bar at the bottom indicates 'Row 276, Col 2'.

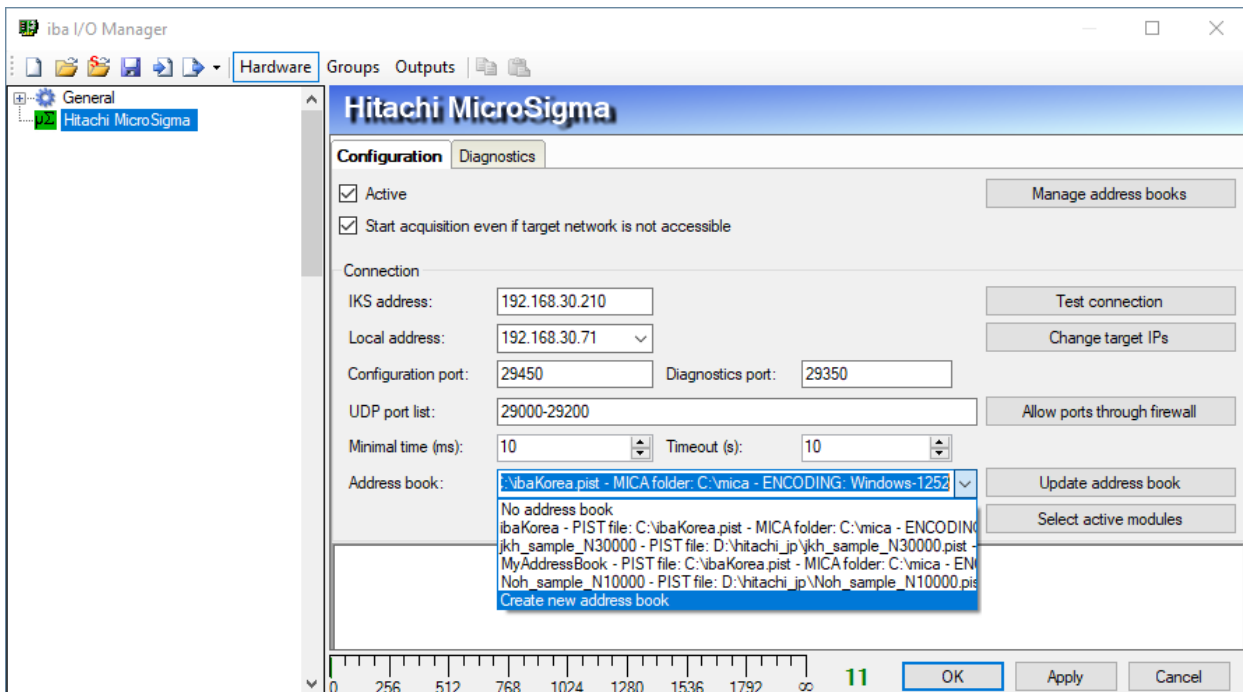
When all variables are defined, select “Download Program and Go Online” from the main menu to update the program in the PLC.



In ibaPDA, press the button “Test” in the connection tab to update the address book. Now you can select the new variables in the MMS symbol browser and add them to your ibaPDA project.

3 Hitachi MicroSigma Interface

The MicroSigma interface in ibaPDA is used to connect to an IKS-LM-SN1G or IKS-LM-SN100 device, which is connected to a MicroSigma PLC network.



The Interface hosts the connection configuration. With the activation of the Hitachi license you can configure one connection to an IKS-LM-SN1G/100. This connection can contain several modules. When you set the interface to active, the appropriate settings can be selected.

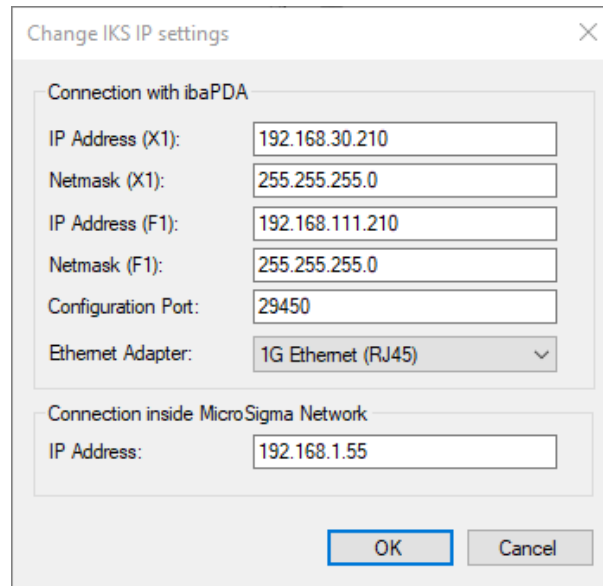
- **IKS address:** The IP address of the target IKS used to communicate with ibaPDA.
- **Local address:** The interface address of the ibaPDA server used to receive data telegrams from the IKS
- **Configuration port:** The IKS port used to update the configuration
- **Diagnostics port:** The port for the module where the IKS sends internal diagnostic data to. Must be outside the UDP port list range
- **UDP port list:** range of ports used for the modules receiving data from the IKS
- **Minimal time:** The minimal update time for the data modules
- **Timeout:** Timeout for communication
- **Address book:** The address book used for the current connection
- **Test connection:** Test connection to the target IKS
- **Change target IPs:** Change the IPs used by the IKS.
- **Get address book:** Parse PIST solution file and Mica project folder for available variables and modules
- **Select active modules:** If an address book is available, select the modules to be used by ibaPDA

The screenshot shows a Windows-style dialog box titled "Import MicroSigma solution". It has a blue header bar with a close button (X) in the top right corner. The dialog is divided into several sections:

- Name:** A text field containing "New Plant".
- Encoding:** A text field containing "iso-2022-jp".
- Code page:** A text field containing "932".
- PIST solution file:** A section containing:
 - Path from server:** A text field with "C:\same_plant.pist" and a browse button (...).
 - Username:** An empty text field.
 - Password:** An empty text field.
 - Test:** A button to test the connection.
- Mica folder:** A section containing:
 - Path from server:** A text field with "C:\mica" and a browse button (...).
 - Username:** An empty text field.
 - Password:** An empty text field.
 - Test:** A button to test the connection.
- Buttons:** At the bottom right, there are "OK" and "Cancel" buttons.

By selecting the option "Create new address book" in the address book dropdown list, a new address book can be created by importing the defined variable settings from a Hitachi project:

- **Name:** The name of the new address book
- **Encoding** and **Code page:** the code page used for reading the project files, if the PC running the Hitachi software is using another language setting than the ibaPDA PC
- **PIST solution file:**
 - Path from server: Path to the PIST solution file, can be selected using the browse button
 - Username: Optional username for accessing the file if it resides on a network share
 - Password: Optional password for accessing the file if it resides on a network share
 - Test: Test access to the path containing the PIST solution file
- **Mica folder:**
 - Path from server: Path to the mica project folder, can be selected using the browse button
 - Username: Optional username for accessing the folder if it resides on a network share
 - Password: Optional password for accessing the folder if it resides on a network share
 - Test: Test access to the path of the folder



Change IKS IP settings

Connection with ibaPDA

IP Address (X1): 192.168.30.210

Netmask (X1): 255.255.255.0

IP Address (F1): 192.168.111.210

Netmask (F1): 255.255.255.0

Configuration Port: 29450

Ethernet Adapter: 1G Ethernet (RJ45)

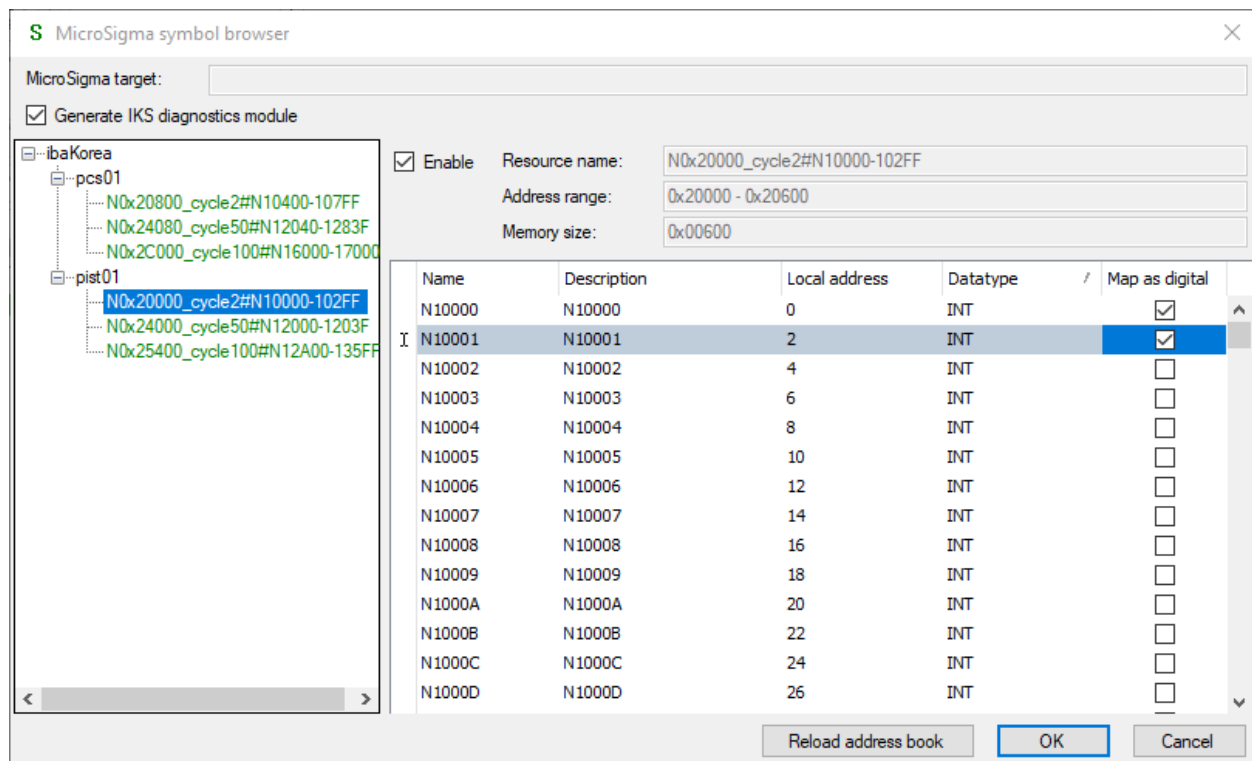
Connection inside MicroSigma Network

IP Address: 192.168.1.55

OK Cancel

With the dialog “Change Target IPs” the IP addresses, Ethernet adapter, and the configuration port used by the IKS device can be changed. After changing an IP address, the IKS has to be rebooted, which takes about 2 minutes.

When an address book is loaded, the contained modules can be selected to be used by ibaPDA. When pressing the button <Select active modules> the MicroSigma signal browser opens.



MicroSigma target:

☒ Generate IKS diagnostics module

☒ Enable Resource name: N0x20000_cycle2#N10000-102FF

Address range: 0x20000 - 0x20600

Memory size: 0x00600

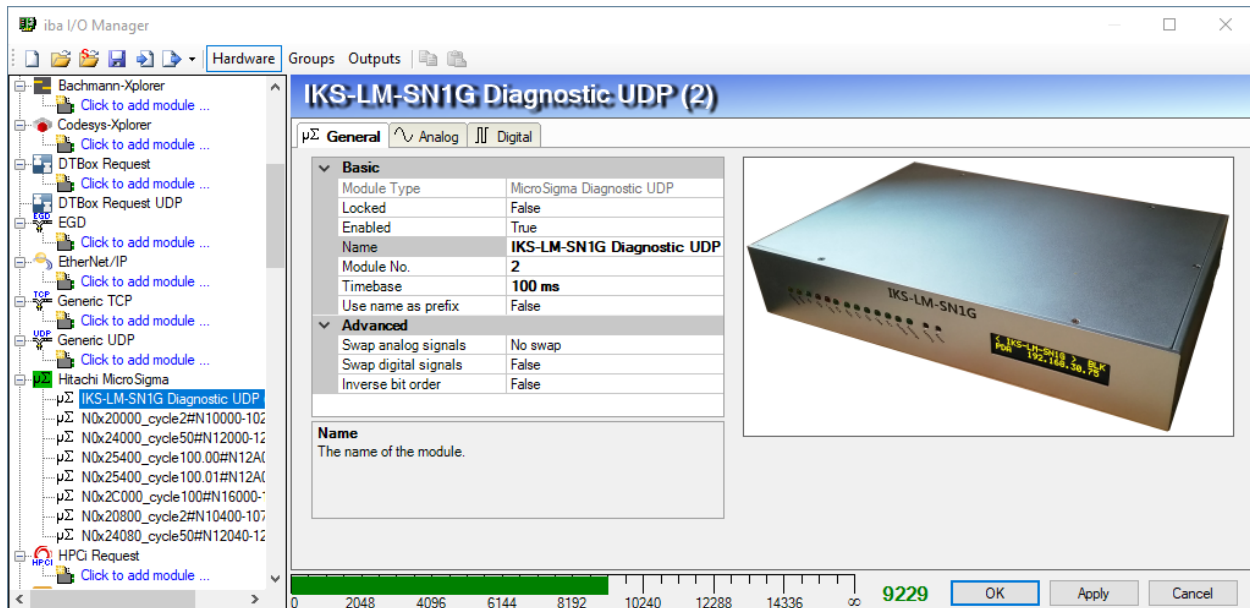
Name	Description	Local address	Datatype	Map as digital
N10000	N10000	0	INT	<input checked="" type="checkbox"/>
N10001	N10001	2	INT	<input checked="" type="checkbox"/>
N10002	N10002	4	INT	<input type="checkbox"/>
N10003	N10003	6	INT	<input type="checkbox"/>
N10004	N10004	8	INT	<input type="checkbox"/>
N10005	N10005	10	INT	<input type="checkbox"/>
N10006	N10006	12	INT	<input type="checkbox"/>
N10007	N10007	14	INT	<input type="checkbox"/>
N10008	N10008	16	INT	<input type="checkbox"/>
N10009	N10009	18	INT	<input type="checkbox"/>
N1000A	N1000A	20	INT	<input type="checkbox"/>
N1000B	N1000B	22	INT	<input type="checkbox"/>
N1000C	N1000C	24	INT	<input type="checkbox"/>
N1000D	N1000D	26	INT	<input type="checkbox"/>

Reload address book OK Cancel

When checking the “Generate IKS Diagnostics module” checkbox, the diagnostic module using the Diagnostics Port will be created.

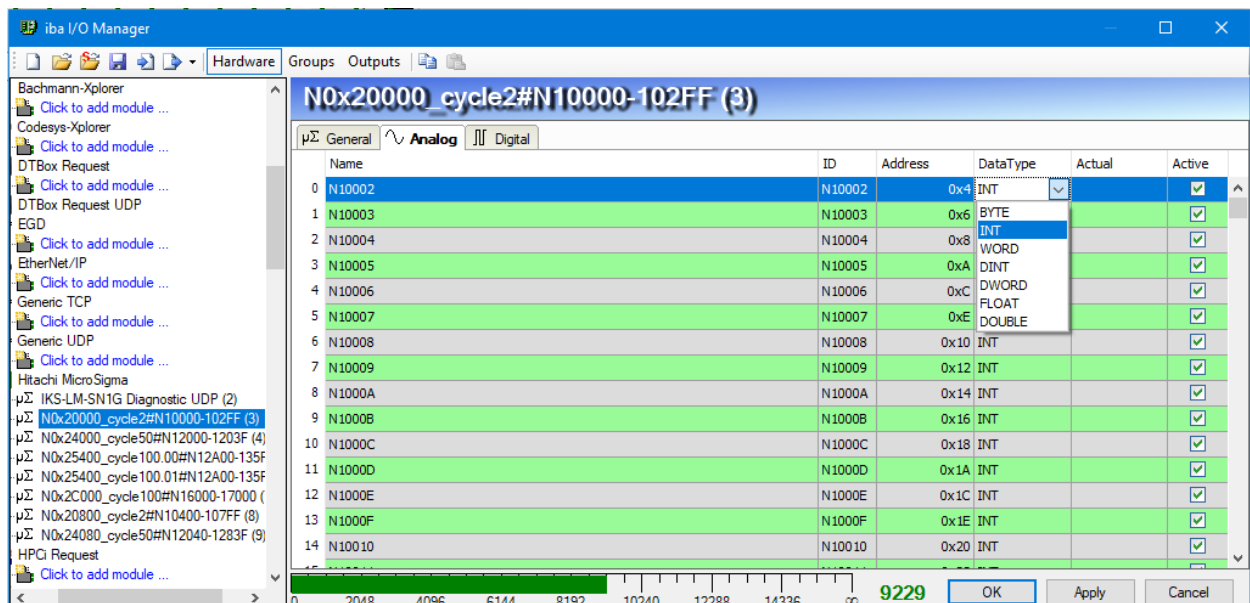
You can activate each module by checking the “Enable” checkbox. For all signals of the modules, you can select if the signal is used as 16bit integer or as 16 digital values by checking the appropriate “Map as digital” checkbox.

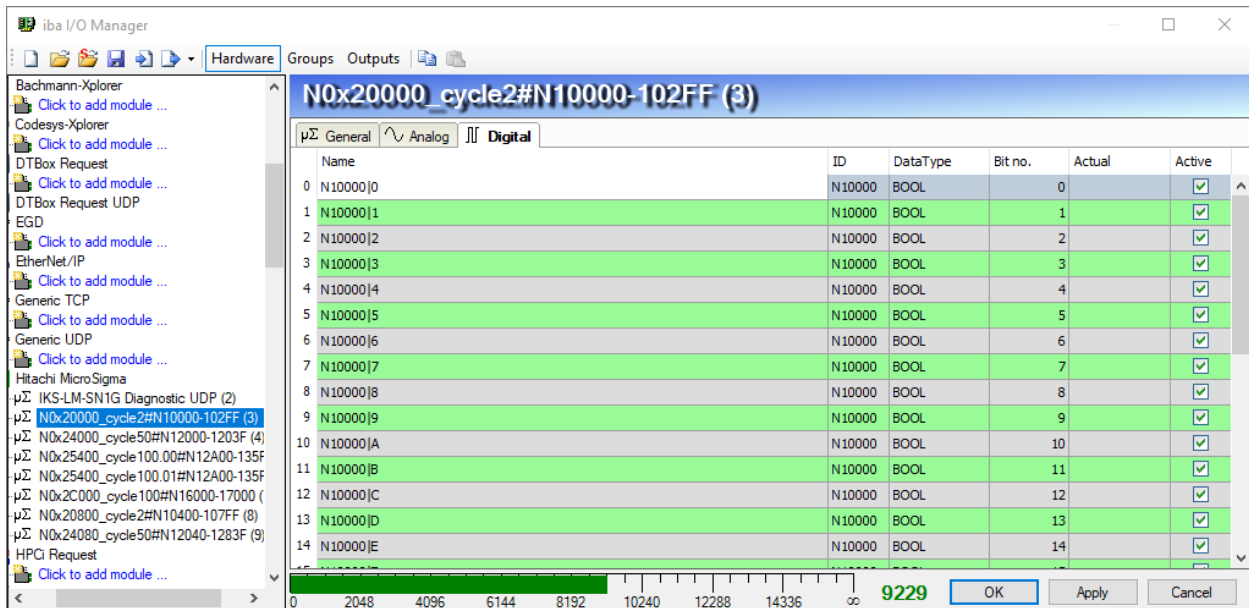
When you press <OK>, the selected modules are generated. If a module is too large for a single UDP telegram (2048 integer signals), it is split into multiple modules.



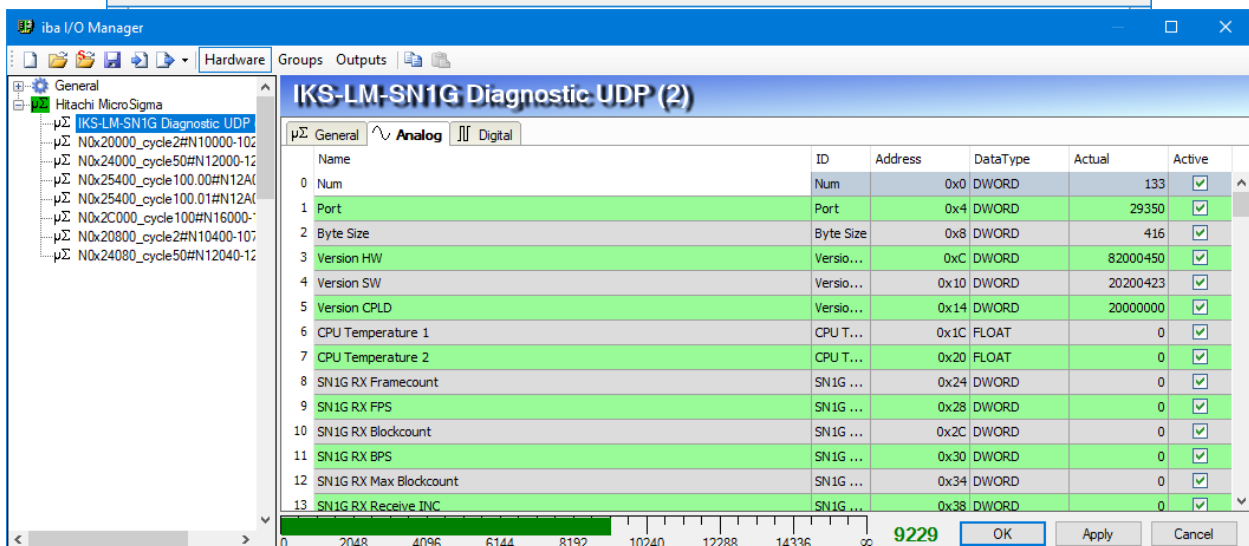
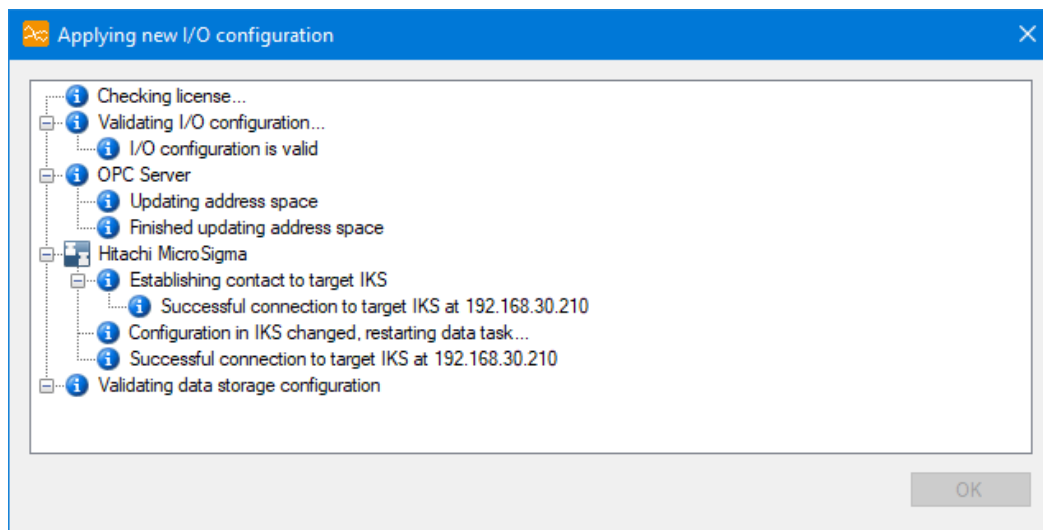
The modules have advanced settings for the handling of the signals. For the Diagnostic UDP module, **Swap analog signals** must be set to **No Swap**, **Swap digital signals** must be set to **False**, and **Inverse bit order** must be set to **False**. For the Data UDP modules, **Swap analog signals** must be set to **Depending on datatype**, **Swap digital signals** must be set to **True**, and **Inverse bit order** must be set to **True**.

Within each module, you can select if the signal will be stored in ibaPDA by checking the appropriate "Active" checkbox. For each analog signal, the data type can be changed. Default data type is **INT** (16bit).

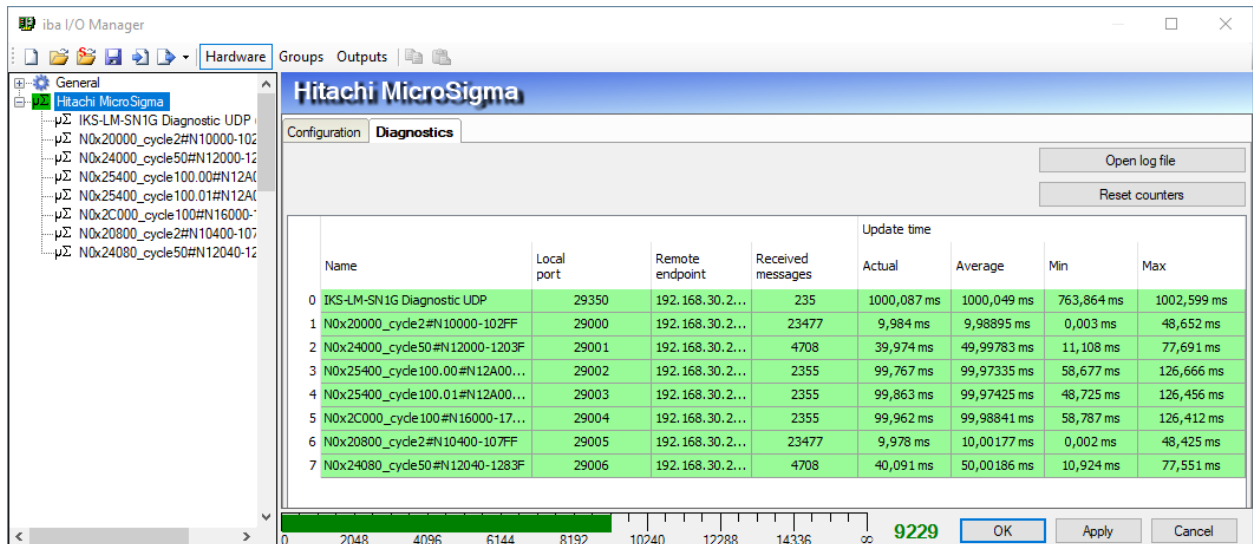




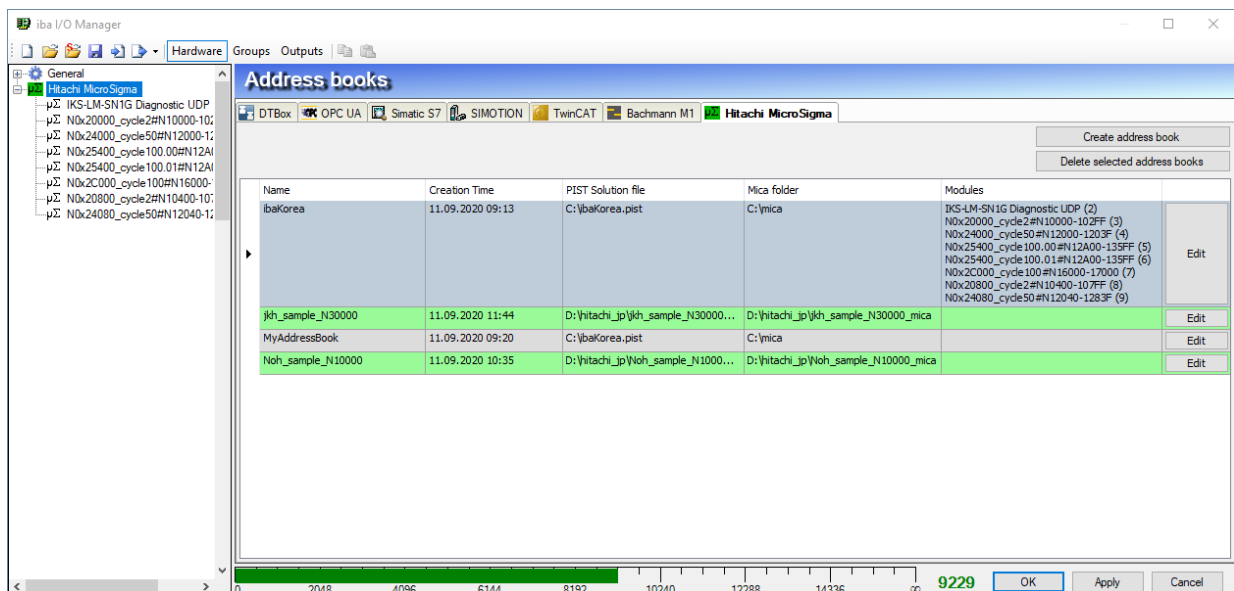
When starting the acquisition, the active configuration is checked and updated if needed. After changing the configuration, the data handling process on the IKS is restarted automatically.



When the acquisition is running, the actual values can be watched in the “Analog” resp. “Digital” tabs of the generated modules.



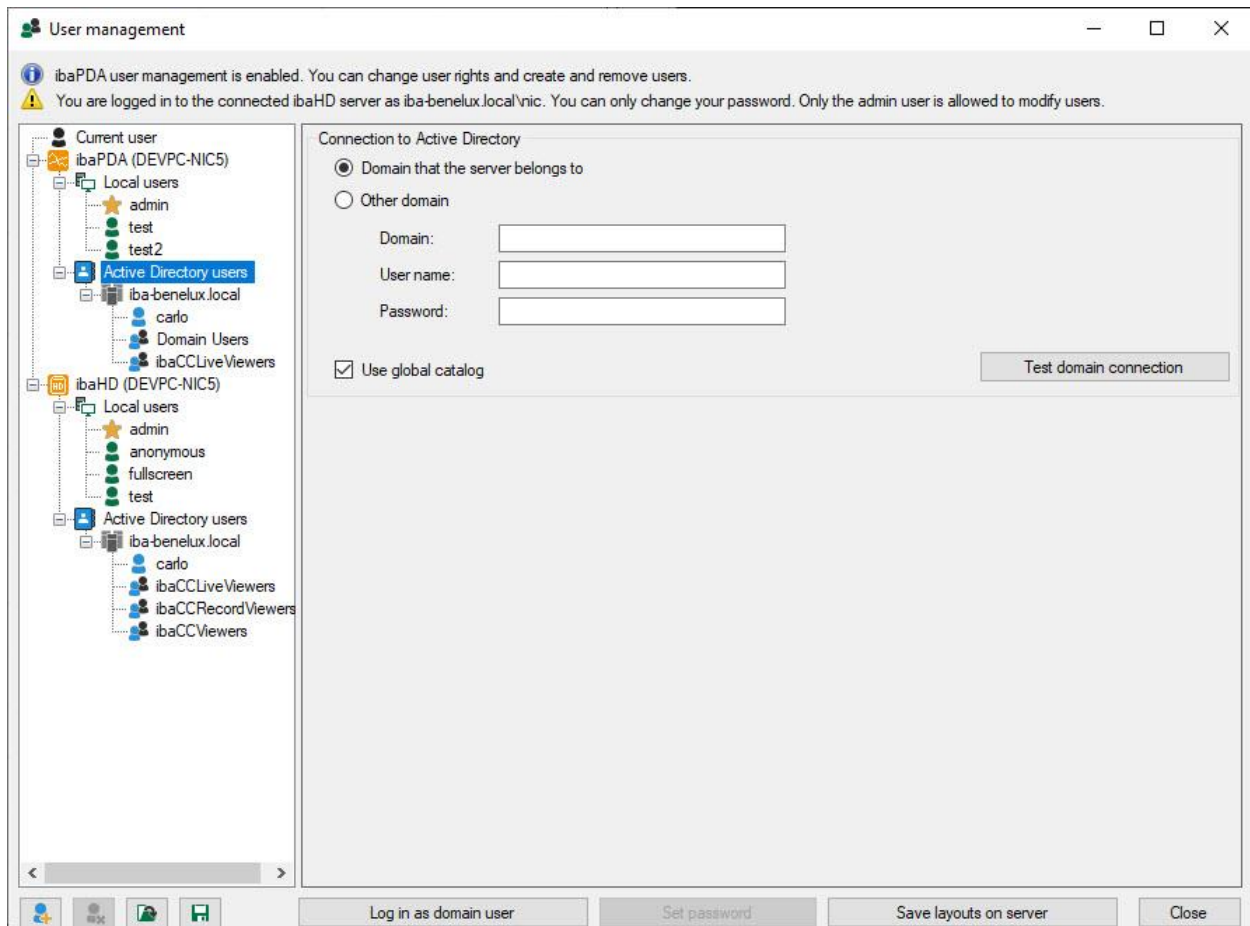
The interface “Diagnostics” tab now shows statistics concerning the data updates received from the IKS.



With the address books manager, the existing imported address books can be checked, edited or deleted.

4 Active Directory support in user management

The user management in ibaPDA is configured in the user management dialog. You can open this via the “*Configure*” menu or via the keyboard shortcut CTRL+U.



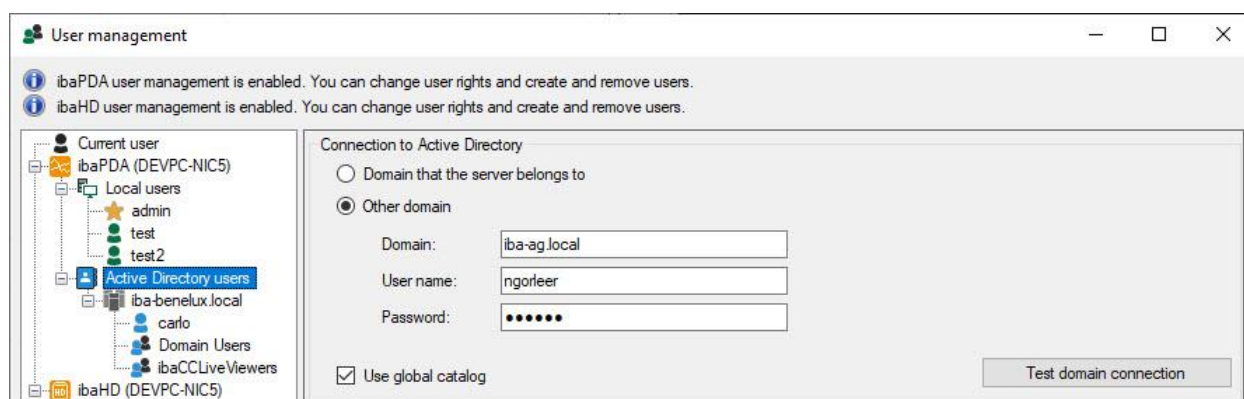
In this dialog you can configure the user management of both ibaPDA and ibaHD. In ibaPDA v7.3.0 and later and ibaHD v2.6.0 and later 2 types of users are supported:

- Local users
- Active Directory users

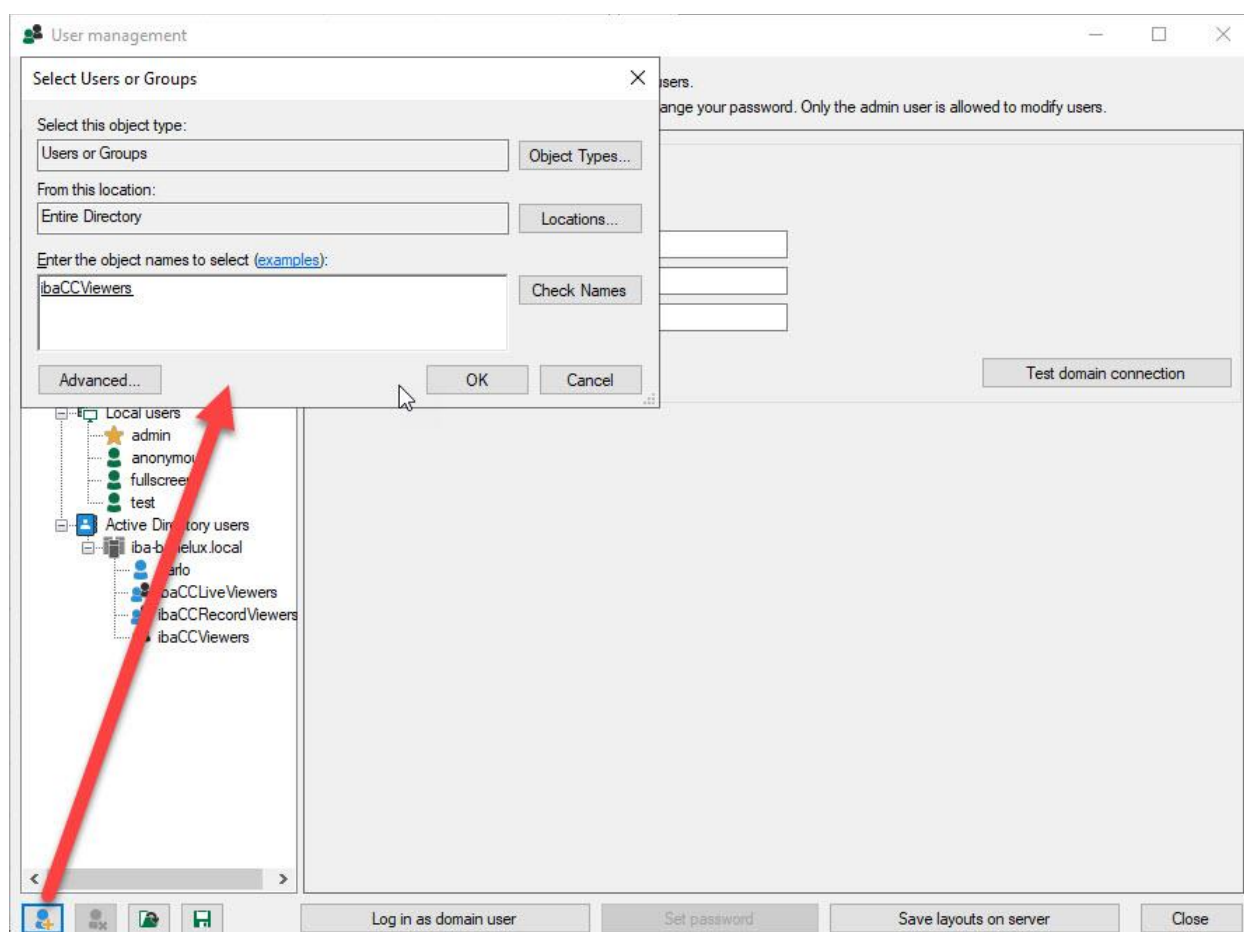
The local users are users created in ibaPDA/ibaHD. The local user “*admin*” is always present. He has the right to create other users and to assign rights to them. User management is enabled by assigning a password to the admin user. You can disable user management again by assigning an empty password to the admin user.

Active Directory users are users or groups defined in Windows Active Directory. On the Active Directory users node you can configure to which Active Directory domain the ibaPDA/ibaHD server should connect. If the ibaPDA/ibaHD server PC is part of a domain then you can choose to use that domain. It is also possible to use another domain by entering the domain name and credentials of a user within that domain. It is recommended to enter the fully qualified domain name, e.g. *iba-ag.local* instead of just *iba-ag*.

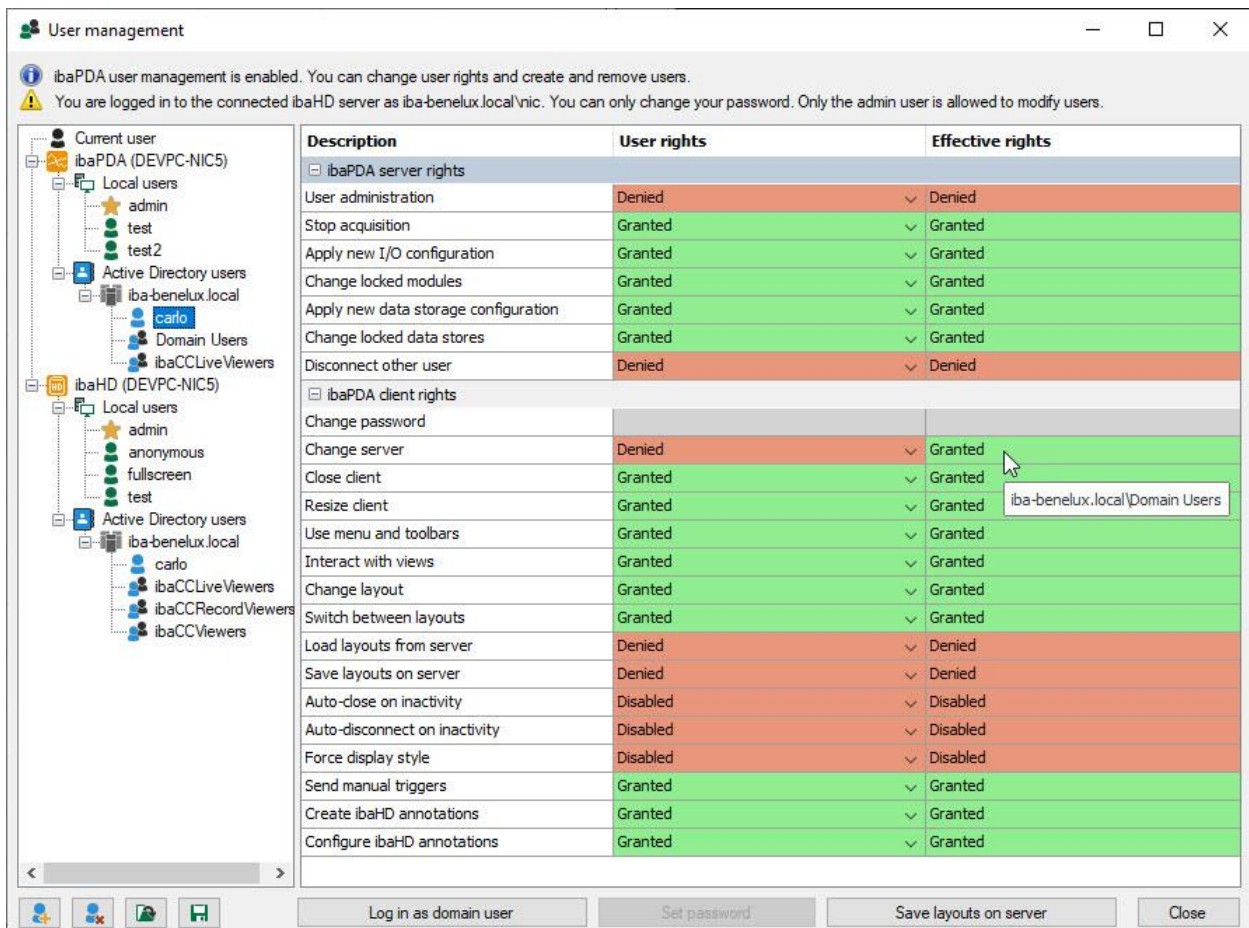
The ibaPDA/ibaHD server will look for Active Directory users and groups in the configured domain and its sub domains. When the “*Use global catalog*” option is checked then ibaPDA/ibaHD server will additionally look in parent domains and trusted domains.



Via the buttons at the bottom under the tree you can add and remove users. You can add a local user when you have the “*Local users*” node or one of its child nodes selected. You can add a domain user or group when you have the “*Active Directory users*” or one of its child nodes selected.



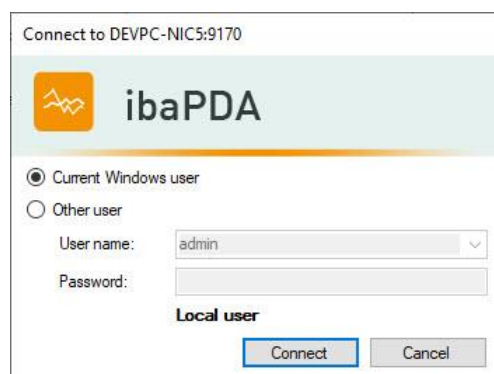
In case of Active Directory users the standard Windows Active Directory selection dialog opens. You can select both users and groups. The added users and groups will appear under a domain node in the tree.



You can assign rights to a user or a group when you are logged in as a user that has the “*User administration*” right. Rights that are not available are grayed out. For example the “*Change password*” right is not available for Active Directory users and groups. By right-clicking on the table you can copy the rights of a user to another user.

An Active Directory user can belong to multiple groups. Rights can be assigned to the user directly or they can be inherited from the groups the user belongs to. The effective rights column shows the result of this mechanism. In the screenshot the “*Change server*” right was denied for user “*Carlo*” but he still gets the right because he is a member of the “*Domain Users*” group and the “*Domain Users*” group has this right granted. If a right is granted then the tooltip shows from which source(s) the right is coming. This effective rights column is only shown for Active Directory users and not for groups and local users.

In the user management dialog you can log in as another user by clicking the log in button at the bottom.



The login form will appear and you have several options to log in.

You can select to log in as the current Windows user. This will only work when you are logged in on Windows as a domain user. The ibaPDA/ibaHD server will try to find the user via the configured Active Directory connection. If the server is able to find the user then it will look for any rights that are granted to the user either directly or via groups the user belongs to. If the user has some rights then he is accepted and the log in succeeds. If the user doesn't have any rights then the log in is denied. So in the case of the *Current Windows user* you don't have to provide a password.

You can also choose to log in by providing complete user credentials. The user name can be a local user. The drop down shows the list of local users. The ibaPDA/ibaHD server will check if the local user exists and if the password is correct. If this is ok then the log in will succeed.

The user name can also be an Active Directory user. You have to use the following syntax: domain\user. The domain name can be the short name or the fully qualified domain name like e.g. iba-ag.local. If the short domain name doesn't work then please try the fully qualified domain name. The user part should be the (legacy) user logon name. You should also provide the Active Directory user's password. The ibaPDA/ibaHD server will try to log in to the domain from the user name using the provided credentials. So the server will **not** use the configured Active Directory connection. If the login to the domain succeeds then the server will look for any rights that are granted to the user either directly or via groups the user belongs to. If the user has some rights then he is accepted and the log in succeeds. If the user doesn't have any rights then the log in is denied.

User management

⚠ You are logged in to the connected ibaPDA server as iba-benelux.local\nic. You can not modify any rights. Only the admin user is allowed to modify users.

⚠ You are logged in to the connected ibaHD server as iba-benelux.local\nic. You can only change your password. Only the admin user is allowed to modify users.

Current user

- ibaPDA (DEVPC-NIC5)
 - Local users
 - admin
 - test
 - test2
 - Active Directory users
 - iba-benelux.local
 - carlo
 - Domain Users
 - ibaCCLiveViewers
- ibaHD (DEVPC-NIC5)
 - Local users
 - admin
 - anonymous
 - fullscreen
 - test
 - Active Directory users
 - iba-benelux.local
 - carlo
 - ibaCCLiveViewers
 - ibaCCRecordViewers
 - ibaCCViewers

Description	ibaPDA user: iba-benelux.local\nic	ibaHD user: iba-benelux.local\nic	Effective rights
ibaPDA server rights			
User administration	Denied		Denied
Stop acquisition	Granted		Granted
Apply new I/O configuration	Granted		Granted
Change locked modules	Granted		Granted
Apply new data storage configuration	Granted		Granted
Change locked data stores	Granted		Granted
Disconnect other user	Denied		Denied
ibaPDA client rights			
Change password			
Change server	Granted	Granted	
Close client	Granted		Granted
Resize client	Granted		Granted
Use menu and toolbars	Granted	Granted	Granted
Interact with views	Granted	Granted	Granted
Change layout	Denied	Granted	Denied
Switch between layouts	Granted	Granted	Granted
Load layouts from server	Denied	Denied	
Save layouts on server	Denied	Denied	
Auto-close on inactivity	Disabled	Disabled	Disabled
Auto-disconnect on inactivity	Disabled	Disabled	
Force display style	Disabled	Disabled	Disabled
Send manual triggers	Granted	Granted	Granted
Create ibaHD annotations	Granted	Granted	Granted
Configure ibaHD annotations	Granted	Granted	Granted

Log in as admin Set password Save layouts on server Close

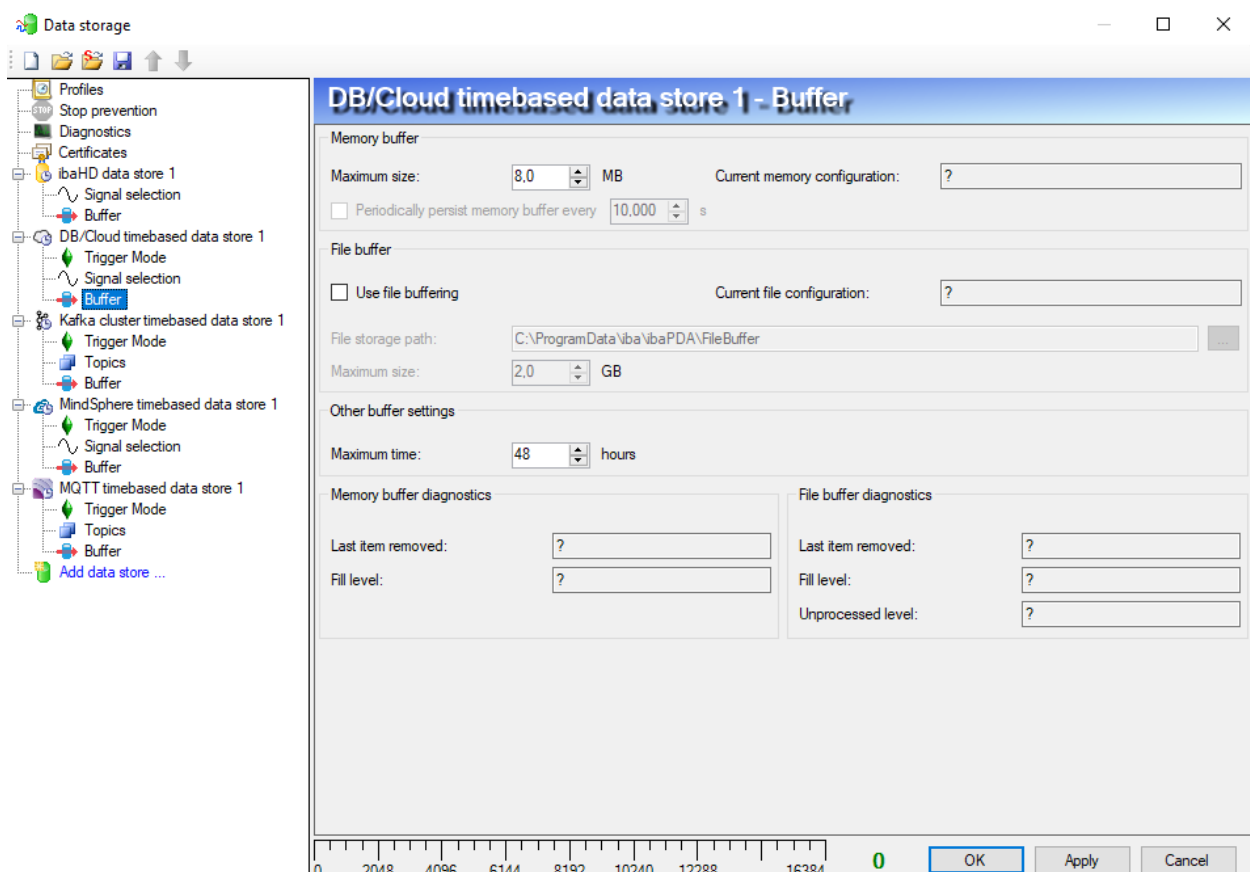
The ibaPDA client can be connected to both an ibaPDA and an ibaHD server. It can be connected as different users to both. The “Current user” node in the tree gives information about the user accounts used to connect to the ibaPDA and ibaHD server. The “*ibaPDA user*” column shows the rights granted to the user that was used to log in to the ibaPDA server. If it is an Active Directory user then the tooltip shows from which user and groups the rights are coming. The “*ibaHD user*” column shows the same for the ibaHD user. The effective rights column shows the result of combining the right from the ibaPDA user with the right from the ibaHD user. For most rights an AND relationship is used. This means that both users need to have the right for the effective right to be granted. In the screenshot you can see that the “*Change server*” right is granted to both users and that the effective right is then also granted. The “*Change layout*” right is denied for the ibaPDA user and granted to the ibaHD user. The effective right is then denied. The ibaPDA server rights are only available for ibaPDA users. Some of the ibaPDA client rights are handled separately for ibaPDA and ibaHD. They are not combined. These rights are grayed out in the “*Effective rights*” column.

- A greater number of items can be buffered, while potentially consuming less RAM memory.
- The buffer can be persisted if the acquisition stops. Buffered items that would otherwise be lost can still be retrieved later on.

The MQTT timebased data store already had the option to buffer data in memory or files, but the functionality is now expanded.

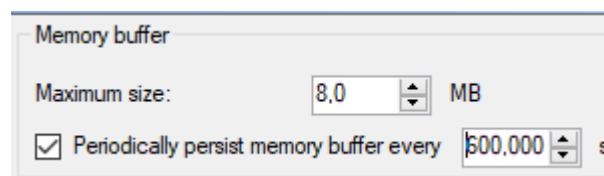
5.1 Configuration options

To configure the buffer, select the buffer node for the corresponding data store. The following picture shows the default settings for most data store buffers:



Depending on your needs, the buffer can be configured.

5.1.1 Memory buffer



The memory buffer is always enabled. It can't be disabled.

5.1.1.1 Maximum size

This is the maximum total size of the items buffered in memory. If the maximum size is exceeded, there are 2 options:

- If file buffering is disabled, the oldest item in memory will be dropped (and it is lost forever).

- If file buffering is enabled, the oldest part of the memory buffer is moved to a buffer file. Setting the maximum size is a tradeoff between performance and RAM memory usage.

5.1.1.2 Periodically persist

This option can only be enabled if the file buffer is enabled. If enabled, the whole memory buffer is persisted to a buffer file periodically. The flush period can also be configured. It must be between 10 s and 10 minutes (600 s).

This option can be used to reduce the probability that data is lost due to a power failure or crash in the ibaPDA server. The typical use case is when a small amount of valuable data must be handled with a bit more certainty.

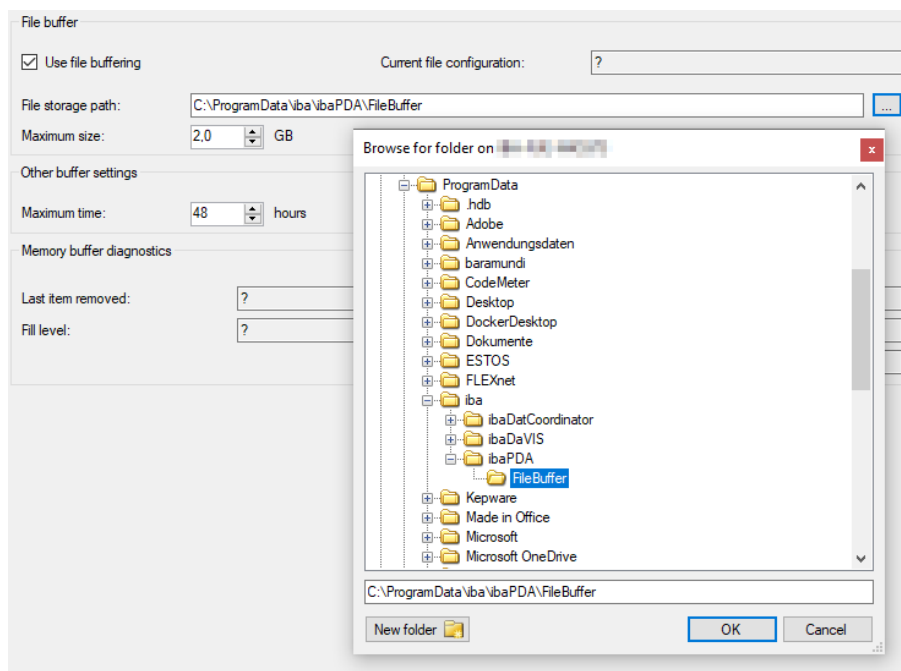
5.1.2 File buffer

By default, the file buffer is not used. It can be enabled by checking the box next to “Use file buffering”.

If the file buffer is used, it can produce files. The location where the files are stored can be chosen. Use a local file system location (or a place recognized as such that e.g. resides on a SAN). The file location is as seen from the computer where the ibaPDA server runs.

The same file location can be used for multiple buffer files, because the files generated by the buffer of each data store have a unique name. Files from different buffers can thus be distinguished by their name.

The location can be typed in the text box, or the “...” button can be used to choose a location with the following dialog:



The maximum size of the buffer files can be configured as well. There can be multiple buffer files in a file buffer. The buffer data files have the `.buf` file extension the index file belonging to

the data files has the file extension *.info*. The maximum size is the combined size of all the buffer data files.

If the maximum size of the file buffer is exceeded, the oldest file will be deleted.

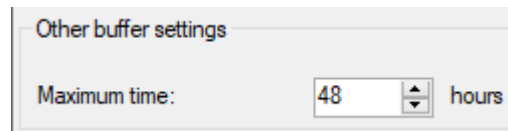
When the acquisition stops while there are still items present in the memory buffer, those items will be written to the file buffer. If file buffering is disabled, those items will be lost if they can't be written to the data sink in a timely fashion.

When the acquisition is started, the buffer resumes with the already existing buffer files. However, if file buffering is disabled, existing buffer files will be deleted when the acquisition starts.

5.1.2.1 Buffer files from the MQTT timebased data store in ibaPDA 7.2.x

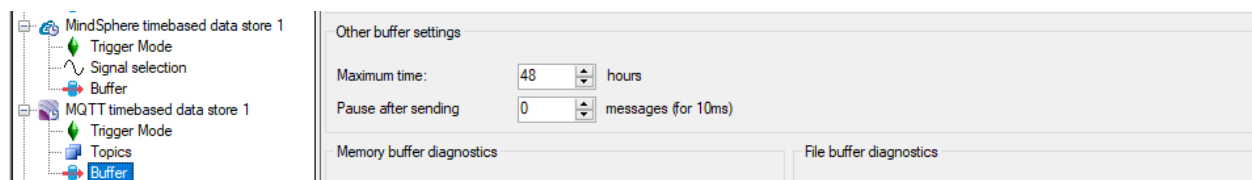
Buffer files that are created with the MQTT timebased data store a ibaPDA version 7.2.x are not compatible with ibaPDA 7.3.0. In the rare case that such buffer files exist and you do a version update to v7.3.0 at this point of time, the buffer files will be lost.

5.1.3 Other buffer settings



An additional setting that is valid for the buffer as a whole is the maximum time. Items older than the maximum time will not be sent to the data sink. Files older than the maximum time can be deleted. The maximum time has a value between 1 and 1000 hours.

For MQTT, there is an additional setting “Pause after sending ... messages”. That option limits the rate by which items are sent to the MQTT broker, to avoid that it collapses when too many items are sent at once. That can happen when the broker comes back online when the buffer has already considerably grown.



The options available for the archiver file buffer are described below.

5.2 Diagnostics

5.2.1 In the buffer configuration control

The highlighted areas in the picture above show the actual status of the buffer in the ibaPDA server. “Current memory configuration” and “Current file configuration” show the settings along with an estimate of how long items can be buffered.

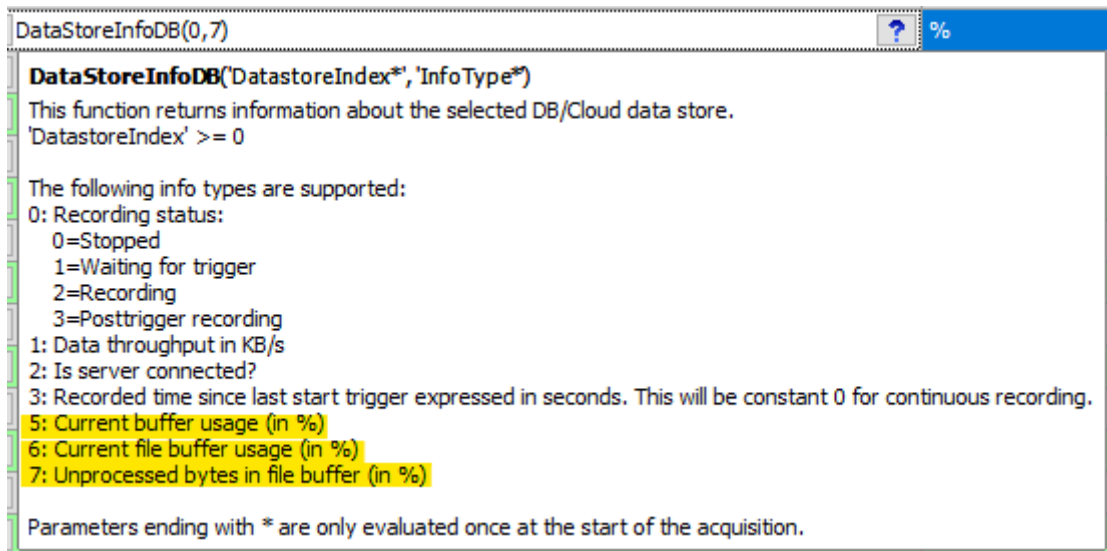
“Last item removed” in both the file and memory buffer diagnostics shows when the last item was taken from that part of the buffer.

The fill level shows the percentage of used space in either the memory or the file buffer. In the file buffer, items that are sent to the data sink are not immediately deleted. Only when a buffer file is read completely, it is deleted. So it is possible that only part of a buffer file contains useful data. The fill level represents the file size in %, while the “Unprocessed level” represents only the useful data in the file buffer, in %.

5.2.2 Virtual functions

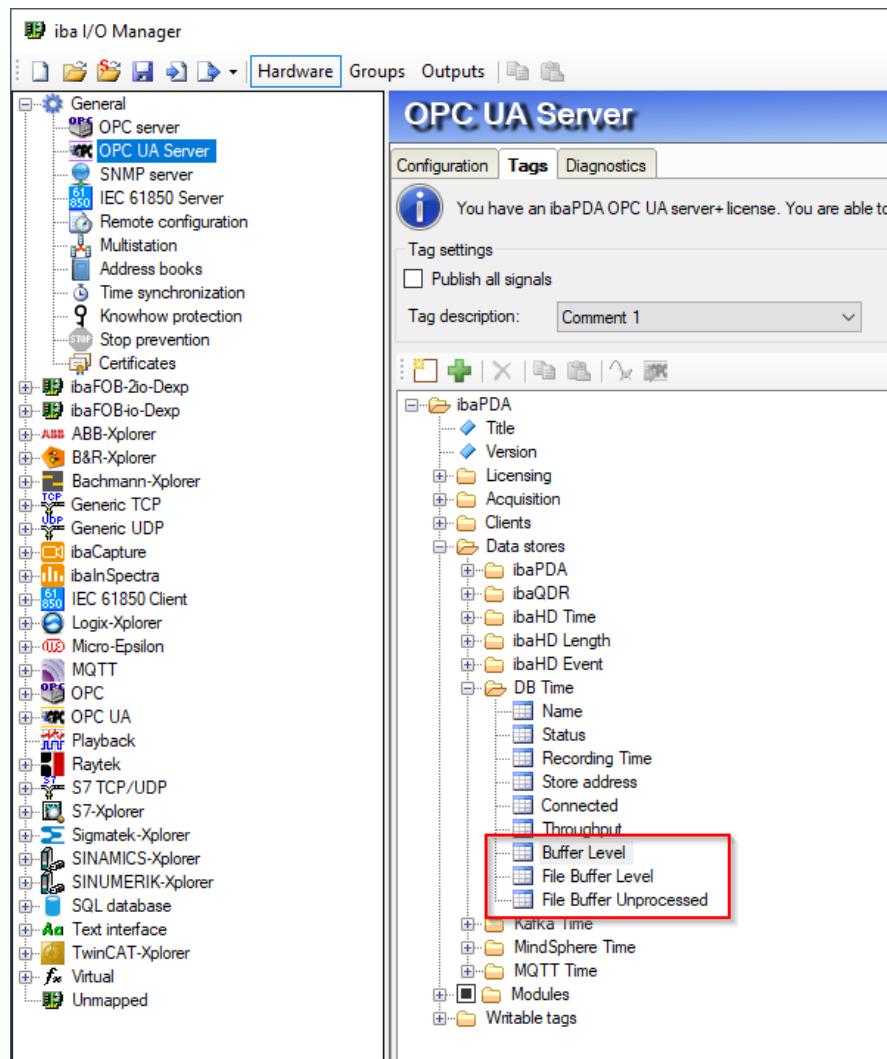
The following functions now also show information about the buffer usage:

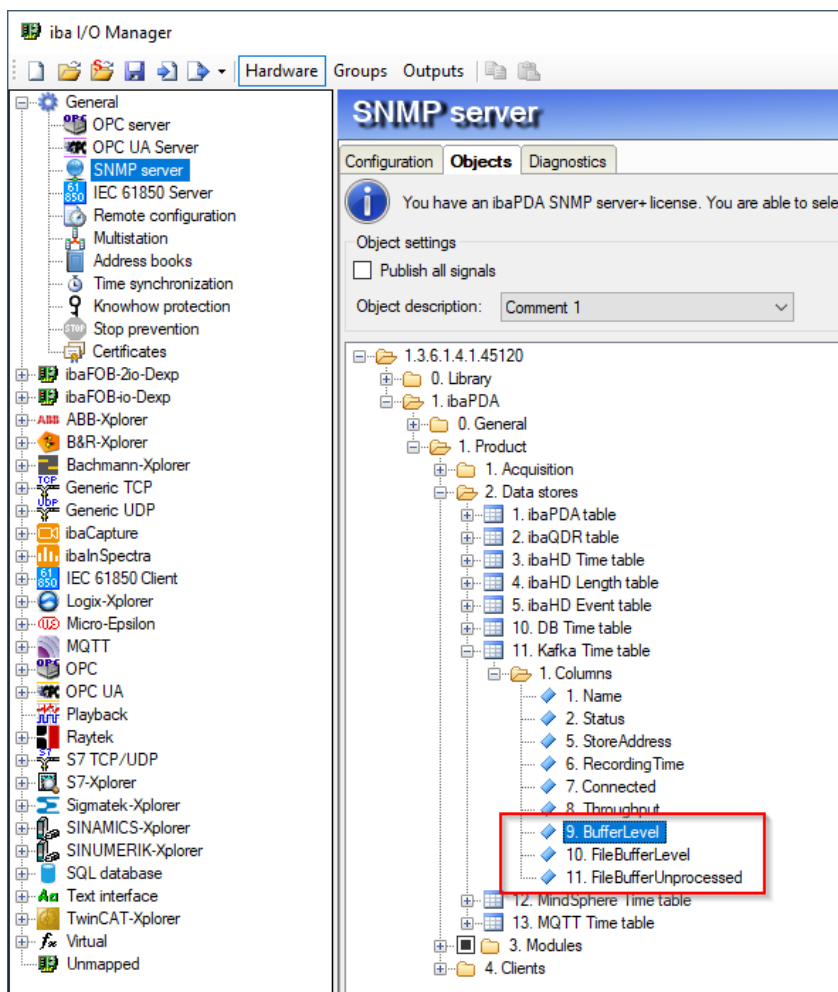
- DataStoreInfoHD
- DataStoreInfoDB
- DataStoreInfoKafka
- DataStoreInfoMindSphere
- DataStoreInfoMQTT



5.2.3 OPC UA Server and SNMP

The same diagnostics information is also available in the OPC UA and SNMP server:



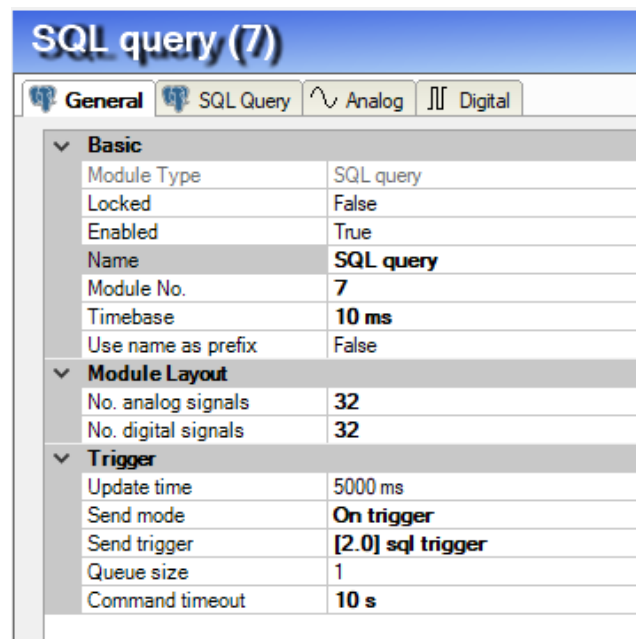


5.3 Updating the data store configuration

The data store configuration can be updated when the acquisition is running. The configuration has an influence on the buffered data. Some changes in the configuration can cause the data that is currently in the buffer to become invalid. For example, when a signal is added to a DB data store, the currently buffered data will become invalid, because its buffer relies on the table and columns to remain the same during acquisition.

If an incompatible change is made to the data store configuration, buffer files can be deleted. When buffer files are lost, the data that was contained in the buffer can no longer be retrieved. Please be warned that buffer files can potentially contain gigabytes of data.

6 File buffer for SQL output module



For SQL queries and SQL commands, there is a queue size. The queue size determines how many statements are preserved if they can't be executed, e.g. because a database is not available. The Queue size is by default 1 for SQL queries and 1000 for SQL commands.

Originally, a queue of statements that still have to be executed was always stored in memory.

SQL command modules ("Outputs" tab in the I/O Manager) now have an additional tab "Buffer" for extended buffer configuration including the queue size. For the input modules (SQL queries) this new functionality is not applicable.

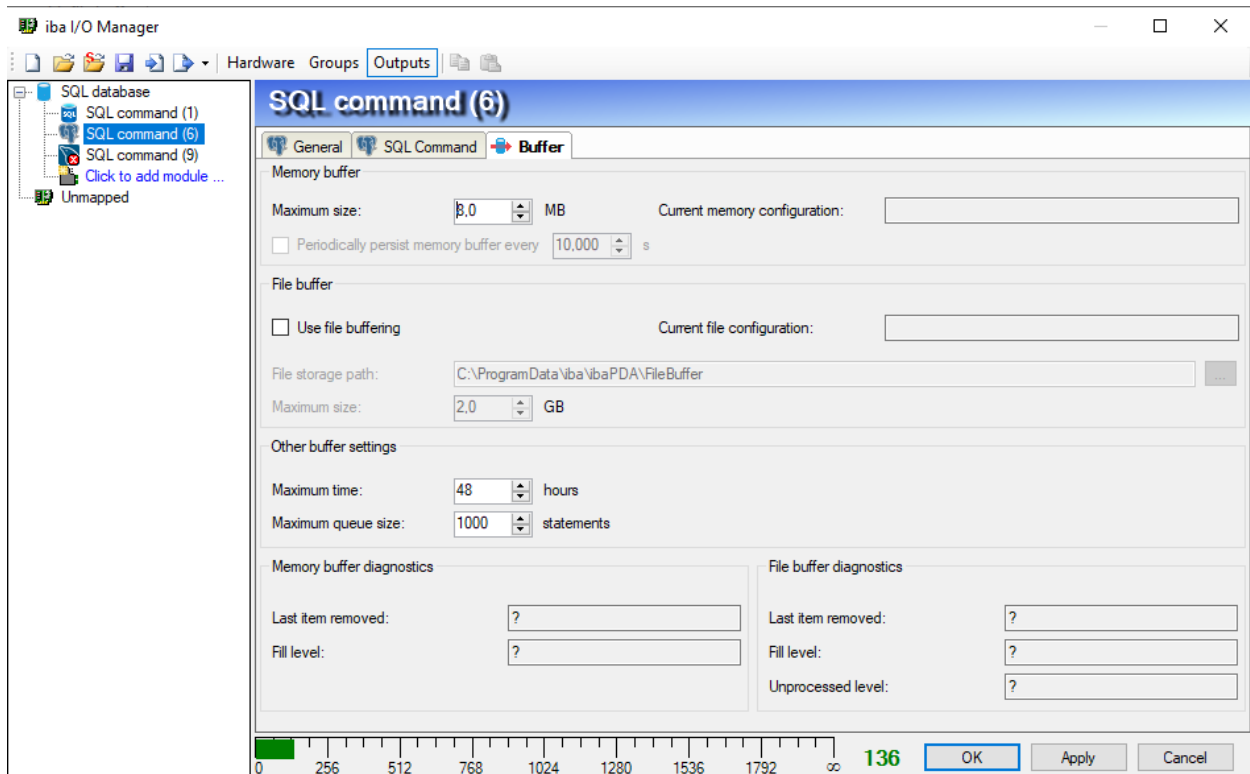
For SQL commands the queue can now have an arbitrary size in memory or in the file system of the ibaPDA server. The queue is now implemented just like a data store buffer, as explained in the previous chapter.

6.1 Configuration

For input modules (SQL queries), there is no configuration for the buffer available. The file buffer is always disabled. Only the queue size can be changed in the properties of the "General" tab.

In general, a queue size of 1 is recommended. It rarely makes sense to select more than 1 queries at a time.

For SQL outputs (SQL commands), the same configuration as for the timebased data store file buffers is available. The image below shows the default configuration for the buffers for SQL command modules.



The maximum queue size (originally configurable as a property on the “General” tab) is integrated now on the “Buffer” tab.

That results in the following ways to limit the buffer size:

- by number of statements
- by memory and file size
- by the amount of time that statements already reside in the buffer

6.2 Diagnostics module

The status of a SQL module can be monitored by using a diagnostic module. This diagnostic module was extended and contains now additional signals about the buffering state.

- Buffered statements: the number of statements currently in the buffer.
- Buffered statements lost: the number of statements that are no longer in the buffer, because the buffer size exceeded its limits.
- Buffer memory size (actual, avg, max): the actual, average and maximum memory size of the buffer.
- Buffer file size (actual, avg, max): the actual, average and maximum size of the part of the buffer that is stored on disk.

iba I/O Manager

Hardware Groups Outputs

General

- OPC server
- OPC UA Server
- SNMP server
- IEC 61850 Server
- Remote configuration
- Multistation
- Address books
- Time synchronization
- Knowhow protection
- Stop prevention
- Certificates
- SQL database
 - SQL command Diagnostics (3)
 - SQL query (7)
 - SQL query (8)
 - PG Diagnostics (10)
 - SQL command Diagnostics (11)**
 - Click to add module ...
- Virtual
 - DataStoreInfo (0)
 - ibaQPanel input (2)
 - Virtual (4)
 - Lookup table (5)
 - Click to add module ...
 - Unmapped

SQL command Diagnostics (11)

General Analog Digital

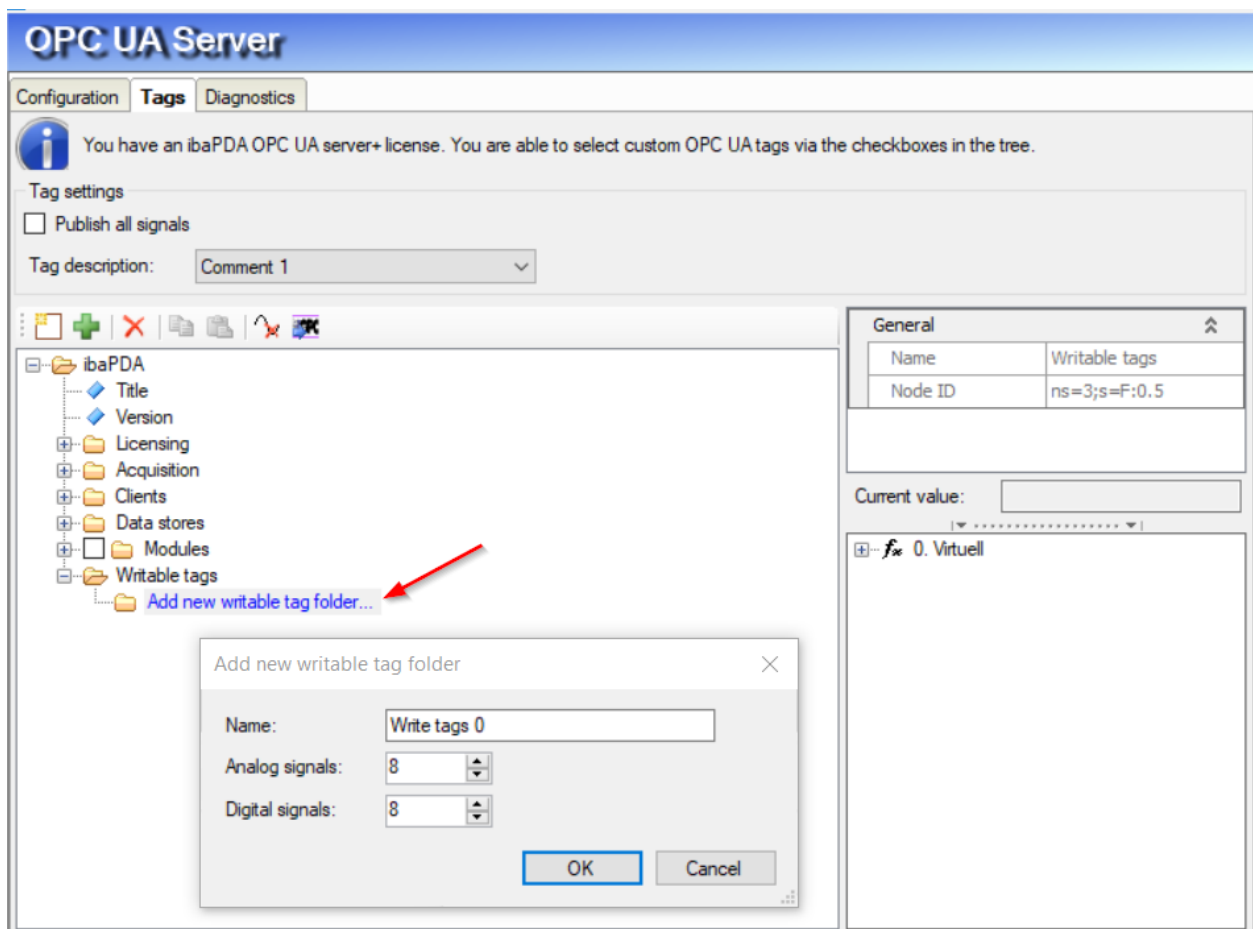
	Name	Unit	Gain	Offset	Active	Actual
0	Error counter		1	0	<input checked="" type="checkbox"/>	
1	Statements processed		1	0	<input checked="" type="checkbox"/>	
2	Rows (last)		1	0	<input checked="" type="checkbox"/>	
3	Rows (maximum)		1	0	<input checked="" type="checkbox"/>	
4	Response time (actual)	ms	1	0	<input checked="" type="checkbox"/>	
5	Response time (average)	ms	1	0	<input checked="" type="checkbox"/>	
6	Response time (min)	ms	1	0	<input checked="" type="checkbox"/>	
7	Response time (max)	ms	1	0	<input checked="" type="checkbox"/>	
8	Buffered statements		1	0	<input checked="" type="checkbox"/>	
9	Buffered statements lost		1	0	<input checked="" type="checkbox"/>	
10	Buffer memory size (actual)	KB	1	0	<input checked="" type="checkbox"/>	
11	Buffer memory size (avg)	KB	1	0	<input checked="" type="checkbox"/>	
12	Buffer memory size (max)	KB	1	0	<input checked="" type="checkbox"/>	
13	Buffer file size (actual)	MB	1	0	<input checked="" type="checkbox"/>	
14	Buffer file size (avg)	MB	1	0	<input checked="" type="checkbox"/>	
15	Buffer file size (max)	MB	1	0	<input checked="" type="checkbox"/>	

0 256 512 768 1024 1280 1536 1792 ∞ **153** OK Apply Cancel

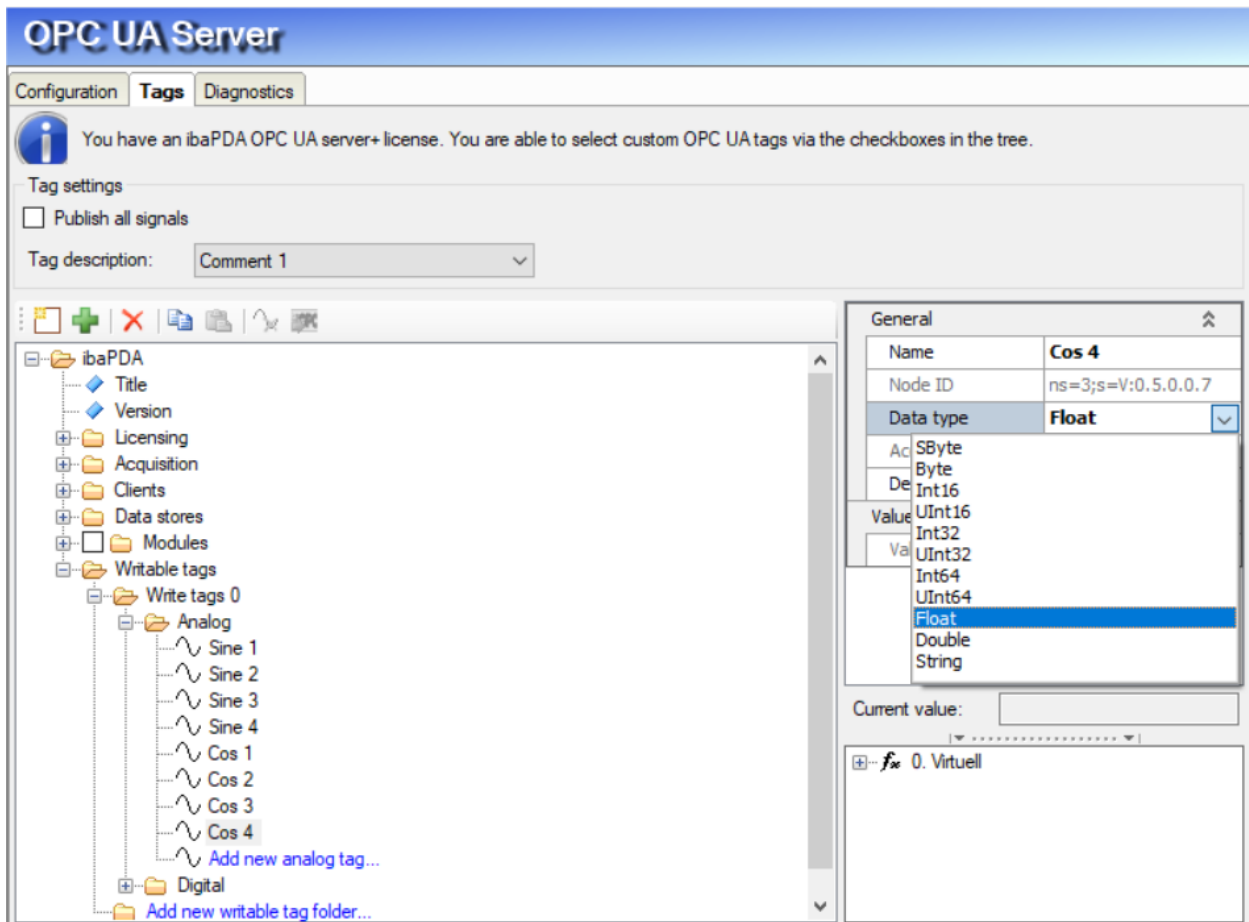
7 OPC UA Server module


Similar to OPC DA Server modules it is now possible to define tags in ibaPDA's OPC UA Server with write access which can then be measured in ibaPDA using an OPC UA Server module. To be able to use this new feature the ibaPDA OPC UA Server+ license needs to be activated on your dongle.

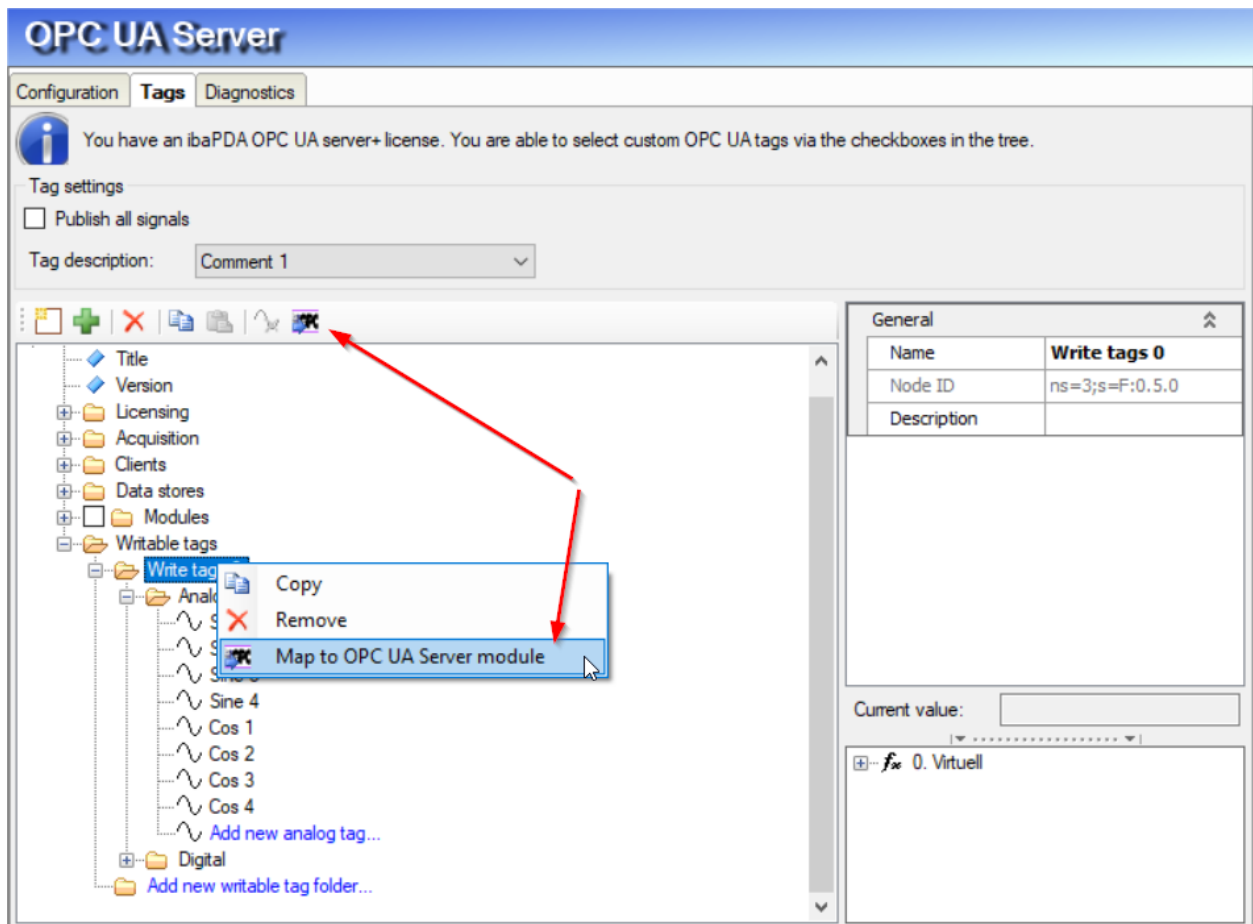
When using classic OPC DA Server modules a module is first created after which the corresponding tags are automatically created in the OPC DA Server when applying the I/O configuration. For the OPC UA Server modules the approach is different: tags first need to be defined in the OPC UA Server after which they can be mapped by the user to an OPC UA Server module. The reason for this decoupling is that with an OPC UA Server module it is possible to measure any tag published by ibaPDA's OPC UA Server; this includes e.g. tags with write access that have been added using the Information Model. An OPC UA Server module is basically an OPC UA Client module with an internal connection to ibaPDA's OPC UA Server.




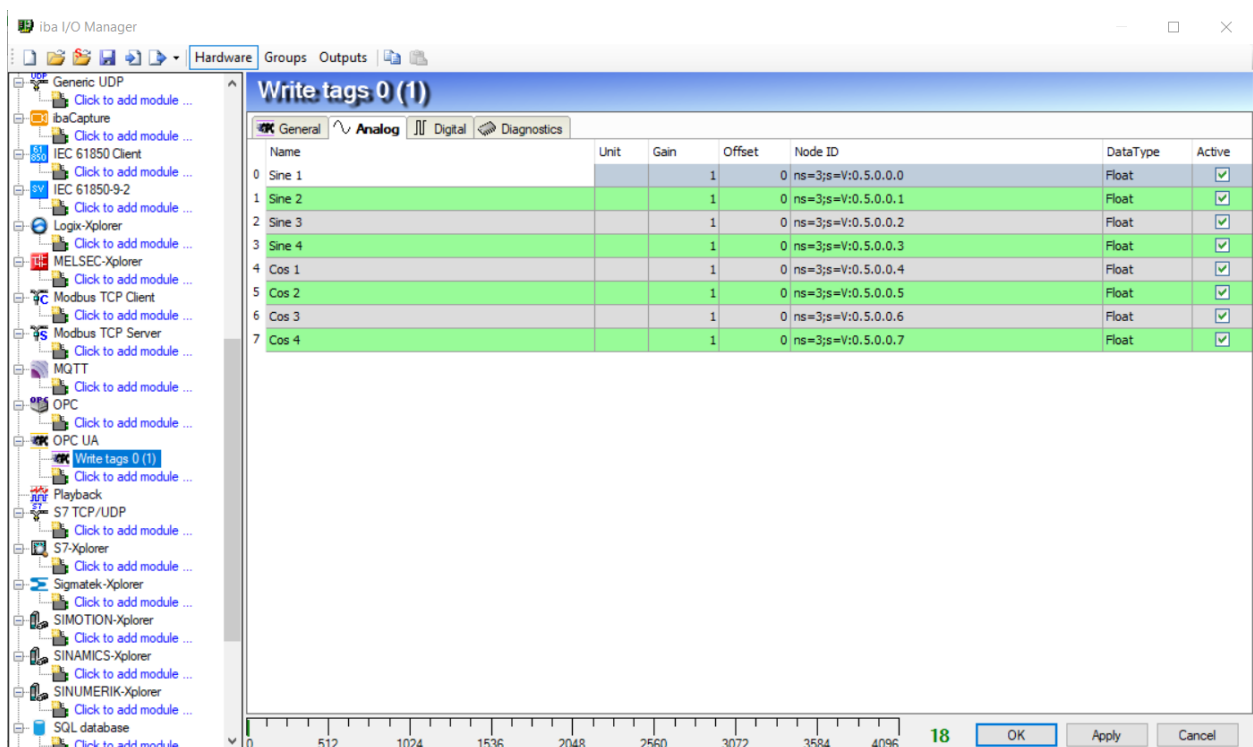
In order to create writable tags open the **I/O Manager** and select the **General > OPC UA Server** node. Expand the **Writable tags** node and click **Add new writable tag folder...**. A dialog will appear asking you to name the new folder (which can then later be mapped to an OPC UA Server module) and define the number of analog and digital signals.



Once the tags have been created you can modify some of their properties using the property grid on the right-hand side. For analog tags you can modify the **Name**, **Description** and **Data type**; for digital tags you can modify the **Name** and **Description** (since the data type is self-evident). Once a writable tag folder has been created, tags can still be added by clicking **Add new analog tag...** or **Add new digital tag...** or using the context menu of the **Analog** or **Digital** folder. Tags can also be removed using the  button in the toolbar, by pressing Delete on your keyboard or using the context menu of the tag.



Once all required tags have been created the writable folder can be mapped to an OPC UA Server module. This can be done using the  button in the toolbar or by using the context menu of the writable tag folder.



The OPC UA Server module containing signals corresponding to the writable tags will be automatically created under the OPC UA interface in the I/O configuration.

Write tags 0 (1)

General Analog Digital Diagnostics

Basic

Module Type	OPC UA Server
Locked	False
Enabled	True
Name	Write tags 0
Module No.	1
Timebase	10 ms
Use name as prefix	False

Module Layout

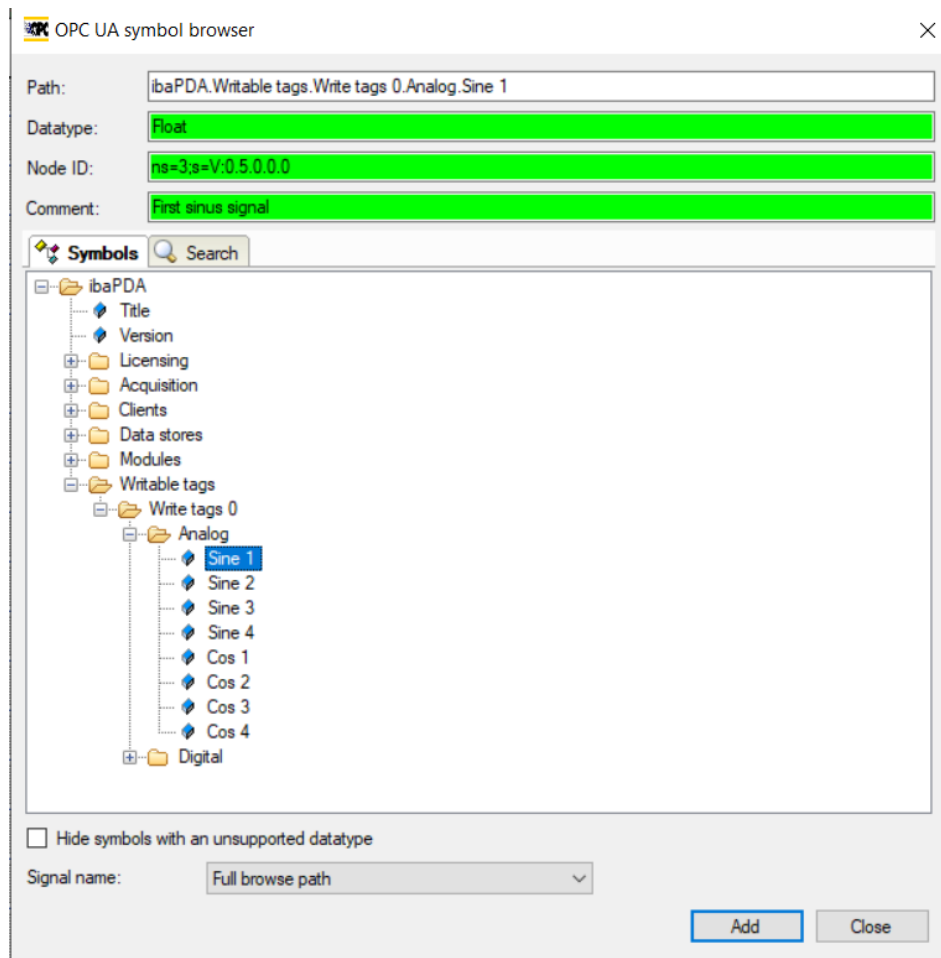
No. analog input signals	8
No. digital input signals	8

Name
The name of the module.

[Select symbols](#)

Contrary to an OPC UA Client module only the basic module properties are present (since communication with the OPC UA Server is internal).

Since an OPC UA Server module is decoupled from the OPC UA Server a module can also be added from scratch (i.e. without mapping from a writable tag folder). This can be useful in case writable tags added using the Information Model need to be measured.



Using the **Select symbols** link in the module's **General** tab an address book of ibaPDA's OPC UA Server is displayed with which signals can be added analogously to a OPC UA Client module.

8 Encrypted client-server communication

In ibaPDA v7.3.0 the communication between client and server can be encrypted. There are 2 connections between client and server:

1. Main connection that is used to transfer configurations and diagnostics.
2. Data connection that is used to transfer signal data when the acquisition is running.

In the server access configuration, you can configure whether the connections should be encrypted or not. Use the menu “Configure\Server access” to open the server access configuration.

Name	Version	IP address	Connected since	Requested signals	Licenses
★ DEVPC-NICS\pic 15	7.3.0	127.0.0.1	2/03/2021 9:30:43	0	ibaQPanel
DEVPC-NICS\pic 19	7.3.0	127.0.0.1	2/03/2021 9:53:45	0	Client
2012-SERVER\pic 20	7.1.7	192.168.123.103	2/03/2021 9:59:26	0	Client

In the client connections part you see the list of currently connected clients. For the main connection you can choose between 2 options:

1. Allow both encrypted and unencrypted connections.
2. Allow only encrypted connections.




ibaPDA client versions older than 7.3.0 will always connect unencrypted. So if you want to allow these older client versions to connect to your ibaPDA server you should select option 1. ibaPDA client versions 7.3.0 or later will connect encrypted to ibaPDA servers with version 7.3.0 or later and they will connect unencrypted to ibaPDA servers with versions older than 7.3.0.

For the data connection you can choose between 3 options:

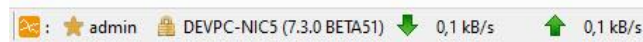
1. Allow both encrypted and unencrypted connections.
2. Allow only encrypted connections.
3. Allow only unencrypted connections.

Option 1 will use an encrypted data connection when the main connection is encrypted and an unencrypted data connection when the main connection is unencrypted. An encrypted data connection is only possible when the main connection is encrypted. So option 2 can only be chosen when the main connection also only allows encrypted connections. When option 3 is selected then the data connection will always be unencrypted independent of the type of main connection. An encrypted data connection requires more CPU power especially when a lot of data is transferred. That is why the default selection is option 3.

The icon in the “IP address” of the client connections table shows the type of the connection. There are 3 possibilities:

1. : Both main and data connection are encrypted.
2. : Main connection is encrypted and data connection is unencrypted.
3. : Both main and data connection are unencrypted.

These same icons are also shown in the status bar of the ibaPDA client.



9 Central certificate store

9.1 Overview

Certificates are required for SSL/TLS communication and can be used for secure authentication. With this new feature, a central place for management of certificates used in ibaPDA is provided.

Some interfaces, like the e-mail outputs, still use Windows certificates. Other parts of ibaPDA, like OPC UA Server and the MQTT data store use certificates from the ibaPDA central certificate store.

The global list of certificates is presented in a grid. It is available on a separate certificates node under the general node in the I/O manager and as a certificates node in the data storage manager. The grid contains one row per certificate.

The screenshots show the 'Certificates' grid in two different contexts: the I/O Manager and the Data Storage Manager. Both interfaces display a table of certificates with the following columns: Name, Properties, Issued By, Expiration Date, Used By, and Thumbprint.

Name	Properties	Issued By	Expiration Date	Used By	Thumbprint
ibaPDA 001@D	✓	ibaPDA 001@D	13/01/2031 ...		5766F54A...
ibaPDA 002@D	✓	ibaPDA 002@D	13/01/2031 ...		37320C40...
ibaPDA 003@D	✓	ibaPDA 003@D	13/01/2031 ...	OPC UA Server	AE1A4C3...
ibaPDA 004@DESKTOP-2ATLR1Q	✓	ibaPDA 004@DESKTOP-...	13/01/2031 ...		FC98487F...
ibaPDA 005@D	✓	ibaPDA 005@D	13/01/2031 ...		8DB092C...
ibaPDA@127.0.0.1	✓	ibaPDA@127.0.0.1	5/02/2031 1...		0F5D43E6...
blabla.com	✓	mosquitto.org	25/05/2021 ...		3DAD4EB...
OpenSSL Test Intermediate CA	✓	OpenSSL Test Root CA	14/06/2118 ...		6AFFDE8...

The Name column shows the name of the certificate. Multiple certificates can share the same name. Only the thumbprint is unique for a certificate.

9.2 Managing certificates

Certificates can be managed in the certificates control in either the I/O manager or the data storage manager.


















To be able to manage certificates, a user must have one of the following rights:

- | | |
|-----------------------------|---------|
| Apply new I/O configuration | Granted |
|-----------------------------|---------|
- | | |
|--------------------------------------|---------|
| Apply new data storage configuration | Granted |
|--------------------------------------|---------|

Even if “Apply new I/O configuration” is granted and “Apply new data storage configuration” isn’t, it is possible to manage certificates in both the I/O manager and the data storage

manager. Also if the data storage configuration can be applied and the I/O configuration can't, it is still possible to manage certificates from both places.

Although the certificates can be managed in 2 places, there is only one central certificate store in ibaPDA.

Certificates				
Name	Properties	Issued By	Expiration Date	Used By
ibaPDA 001@D	  	ibaPDA 001@D	13/01/2031 9:38:17	
ibaPDA 002@D	 	ibaPDA 002@D	13/01/2031 9:38:30	
ibaPDA 003@D	 	ibaPDA 003@D	13/01/2031 9:39:14	OPC UA Server
ibaPDA 004@DESKTOP-2AT...	  	ibaPDA 004@DESKTOP-2AT...	13/01/2031 10:23:23	
ibaPDA 005@D	  	ibaPDA 005@D	13/01/2031 10:41:51	
ibaPDA@127.0.0.1	 	ibaPDA@127.0.0.1	5/02/2031 16:18:36	
blabla.com		mosquitto.org	25/05/2021 11:16:03	
OpenSSL Test Intermediate ...		OpenSSL Test Root CA	14/06/2118 14:46:28	

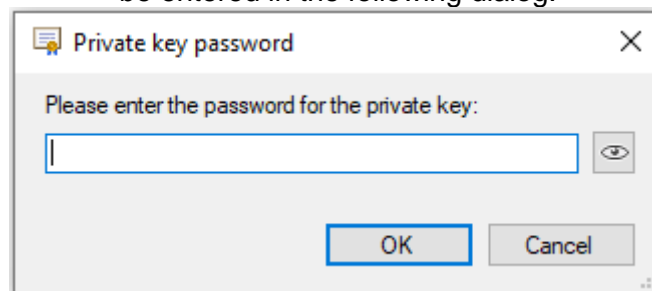
9.2.1 Toolbar buttons

The following actions can be performed with the buttons in the toolbar:




Add a certificate from a certificate file. When this button is clicked, an open file dialog is shown that you can use to select a certificate file. Various certificate file formats are supported. If your certificate is in a file with an unrecognized file extension, try to import it anyway by selecting the all files (*.*) filter. In most cases, that will work.

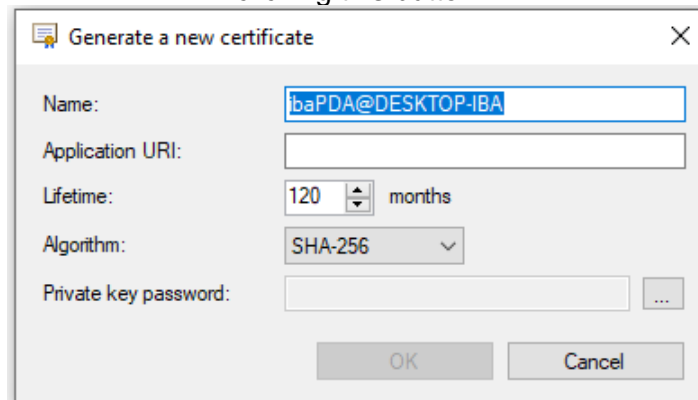
If the certificate you want to import contains a private key, the password for the private key must be entered in the following dialog:



If the wrong password is entered, the certificate is not added. If a certificate with the same thumbprint already exists in the certificate store, it is replaced if the password is correct.

For certificate files that don't contain a private key, no password must be entered.

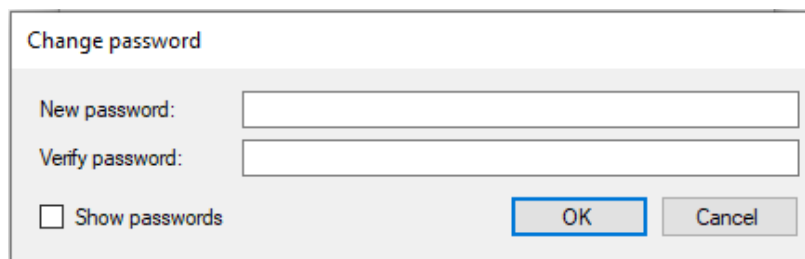
 Let ibaPDA generate a new self-signed certificate for you. The following form is shown when clicking this button:



The dialog box titled "Generate a new certificate" contains the following fields and controls:

- Name:
- Application URI:
- Lifetime: months
- Algorithm:
- Private key password: ...
- Buttons: OK, Cancel

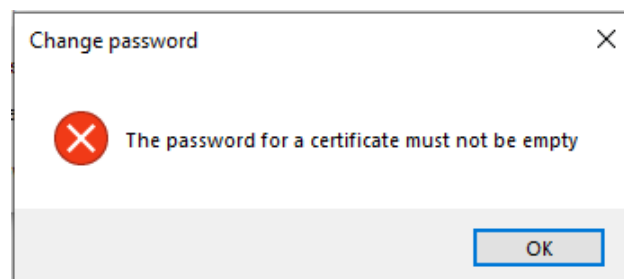
The application URI is optional. All the other fields must be filled in. As long as no password is supplied, the OK button will be disabled. A password can be chosen by clicking the ellipsis (...) button. A form to enter a password then appears.



The dialog box titled "Change password" contains the following fields and controls:

- New password:
- Verify password:
- ☐ Show passwords
- Buttons: OK, Cancel


The password must not be empty. When clicking OK while the password is empty, you get the following error message.



The error message dialog box titled "Change password" displays the following:

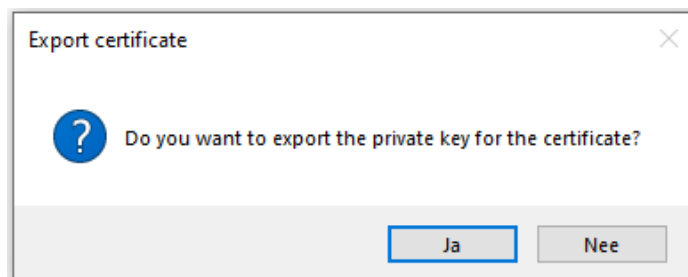
- Icon: Red circle with a white X
- Text: The password for a certificate must not be empty
- Buttons: OK

There are no further requirements for the password. You are responsible to keep the password entered here in a safe place, so that the generated self-signed certificate can be exported for use in Windows or other applications.

 Export the selected certificate to a certificate file. The certificate file can be used for Windows or other applications. It can also be used to re-import a certificate in ibaPDA, whether it is the same instance or another one.

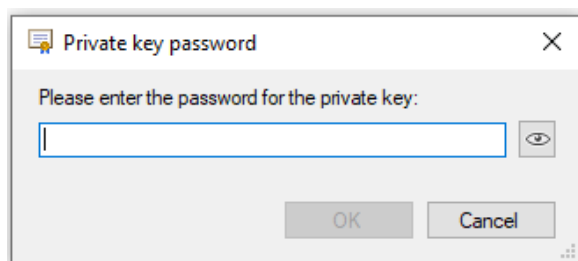
Exporting a certificate without private key is straightforward. A save file dialog is presented where you can choose the name and the location of the file where you want to store the certificate file.

If a certificate file is accompanied with a private key, there are more options. First of all, you can choose whether the private key should also be exported.

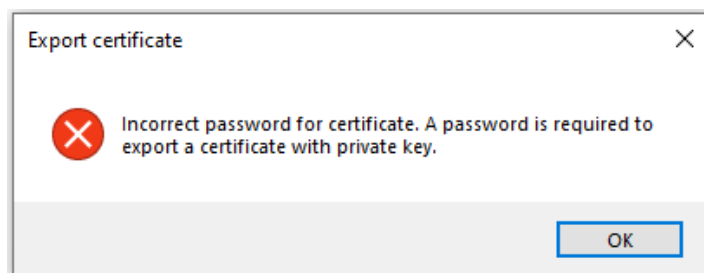


If not, you are presented with a save file dialog just like when a certificate without private key is exported.

If you want to export the private key as well, a dialog is presented where you need to enter the password for the private key.

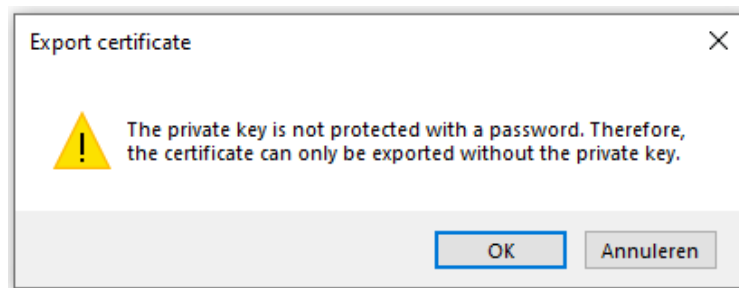


If you enter the wrong password, the certificate will not be exported.




The correct password is the password that was originally used to import or to generate the certificate in ibaPDA. If that password is entered, the certificate can be exported as a pfx file. That file is password-protected and contains the certificate and the private key.

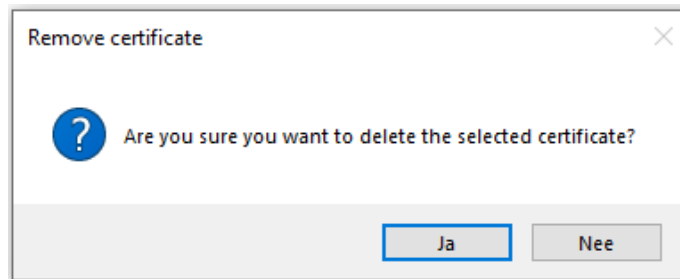
A certificate with private key can be stored in ibaPDA, while the private key is not protected with a password. Especially when the Certificates/settings.xml file is updated from an older version of ibaPDA or if the certificate is imported from an external file that can occur. In that case, it's not possible to export the certificate including the private key. The following warning will then be shown:





If you click OK, you can continue with the export of the certificate, but without private key.


If export of the private key is required anyway, we encourage you to generate a new certificate and private key instead. Your PKI infrastructure may be able to generate a new certificate file which contains a password protected private key. If that's not possible and you definitely need to export the private key stored in ibaPDA, please contact iba support.

 To delete the selected certificate. You have to confirm this decision before the certificate is actually deleted.



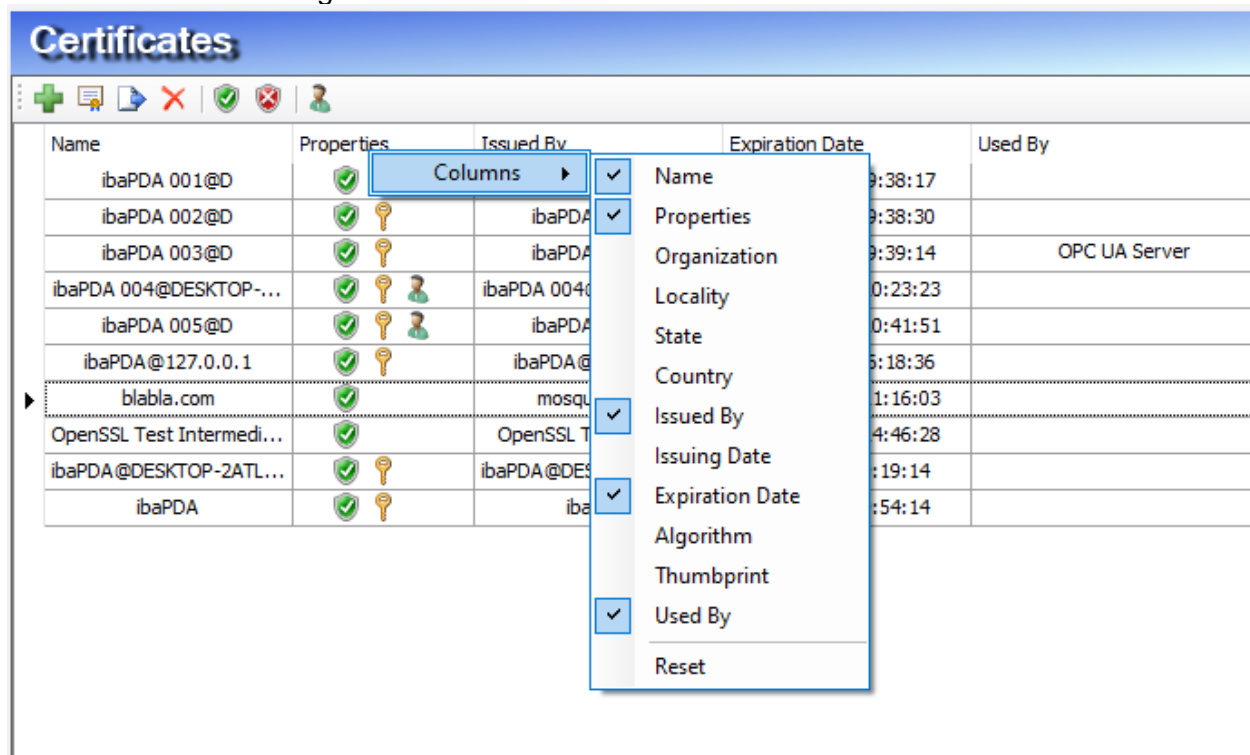
 Trust a certificate.

 Untrust a certificate. Certificates that are not trusted will not appear in the combo boxes where certificates can be chosen for a specific purpose.





 Allow this certificate to be used for authentication of OPC UA. Please refer to the OPC UA documentation for more information.

9.2.2 Columns

The columns shown in the certificates control can be selected by right clicking on a column name in the certificates grid.

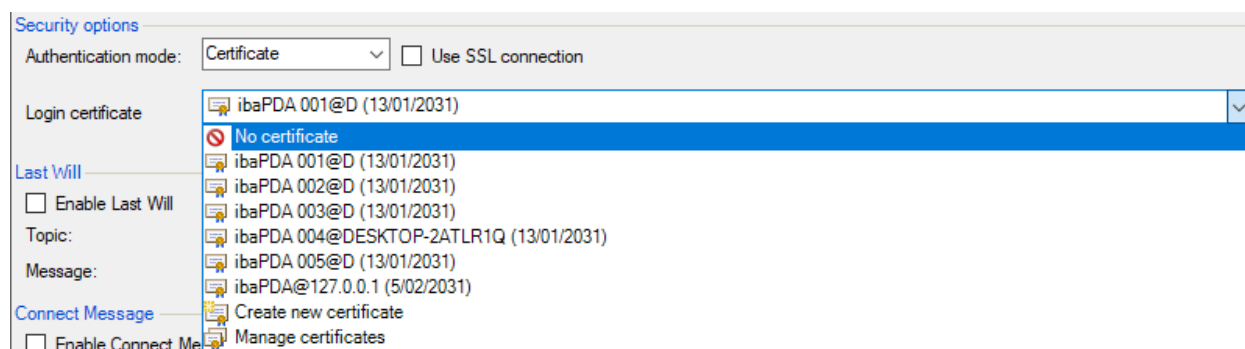


More properties of the certificates can be shown like that. “Reset” restores the columns to the defaults. Most of the columns show properties of the certificates. Some require a bit more explanation.

- Name: The common name of the certificate
- Properties:
 -  : the certificate is trusted in ibaPDA, as long as it is not expired
 -  : the certificate is not trusted in ibaPDA
 -  : the certificate is accompanied with a private key
 -  : the certificate can be used for user authentication in ibaPDA's OPC UA implementation
- Thumbprint: uniquely identifies the certificate
- Used by: this column shows where the certificate is used. The values shown here are a combination of the open manager (either I/O manager or data storage manager) and the other one. For the open manager, the values in this column refer to the configuration that is being edited. You can double click on a line in the used by column to navigate to where the certificate is being used. The navigation with double click is only available in the open manager form. In the example above, “OPC UA Server” is displayed. Double clicking on that will navigate to the OPC UA Server node if the I/O manager is open, but not if the data storage manager is the active manager form. For the manager that is not open, the values that are active in the ibaPDA server are displayed. Double clicking on those values has no effect.

9.3 Using a certificate

In places where a certificate can be used, you will find a certificates combobox. The certificates combobox is a dropdown that contains the certificates that can be used. For example, in the MQTT time based data store, a certificate can be used for client authentication. The selection of the certificates looks like this:



- No certificate: don't use a certificate here. That will usually cause validation to fail.
- The items under “No certificate” are certificates that can be selected here. The listed certificates are valid and are suitable for the place where you can find the certificates combobox.
- There is an option to create a new certificate. If the creation succeeds, the newly created certificate is selected. If not, “No certificate” is selected. That is to avoid confusion about which certificate is actually in use.

- “Manage certificates” brings you back to the central certificate store, either in the data storage manager or in the I/O manager, depending on which of the two is open.

To know what a specific certificate is used for, please refer to the documentation of the module or data store where the certificate can be selected.

The selection of a particular certificate is stored in the registry of the computer running the ibaPDA client. In a new configuration, the same certificate is selected if no selection was made yet.

9.4 Storage and protection of certificates

Certificates are stored in a file named settings.xml, in the Certificates folder of the ibaPDA server installation. If such a file was already present from a previous installation that used certificates for OPC UA, the file will be replaced with a new version. In the new version, the certificates are encrypted for more security.

Private keys are key to your or your organization’s digital identity. Measures are taken to protect your identity when using certificates with private keys in ibaPDA: More specifically, the following features help prevent that a certificate used in ibaPDA can be exported for use in Windows or other applications.

- Certificates are always stored in an encrypted form
- A password is always required for certificates with a private key:
 - When generating a certificate
 - When exporting a certificate with private key
 - When importing a certificate with private key
- A certificate can only be exported if there is a password for the private key. Private keys of which there is no password or the password is unknown can unfortunately no longer be exported. Please keep your passwords in a safe place (e.g. a KeePass file). An exported certificate can be used for Windows or other applications.
- The password of a private key cannot be changed by ibaPDA. That ability will also not be added in the future.
- A password is never required to use a certificate within ibaPDA. The Certificates\settings.xml file can also be moved from one ibaPDA installation to another. No passwords are required to install certificates in that manner.

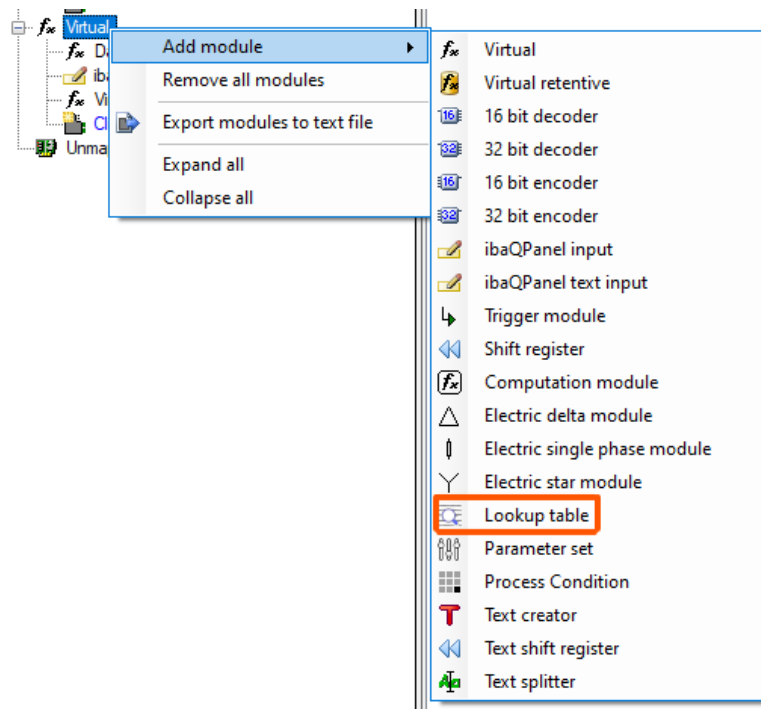
If someone gets your private key, it can be used to impersonate you, to view network traffic that is intended for you and many more exploits are possible.


It is possible that a certificate with a private key exists in ibaPDA, while that private key is not protected with a password. The private key cannot be exported for use in other applications in that case.

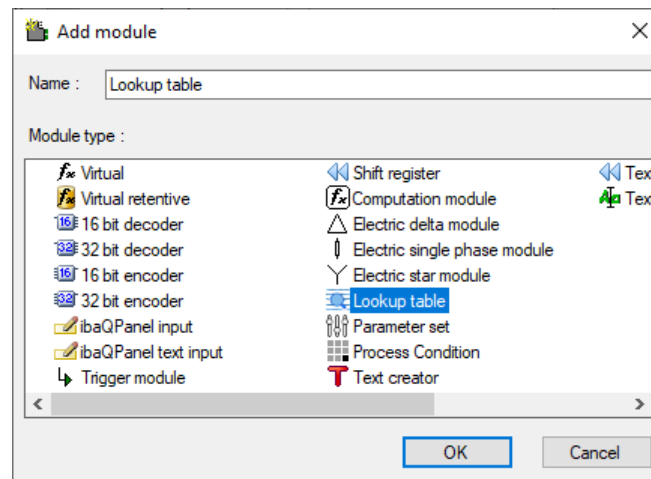
10 Lookup table module

There is a new module named “Lookup table” in the virtual interface. The goal of the lookup table is to map codes to more conceivable values. For example, to map error codes from a device to meaningful error messages.

A lookup table module can be added by right clicking the Virtual interface, open *Add module* and select the lookup table.



You can also click on  [Click to add module ...](#) , then select the lookup table and click OK



10.1 General tab

The following properties are available on the “General” tab:

Lookup table (5)

General Analog

Basic

Module Type	Lookup table
Locked	False
Enabled	True
Name	Lookup table
Module No.	5
Timebase	10 ms
Use name as prefix	False

Lookup tables

Table profile	
---------------	--

Module Layout

No. analog signals	10
--------------------	----

The *Table profile* is specific for this module. Choose a profile from the dropdown. If no profile is chosen, the module is not valid.

Lookup tables

Table profile	profile 1
---------------	-----------

Module Layout

No. analog signals	profile 1 profile 2 <Add lookup table profile>
--------------------	--

More details about profiles can be found in the section below.

10.2 Signals

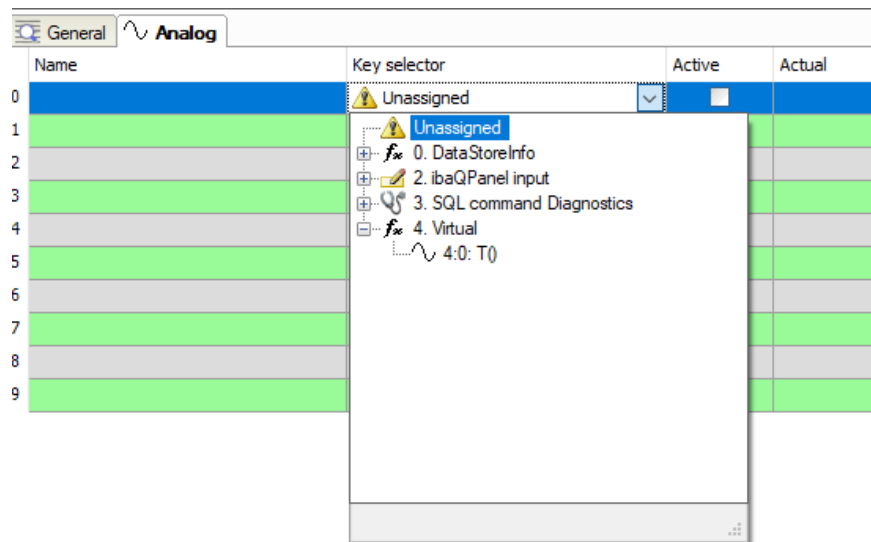
Lookup table (5)

General Analog

	Name	Key selector	Active	Actual
0		Unassigned	<input type="checkbox"/>	
1		Unassigned	<input type="checkbox"/>	
2		Unassigned	<input type="checkbox"/>	
3		Unassigned	<input type="checkbox"/>	
4		Unassigned	<input type="checkbox"/>	
5		Unassigned	<input type="checkbox"/>	
6		Unassigned	<input type="checkbox"/>	
7		Unassigned	<input type="checkbox"/>	
8		Unassigned	<input type="checkbox"/>	
9		Unassigned	<input type="checkbox"/>	

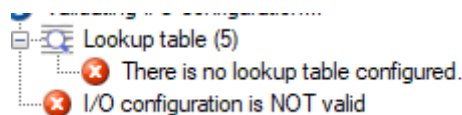
A Lookup table module has only analog signals. All the signals use the same lookup table profile. The key selector in a signal defines a signal whose value is evaluated. The value of the key selector is used as the key in a lookup table profile. The value by applying the profile to the key becomes the value of the signal.

A key selector can be assigned with the dropdown in the key selector column.



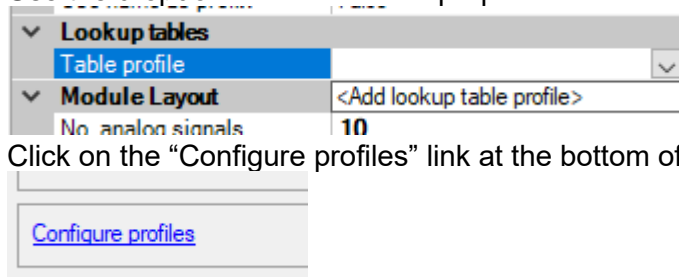
10.3 Lookup table profiles

If no profile is selected, the lookup table will not work, resulting in the following error during validation:



A profile for the Lookup table can be created in the following ways:

- Use the dropdown in the module properties and select "<Add lookup table profile>"



- Click on the "Configure profiles" link at the bottom of the General tab






Both actions open the profile editor form. When using the dropdown, a profile will already be created in the form. Clicking the "Configure profiles" link only opens the form, without adding a new profile.

10.3.1 Profile editor form

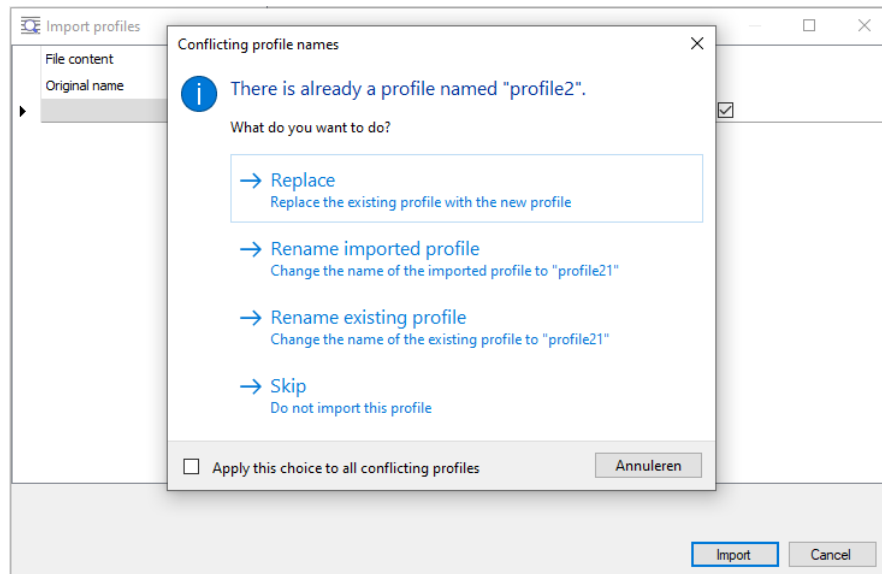
The screenshot shows the 'Lookup tables' window. On the left, a list box titled 'Profiles' contains 'profile1'. On the right, the 'Key type is:' dropdown is set to 'Numeric', and the 'Value for unknown key:' field is empty. Below these, a table with 'Key' and 'Value' columns is shown, with a '*' symbol in the 'Key' column. A toolbar at the bottom left contains icons for adding, copying, deleting, saving, and importing profiles. The bottom right has 'OK' and 'Cancel' buttons.

On the left-hand side of the form, there is a list of lookup table profiles. On the right-hand side the selected profile can be edited.

The list of profiles can be edited as follows:

-  to add a new profile
-  to make a copy of the selected profile
-  to delete the selected profile from the list
-  to save a profile to a file.
-  to import a previously exported profile and place it back into the list of profiles.

Different profiles must have different names in each instance of ibaPDA. If you want to import a profile with the same name as one that already exists in the profile list, you get the following question:



10.3.2 Editing a profile

10.3.2.1 Key type

The key type for a lookup table can be either numeric or text

Key type is: Numeric
 Value for unknown key: Text

If the key type is numeric, only numbers can be entered as key values

Key type is: Numeric
 Value for unknown key:

Key	Value
I ✖	abc

10.3.2.2 Value for unknown key

Value for unknown key: (I don't know)



That is the signal value for key selector values that don't correspond to a key in the profile. For example, if the key selector signal has value "pda" and the key "pda" is not defined, then the signal value will be "(I don't know)".

10.3.2.3 Keys and values

The keys that can be matched against a signal value and the corresponding values can be entered in the middle of the right part of the form.

Key	Value
▶ 0	
*	

Values can be entered manually, by typing them.

If the values are already available in a text file or in Excel, they can be copied from there and pasted with the  button on the right. The other way around, the key and value pairs in the profile can be copied to the clipboard with the  button. The values from the profile can then be pasted as text or into Excel.

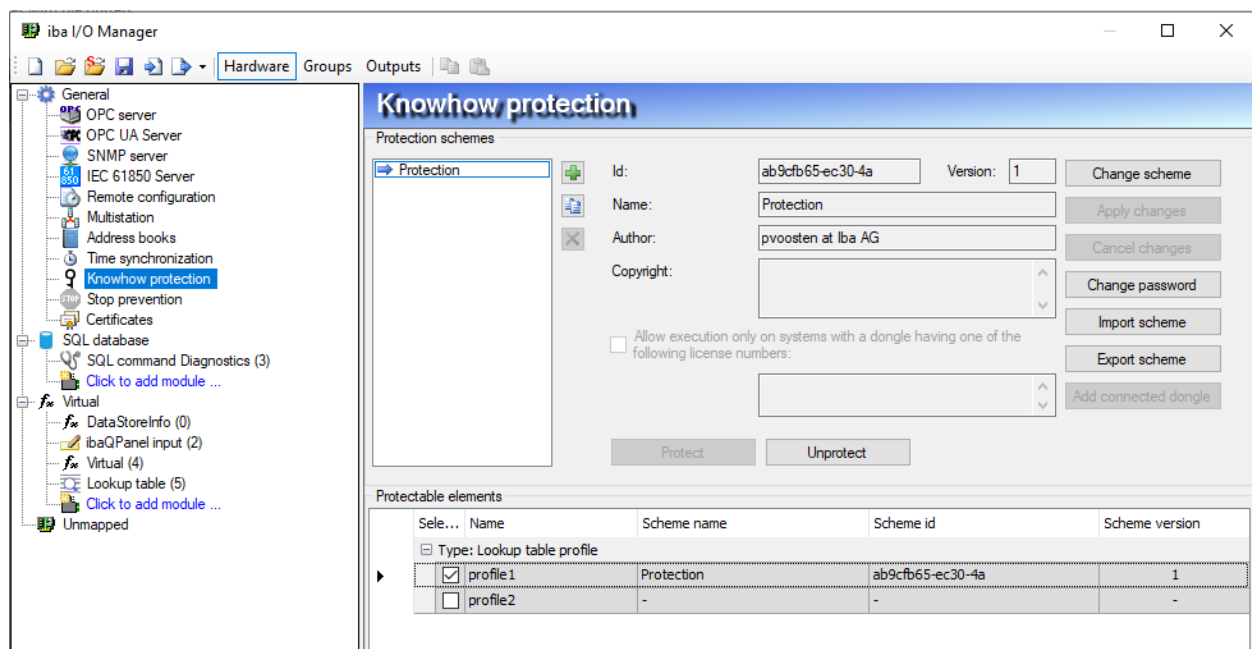
With the remaining buttons on the right, a key-value pair can be moved up or down, or it can be deleted.

Moving a key-value pair up or down can make sense when multiple rows have the same key. The first row with the same key will provide the actual value of the signals that use this profile, for the same key.



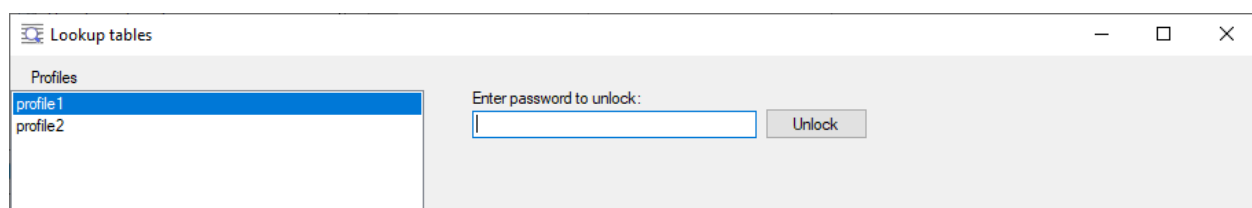
10.3.3 Knowhow protection

As for all profile based module types also a Lookup table profile can be password protected, so that the content is no longer visible in the profile editor form. A description of the “Knowhow protection” feature can be found in the ibaPDA general manual part 2 covering the I/O manager.



In the example above, *profile1* is protected with a password. The knowhow protection affects import, export and visibility of the content of profiles.

The password of the protection scheme is required to edit the profile. The password doesn't need to be entered again until the I/O manager is closed.



When exporting a protected profile, you also need to enter the password of the protection scheme.

Raise protection for export

The following elements are protected by a protection scheme:

- profile 1

Scheme name:

Protection

Scheme id:

7503b41c-50ff-46

Scheme version:

1

Enter password:

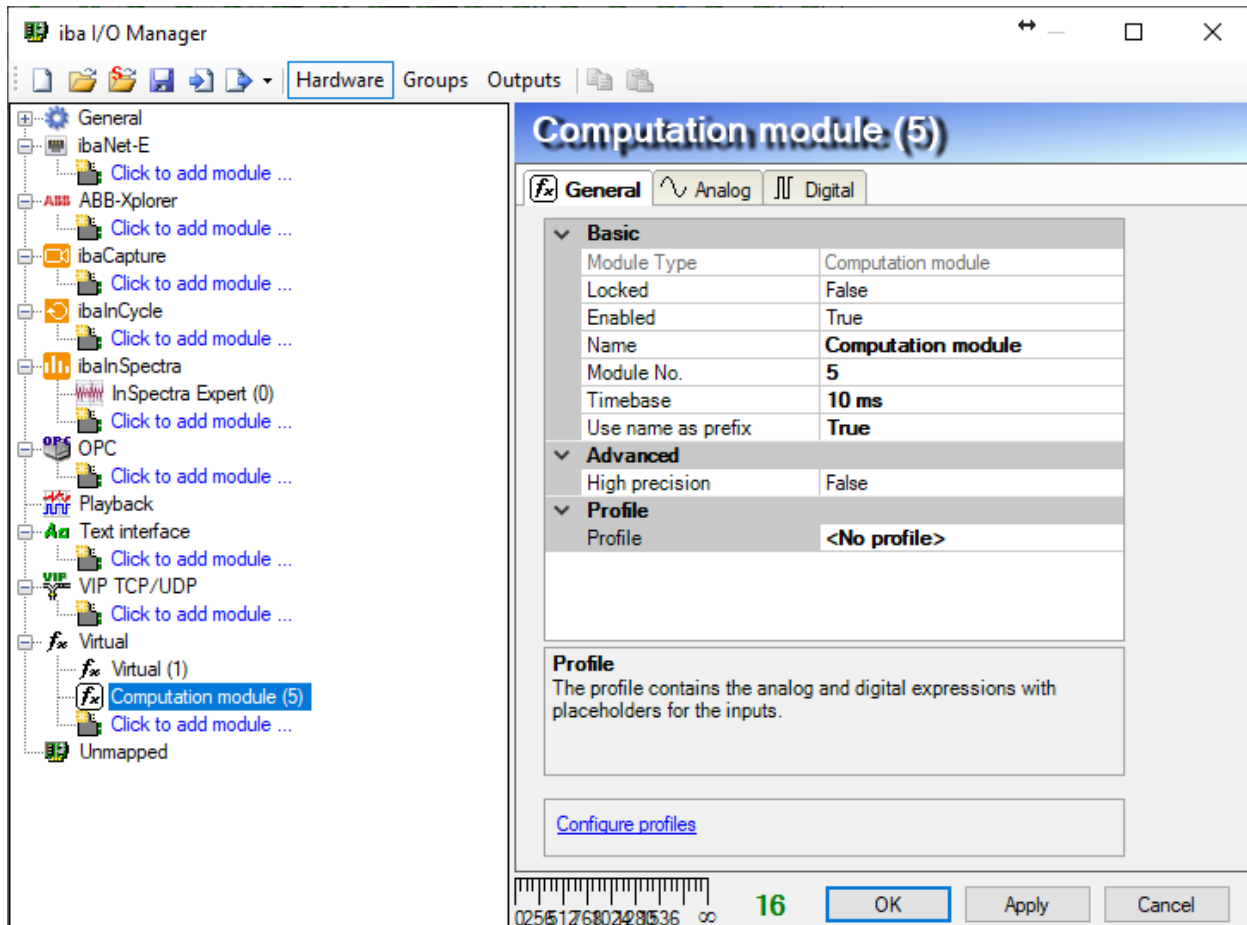
OK

Do not export

Cancel

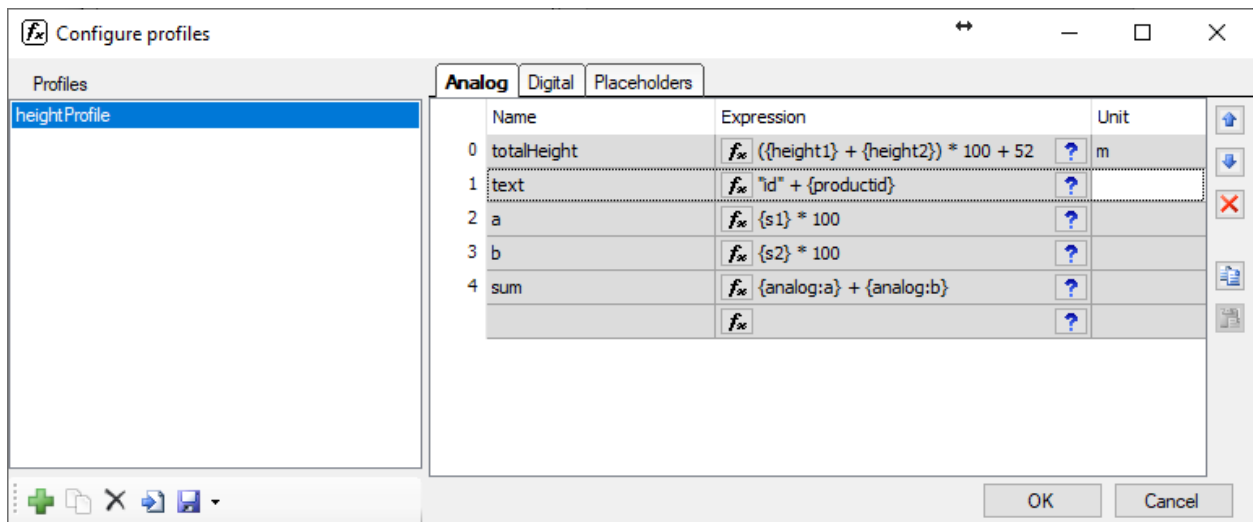
11 Computation module

The computation module is similar to a virtual module, but in a computation module the expressions are saved in a profile that can be reused. To each computation module a computation profile has to be assigned:

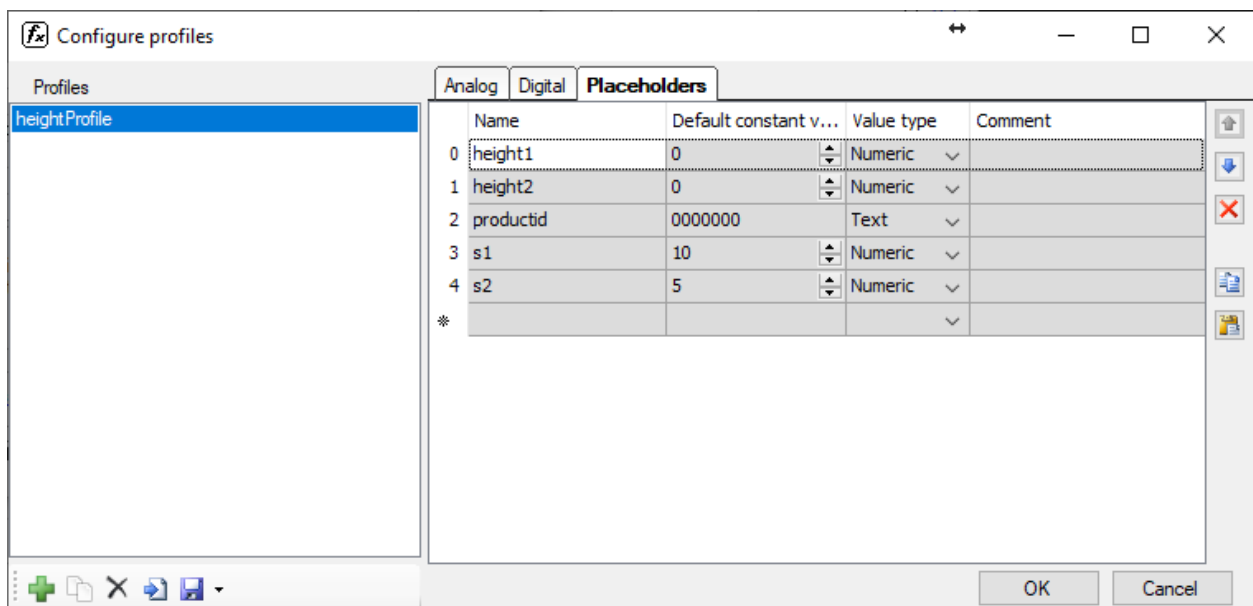


A computation profile consists of analog expressions, digital expressions and placeholders.

In a normal virtual module, you can use signals from other modules directly in the expressions, for instance $([1:0] + [1:1]) * 100 + 52$. In this new computation module, the signals from other modules can be used via placeholders, for instance $(\{height1\} + \{height2\}) * 100 + 52$. You can use the result of one signal in the profile as input for other signals in the same profile. This can be done using the predefined placeholders *analog:X* or *digital:Y*, where X and Y are the name or signal number of the signals you want to refer to.



The placeholders can be configured in the tab "Placeholders" of the profile dialog box.



For each placeholder, one can configure:

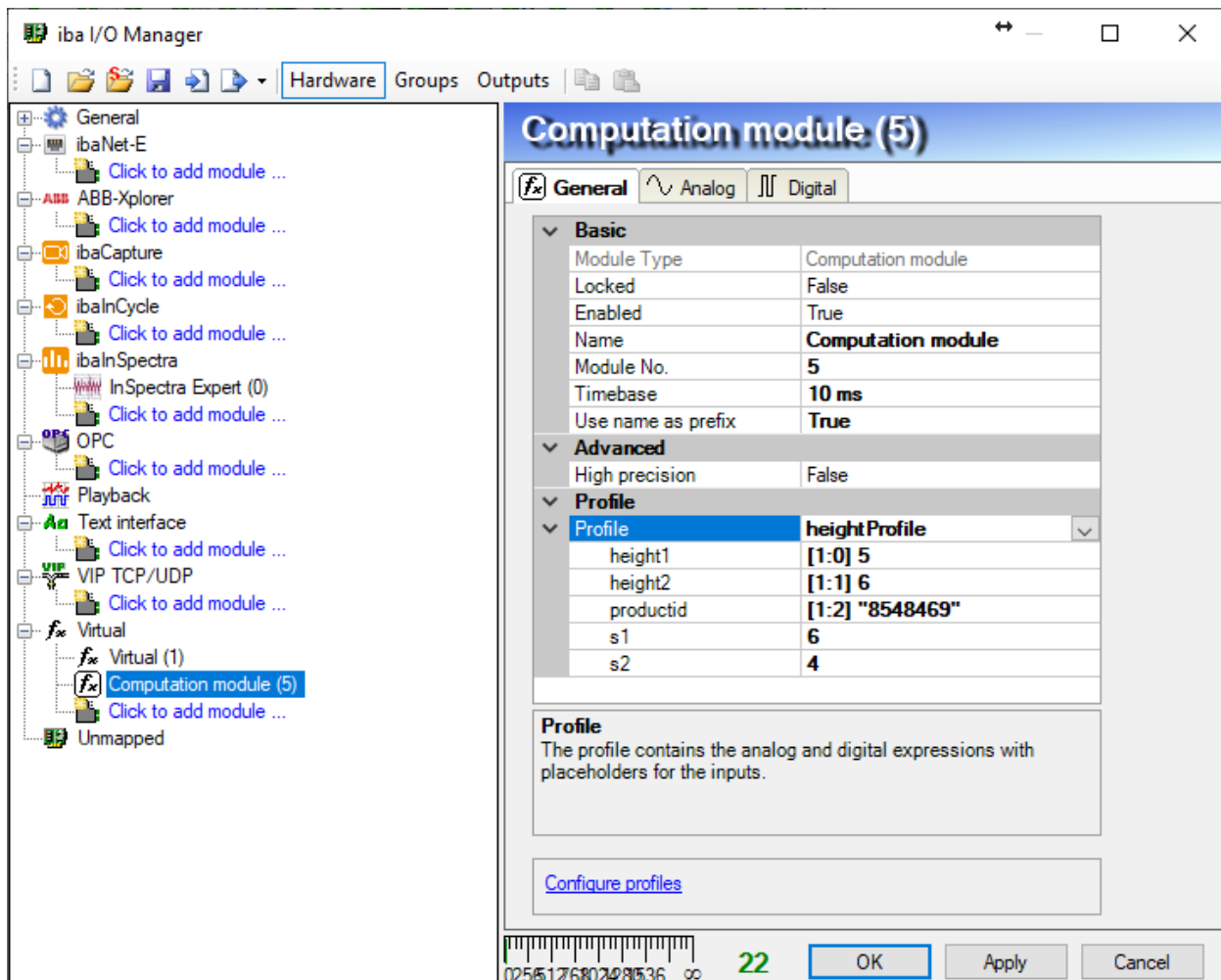
- the placeholder name
- the default constant value
- the value type
- a comment (could be a short description)

The value type is the type of the signal/static value that can be assigned to the placeholder in the general tab of the module:

- Numeric
- Digital
- Text
- Any (numeric default)
- Any (digital default)
- Any (text default)

The profiles can be exported and imported via the buttons at the bottom left in the profile dialog box.

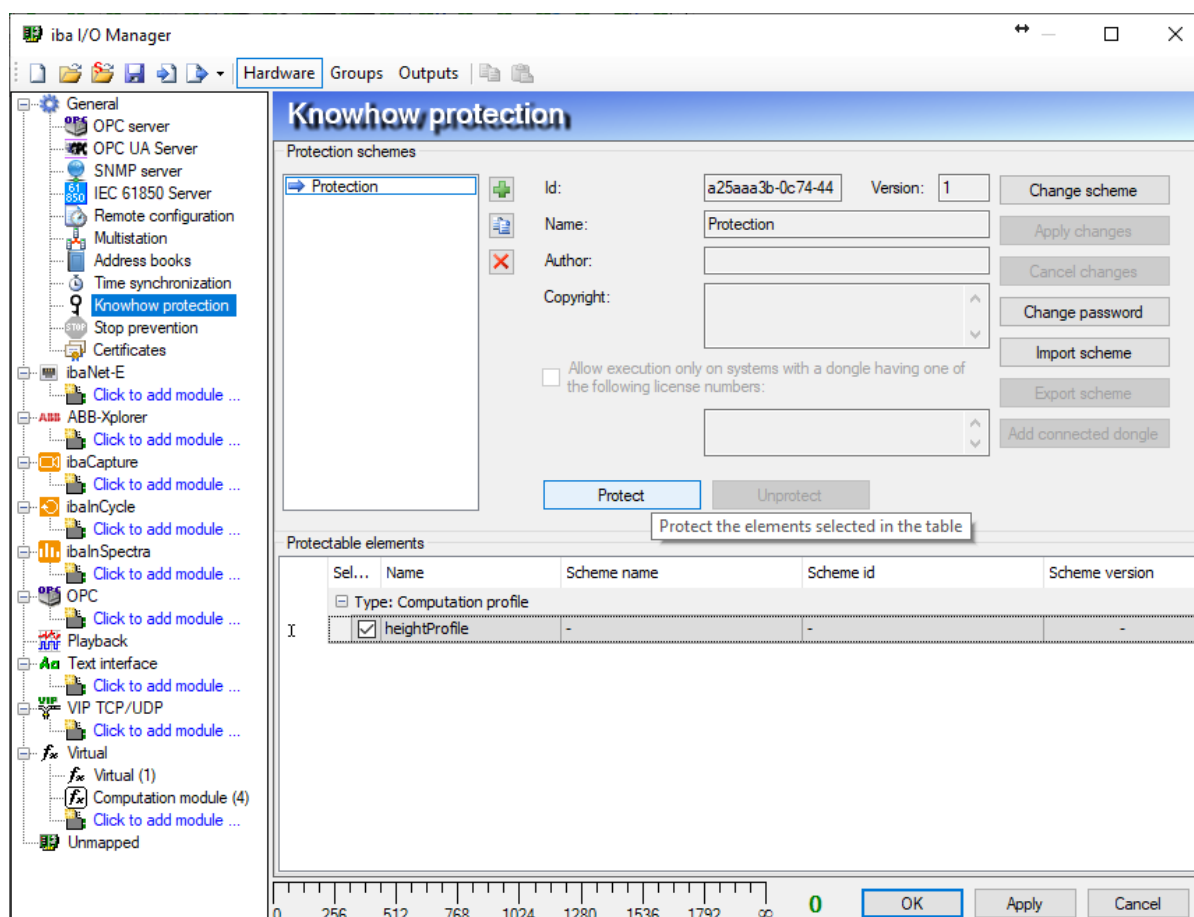
In the tab “General” of the computation module signals need to be assigned to the placeholders (in this example *height1*, *height2*, *s1*, *s2* and *productid*):



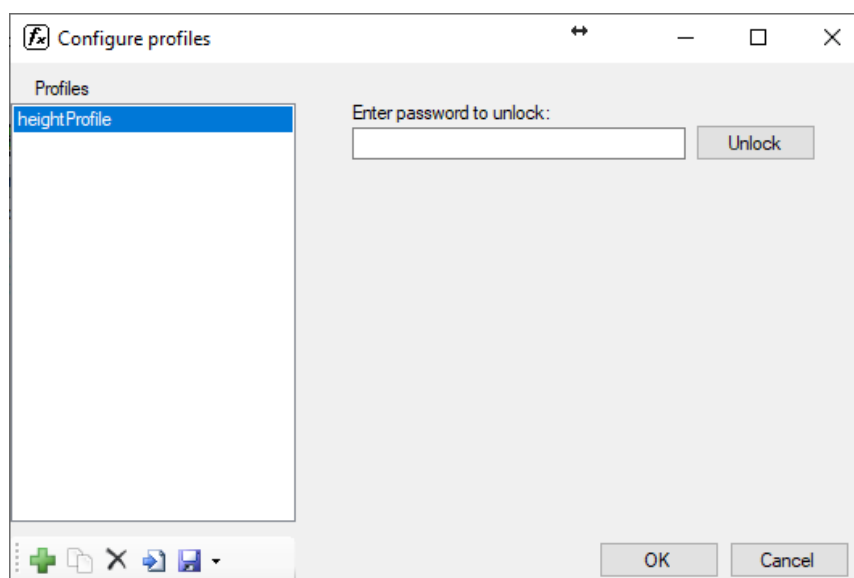
This way the calculations defined in the profile can be reused efficiently by assigning different signals to the placeholders in multiple instances of the computation module.

A computation profile can be protected with a protection scheme using the “Knowhow protection” node in the I/O Manager. A description of the “Knowhow protection” feature can be found in the ibaPDA general manual part 2 covering the I/O manager.

If no protection schemes have been configured before, you have to create one first. The most important settings to be configured are the name and a password. After checking the checkbox next to the profile you want to protect, click the *Protect* button. A dialog box will popup where you have to enter the password to confirm the action.



From then on, you will not be able to see the expressions of the profile anymore in the profile dialog box, until you enter the password. Your expressions (could be intellectual property) are protected:

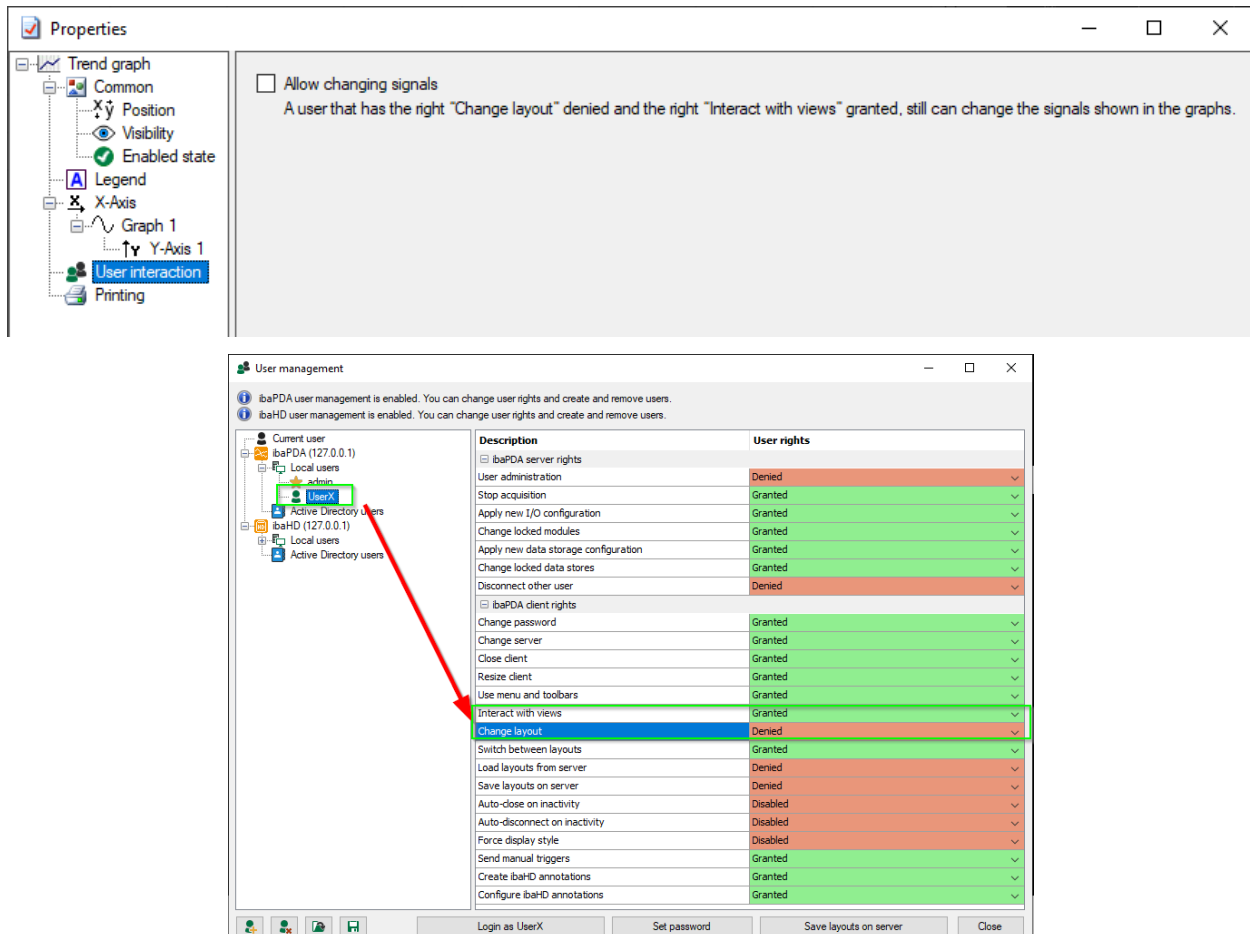


It is also possible to put one or more dongle license numbers in the protection scheme (see screenshot of the "Knowhow protection" node). If the profile is protected with a scheme configured with dongle license numbers, the profile can only run on a computer licensed with one of these dongles.

12 Trend graph changes

12.1 User interaction

In relation to the signal tree there has also been a change in the trend graph. It now has an additional property section called “*User interaction*”. Here you can determine what the user is allowed to do when he doesn’t have the “*Change layout*” right but he does have the “*Interact with views*” right.

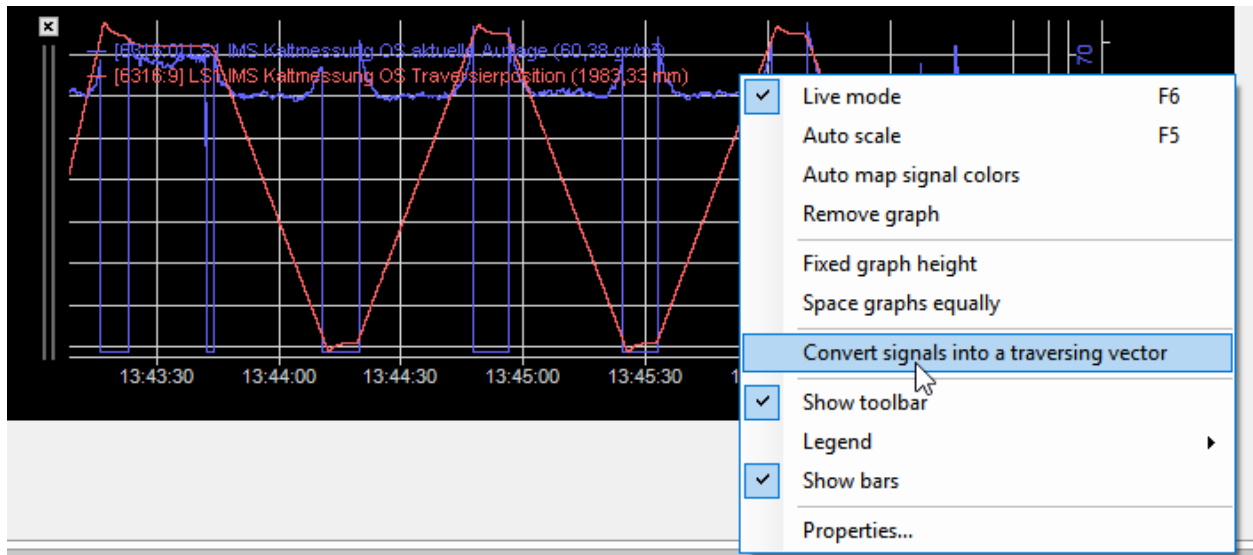


The user is allowed to zoom in and move the X-axis. In addition you can let him change the signals by checking the “Allow changing signals ...”. The user can then drag & drop signals from the signal tree to the trend graph. He can also remove signals and rearrange the signals. When the option is not checked then the signals are fixed. This allows you to configure fixed trend graphs and “free” trend graphs that the user can change.

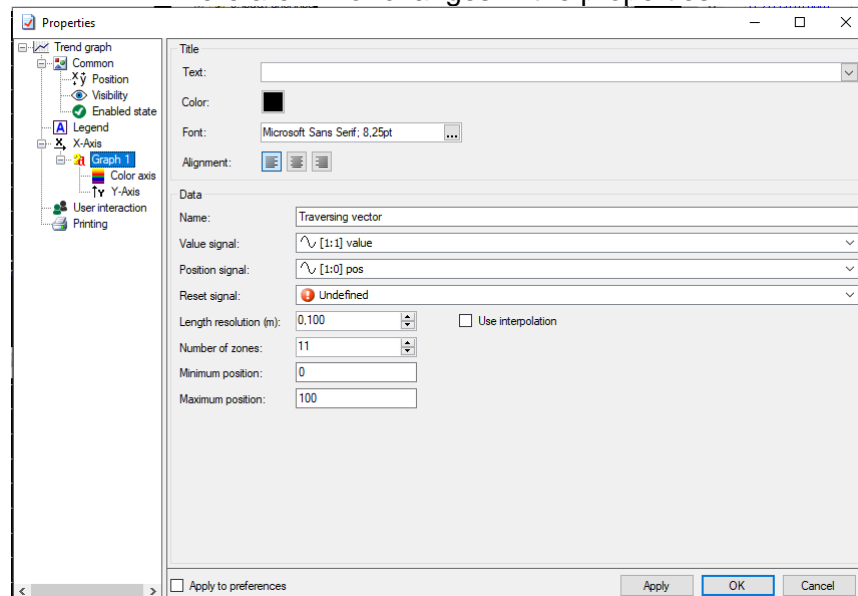
12.2 Traversing vector

The traverse function was introduced in v7.1.0. A traversing vector is a vector that is generated by 2 signals: a position signal and a value signal. It was supported in ibaQPanel in the timebased trend graph and in the timebased and lengthbased offline trend graph. This is now also available in ibaQPanel for the timebased HD trend graph, the lengthbased trend graph and the offline trend graph with length conversion.

Remarks: The function does **NOT** exist for the lengthbased HD trend graph

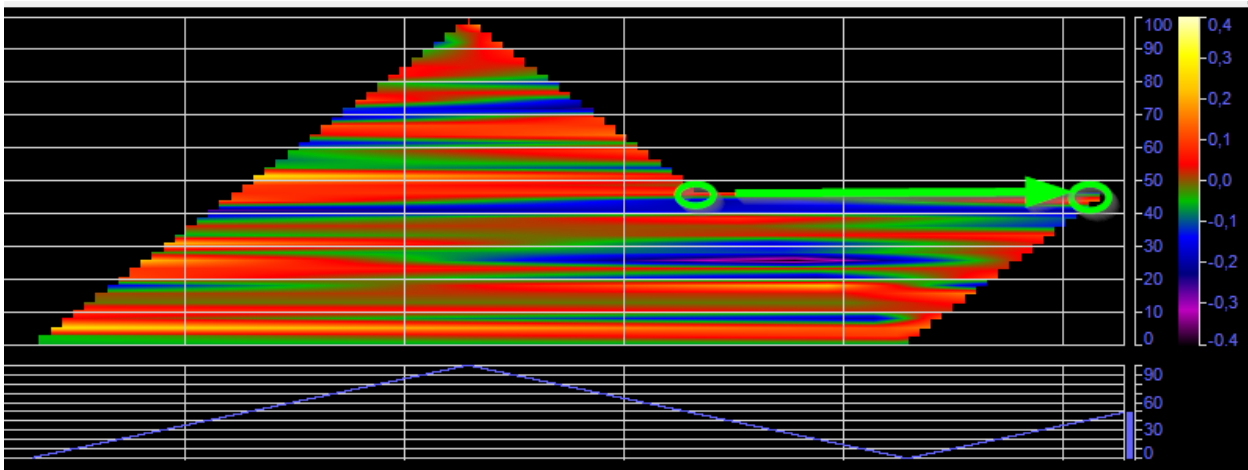


There are minor changes in the properties.

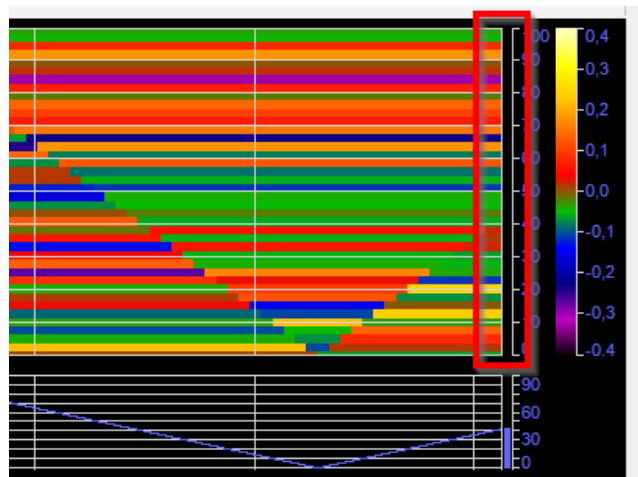


The length resolution refers to the resolution of the position signal.


There is a new option "*Use interpolation*". When interpolation is used then the data is interpolated in both the X and Y direction. In order to do the interpolation we need 2 values so a zone can only be painted when a new point arrives. That is why in interpolated mode you get a jagged edge.

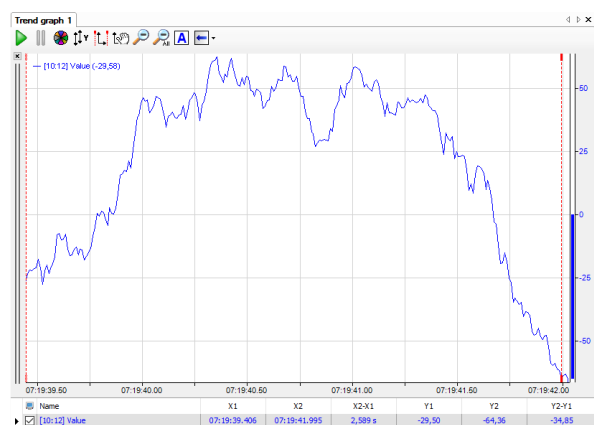
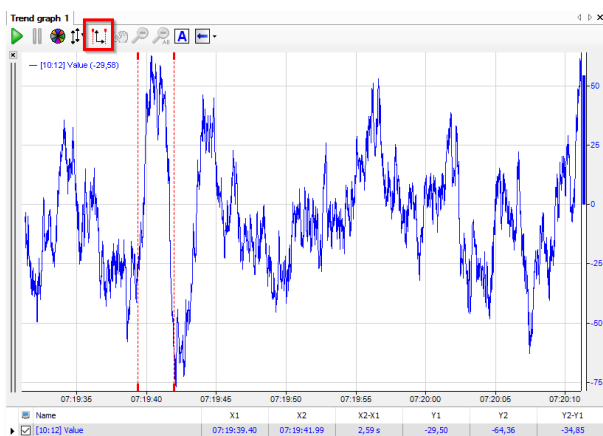


Without interpolation the value only changes when a new value arrives and otherwise the value is the last one. So the painting must not wait for the next value and a straight line can be painted.



12.3 Zoom between markers

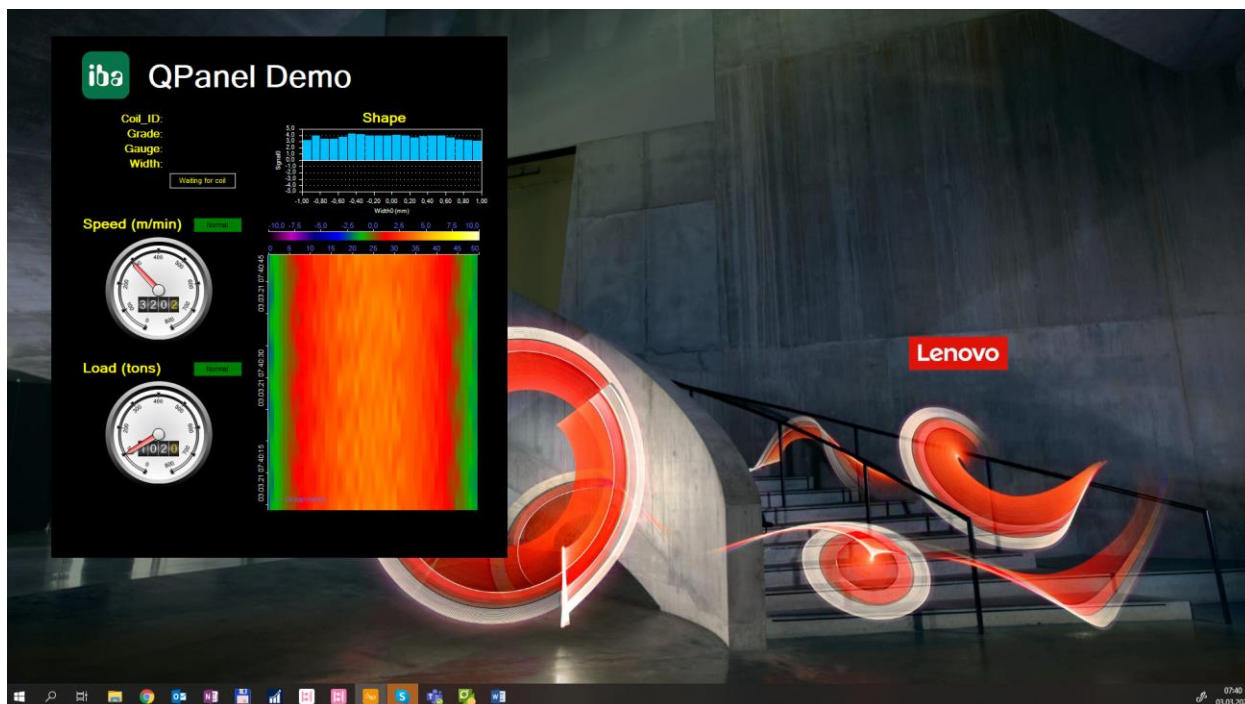
All the trend graphs have now an additional button to zoom with one click between the X1 and X2 markers. 



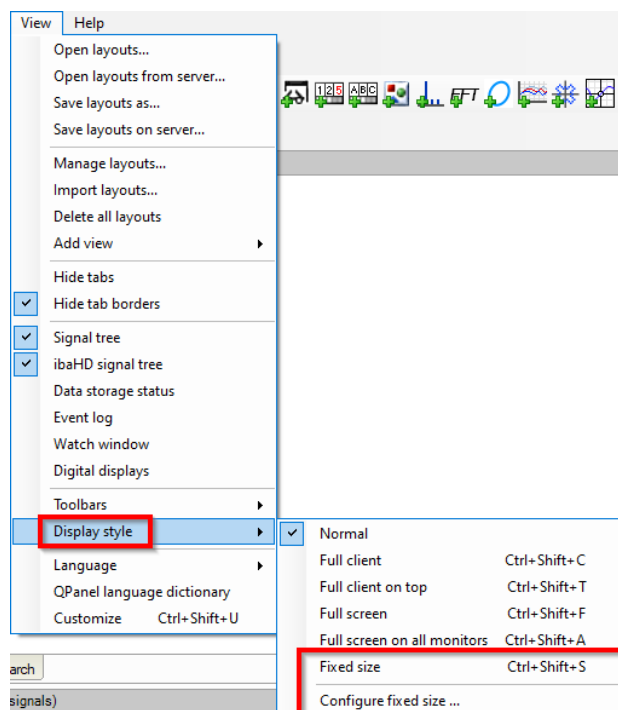
This allows e.g. to set the cursors to a certain timestamp with the mouse by using the timestamp in the marker grid and then zoom with one click into that area.

13 Display style: Fixed size

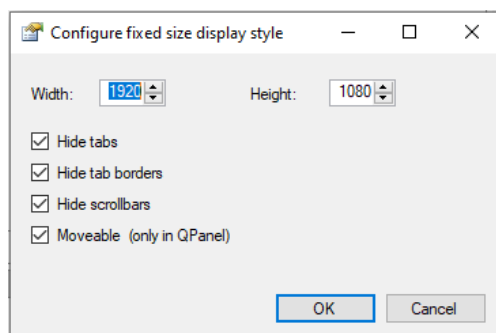
There is a new display style “*Fixed size*” available. It allows to create a window with always the same size on the screen. The window is always on top.



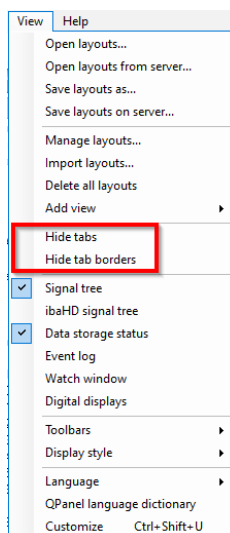
If there is an ibaQPanel inside this layout then the user can move this fixed sized ibaPDA client on the screen by clicking an empty area without an element in ibaQPanel and dragging it around.



Use the “Configure fixed size” menu command to configure the settings of this display style.



The “*Hide tabs*” and “*Hide tab borders*” options override the settings of ibaPDA in the view tab.

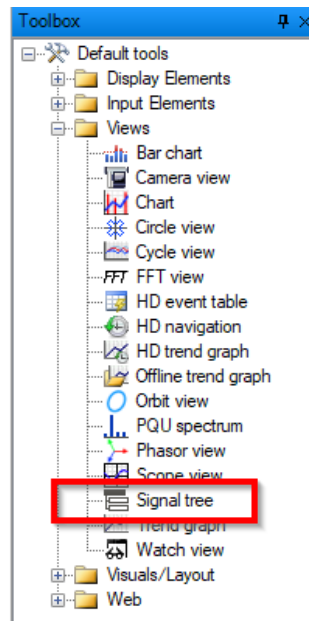


When the configured fixed size is smaller than the layout size then scrollbars will appear. That can be avoided by the option “*Hide scrollbars*”. In that case only the fixed size is visible and the other parts of the layout can’t be reached.

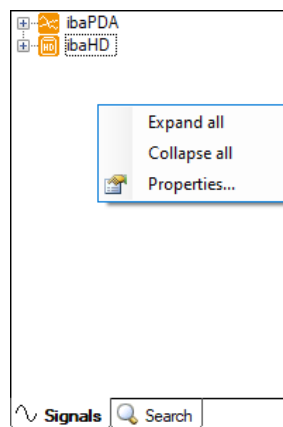
14 ibaQPanel signal tree

The new tool “Signal tree” in ibaQPanel can be used like the normal ibaPDA or ibaHD signal tree. As a tool inside ibaQPanel it allows the user to drag & drop signals to a trend graph or to other tools. This works also in full screen mode.

The tool was introduced in ibaPDA v7.2.1 and it has been extended in ibaPDA v7.3.0 with several new features like a configurable signal list, configurable display style and a search function. You will find the “Signal tree” tool under “Views” in the ibaQPanel toolbox:

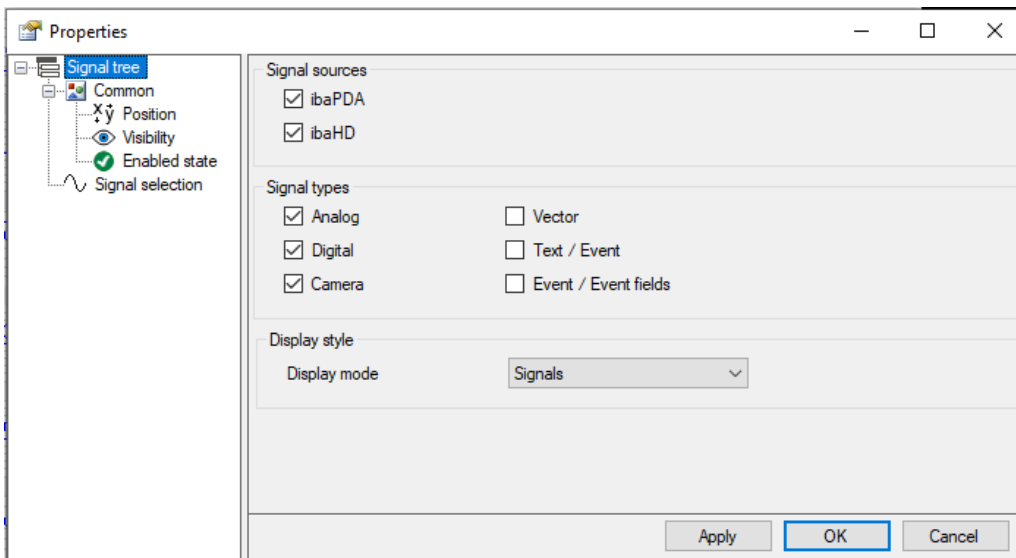


Placing the “Signal tree” via Drag & Drop on the panel, the user will see this signal tree:



Doing a right mouse click within the signal tree a context menu is shown. An option to expand/collapse the tree is available and you can open the properties dialog.

Properties



In the properties dialog you can configure what will be shown in the signal tree.

Signal sources

ibaPDA the ibaPDA signal tree from the currently connected ibaPDA server will be shown

ibaHD the ibaHD signal tree from the currently connected ibaHD server will be shown

Signal types

Analog Analog signals will be shown.

Digital Digital signals will be shown.

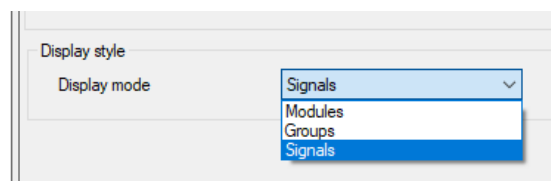
Camera Cameras will be shown.

Vector Vector signals will be shown.

Text / Event All text signals will be shown (events are also text signals).

Event / Event fields The events and also the variable analog, digital or text signals inside an event message will be shown.

Display style



The display style determines how the signals are grouped in the signal tree.

Modules Signals are grouped by module.

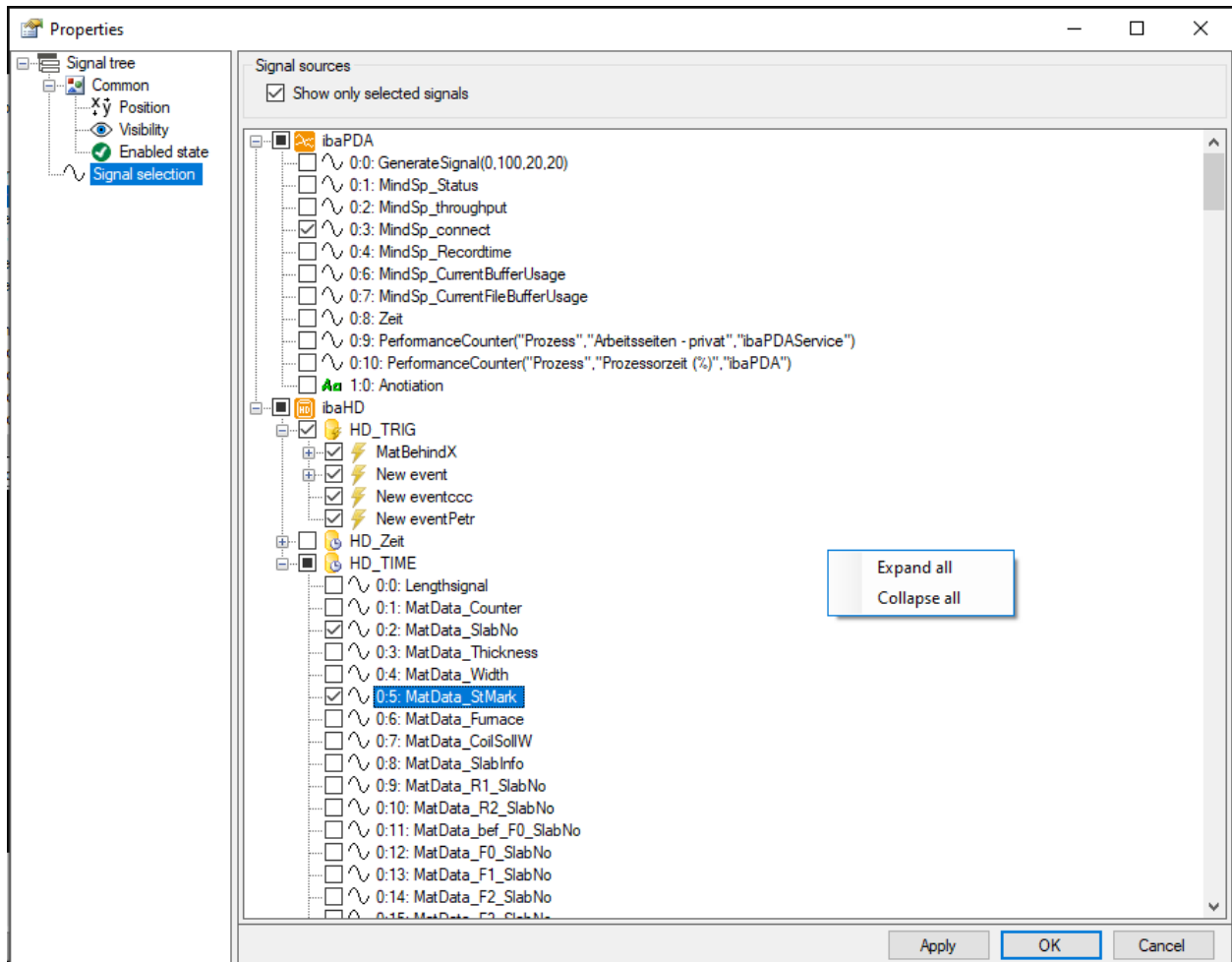
Groups Signals are grouped by group.

Signals Signals are not grouped.

Remarks: Use the “Groups” option to have a more plant specific view. The groups must be defined in the I/O manager of ibaPDA.

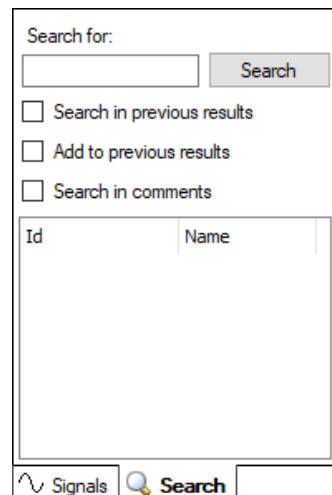
Signal selection

The signal selection allows you to determine which signals should be available in the signal tree. By default, all signals are available. If you want to limit the available signals then select “Show only selected signals” and make your selection by using the checkboxes in the signal tree.

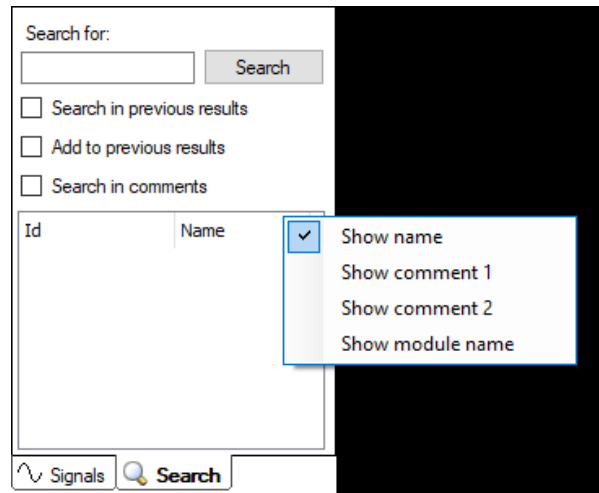


Search

The search function allows the user to find certain signals. This function is similar to the normal ibaPDA signal tree. There are some differences in the details.



The first difference is that it is possible to search also in groups if the group display style is selected. Another difference is that the module name can be made visible in a column.

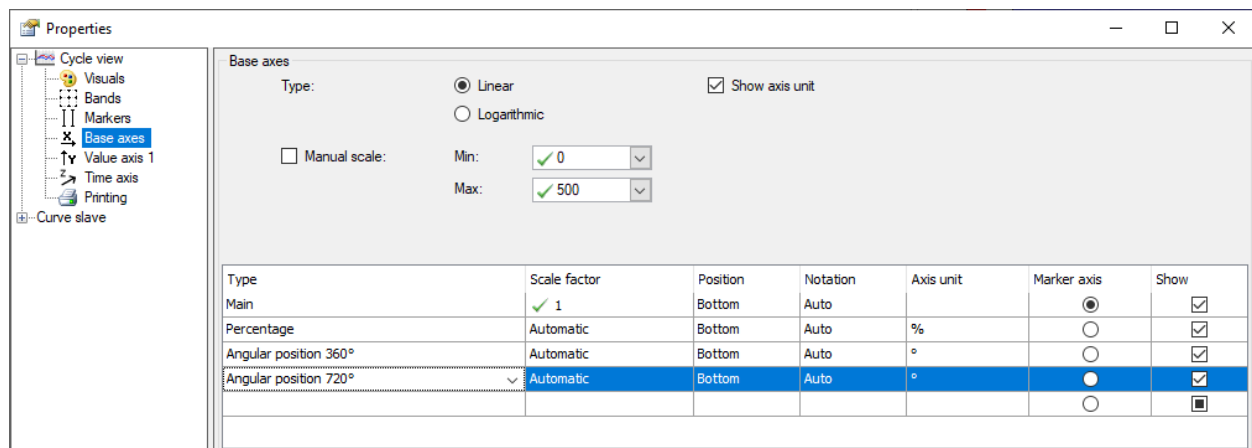


15 New options for X-Axis in the cycle-view

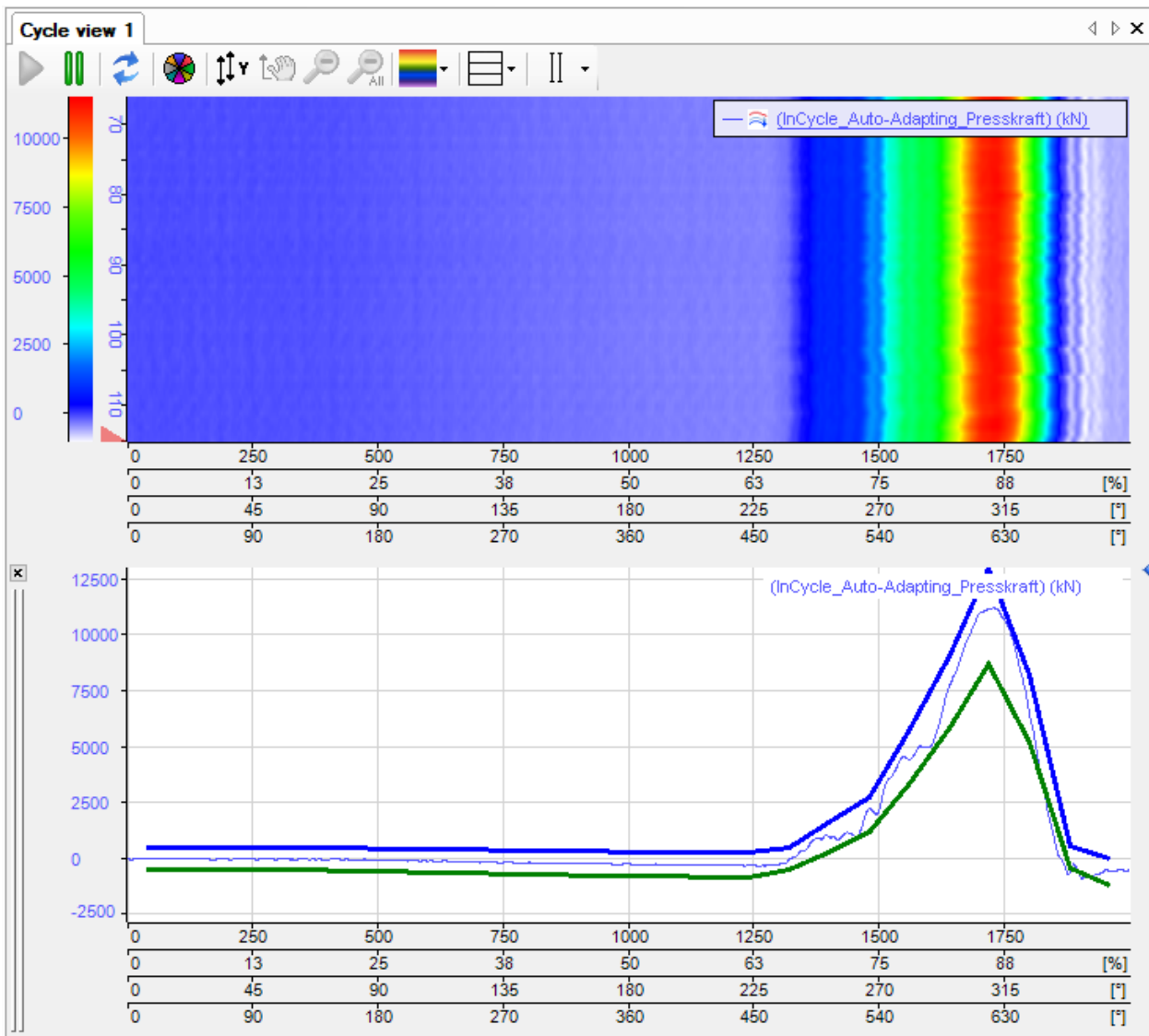
The base axes in the Cycle view now contain three preconfigured alternative axes. While the existing “Main”-axis displays the number of the samples from the time-synchronous-averaging, the new axes options do an automatic scaling to the most common representations of cyclic processes:

- **Percentage:** Shows the cycle in percentage from 0 % to 100 %. This is a common representation for most repeating processes for non-rotating applications.
- **Angular position 360°:** This is the most common representation for rotating machinery, showing the angular position in degree per rotation.
- **Angular position 720°:** This is a special representation mainly used for combustion engines. Since each cylinder fires every second revolution a display of the cycle over two revolutions is required to get reproducible results. (Hint: For triggering these cycles correctly using two subcycles with a trigger occurring once per revolution can be used.)

When adding a new axis in the view properties, the type can be selected in the first column of the table. A custom axis with an individual scale factor can be configured by selecting “custom”.



The settings automatically apply to the main window and the cycle slave graph.



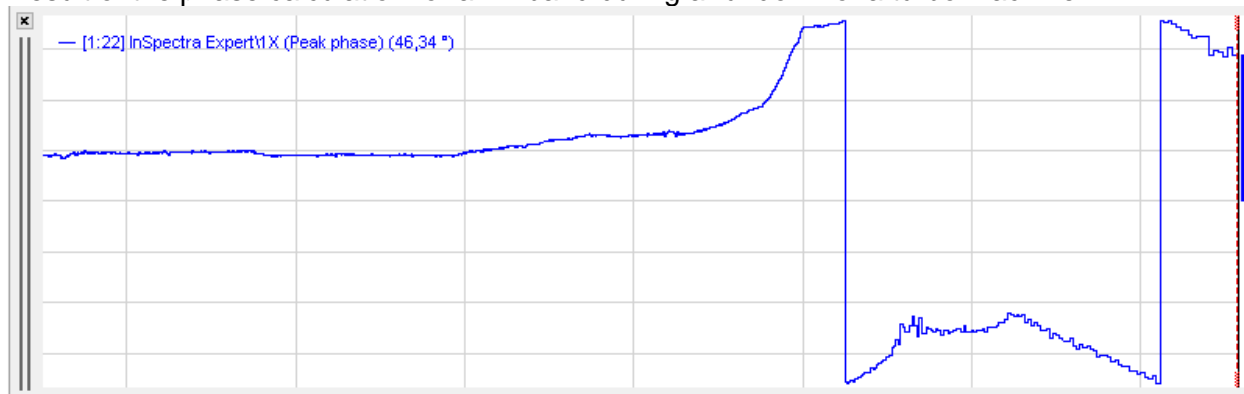
Note: The main axis cannot be removed, but will not be displayed by deactivating the “show” checkbox in the last column of the table.

16 InSpectra-Expert new band-indicators: Crest and Phase

For ibalInSpectra-Expert two new band-indicators are now available:

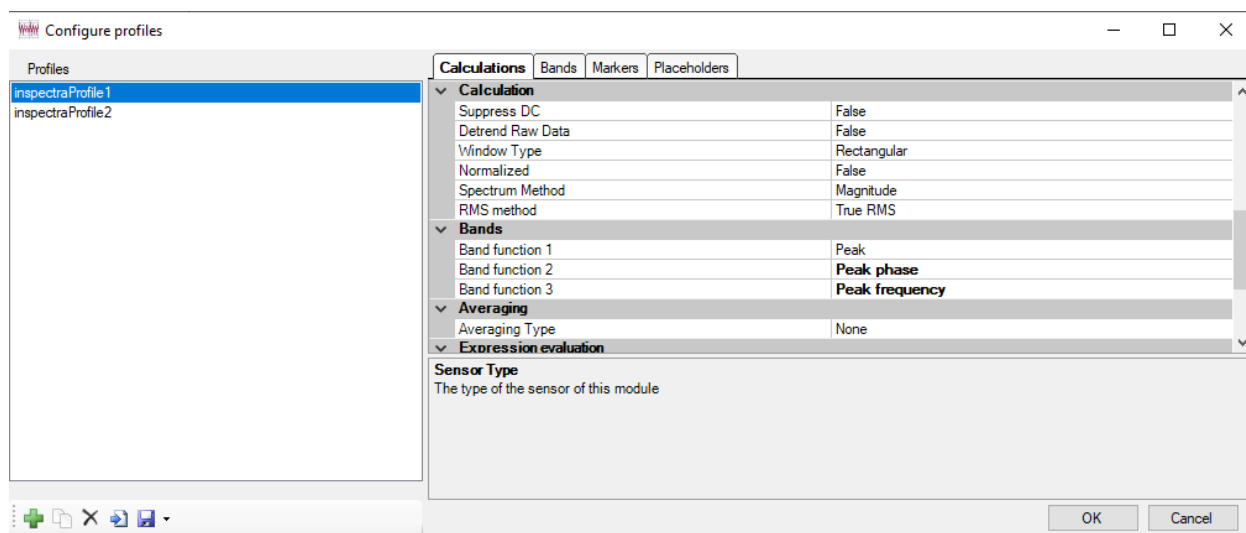
- **Crest:** The crest factor is a common indicator in vibration analysis, showing the ratio of peak values to the effective value of the signal. Hence it mainly should be used on broad frequency bands. It is calculated as the peak within the band divided by the mathematical RMS value of the band. (https://en.wikipedia.org/wiki/Crest_factor)
- **Peak phase:** The phase is calculated based on the complex result of the FFT-calculation. The phase value for the frequency where the highest peak occurs within the bands is outputted as result. This result is given as an angular value between -180° and $+180^\circ$.

Result of the phase calculation for a 1X band during a rundown of a turbo machine:



Note: For getting reliable results it is recommended to use orderresampling with an integer number of revolutions (oscillations) per FFT calculation. E.g. 32 samples per revolution and 512 as number of samples. The calculation also requires a reliable speed signal. A pulse train or pulse counter works best for reproducible results.

For configuration of these band indicators an option to define the three possible results per band was added in the calculation settings in the profile. Here three out of the five possible band functions can be selected and ordered. The available band functions are: peak, peak frequency, RMS, peak phase, and crest.



17 Results as vector for ibalnspectra- and ibalncycle-Expert

The Expert-Modules in ibalnspectra and ibalncycle now offer the option to create vectors for one band indicator from all bands directly. In the general tab of the module the required band indicator can be selected in the new section “Data recording”.

The screenshot shows the 'iba I/O Manager' window with the 'Hardware' tab selected. On the left, a tree view lists various modules, with 'InSpectra Expert (1)' highlighted. The right pane displays the configuration for this module, organized into several sections:

- General** (selected):

Module Type	InSpectra Expert
Locked	False
Enabled	True
Name	InSpectra Expert
Module No.	1
Timebase	10 ms
- Calculations**:

Enable Calculations	Always
Hold values	True
Frequency Resolution	0,0625 Order
Max Frequency	12,4375 Order
Update Time	Speed dependent
- Preprocessing**:

Preprocessing	<No preprocessing>
---------------	--------------------
- Profile**:

Profile	inspectraProfile 1
---------	--------------------
- Settings**:

Input signal	[8191:1] Displacement_Y
Speed signal	[0.0] Keyphase
- Snapshots**:

Periodic snapshots	False
External trigger	Unassigned
Frequency Resolution	6,1E-05 Order
Max Frequency	12,4999 Order
- Data recording**:

Band vectors	<None>
--------------	--------

 A dropdown menu is open for 'Band vectors', showing the following options:
 - ☒ Peak
 - ☐ Peak frequency
 - ☐ RMS
 - ☐ Peak phase
 - ☐ Crest

For the selected indicator a vector over all bands is automatically created and can be recorded. This vector enables easier analysis in ibaAnalyzer especially for comparing data from several sensors.