New Features in ibaPDA v6.36.0

Author: iba Gent

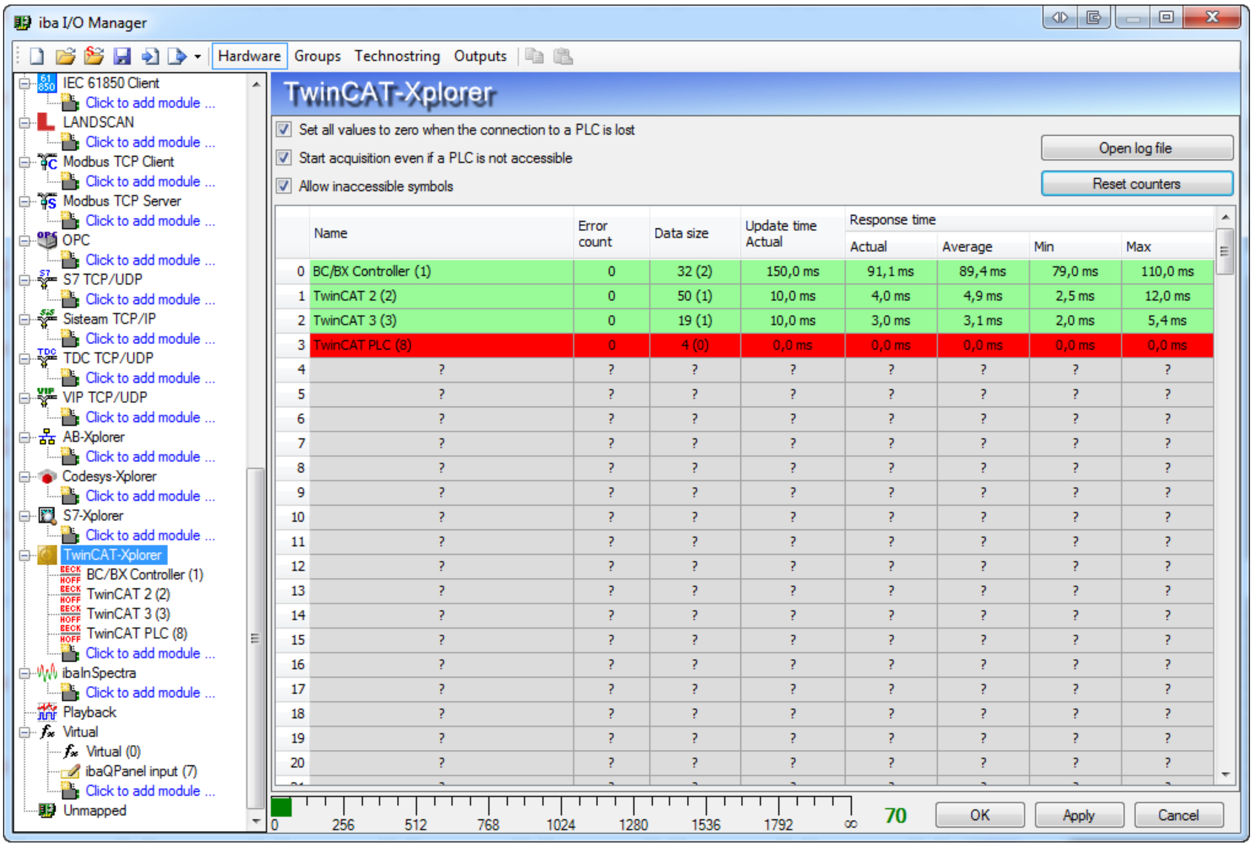
Date: 04/07/2016

iba AG • Königswarterstr. 44 • D-90762 Fürth • Germany • www.iba-ag.com

Table of contents

1 TwinCAT-Xplorer

The TwinCAT-Xplorer interface in ibaPDA is used to measure data from Beckhoff PLCs. The interface supports TwinCAT versions 2 and 3 running on industrial PCs, embedded PCs (CX series) and bus controllers (BC/BX series). It is an Xplorer interface which means that the data is cyclically read by ibaPDA instead of being sent by the PLC. ibaPDA uses the Beckhoff ADS protocol to communicate with the PLCs.



☑ Set all values to zero when the connection to a PLC is lost

☑ Start acquisition even if a PLC is not accessible

☑ Allow inaccessible symbols

	Name	Error count	Data size	Update time Actual	Response time			
					Actual	Average	Min	Max
0	BC/BX Controller (1)	0	32 (2)	150,0 ms	91,1 ms	89,4 ms	79,0 ms	110,0 ms
1	TwinCAT 2 (2)	0	50 (1)	10,0 ms	4,0 ms	4,9 ms	2,5 ms	12,0 ms
2	TwinCAT 3 (3)	0	19 (1)	10,0 ms	3,0 ms	3,1 ms	2,0 ms	5,4 ms
3	TwinCAT PLC (8)	0	4 (0)	0,0 ms	0,0 ms	0,0 ms	0,0 ms	0,0 ms
4	?	?	?	?	?	?	?	?
5	?	?	?	?	?	?	?	?
6	?	?	?	?	?	?	?	?
7	?	?	?	?	?	?	?	?
8	?	?	?	?	?	?	?	?
9	?	?	?	?	?	?	?	?
10	?	?	?	?	?	?	?	?
11	?	?	?	?	?	?	?	?
12	?	?	?	?	?	?	?	?
13	?	?	?	?	?	?	?	?
14	?	?	?	?	?	?	?	?
15	?	?	?	?	?	?	?	?
16	?	?	?	?	?	?	?	?
17	?	?	?	?	?	?	?	?
18	?	?	?	?	?	?	?	?
19	?	?	?	?	?	?	?	?
20	?	?	?	?	?	?	?	?

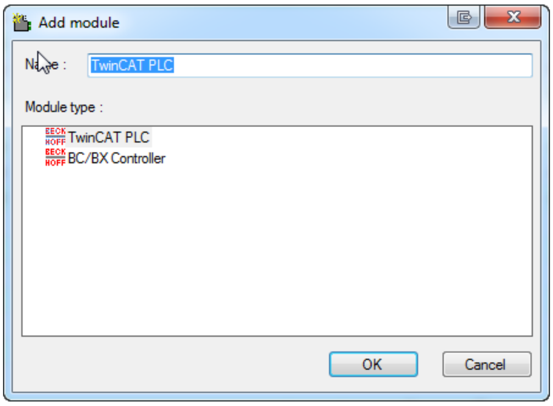
The TwinCAT-Xplorer interface shows a table of the available connections. Per TwinCAT-Xplorer license you get 16 connections. A maximum of 240 connections is allowed. This means that maximum 15 licenses can be used. Each connection corresponds with a row in the table. The row is green when the connection is ok and data is being read. The row is orange when the connection is ok but the data is coming in 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 of the response time. The update time is the time between 2 reads. The data size shows how much data is read per read. The number in brackets shows how many ADS commands are required to read the data. The higher the number of required ADS commands is the slower the response time will be. You can use the “Reset counters” button to clear the counters for all 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 get the address of a symbol that is no longer available in the PLC then the PLC will return an error. If the option “Allow inaccessible symbols” 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.



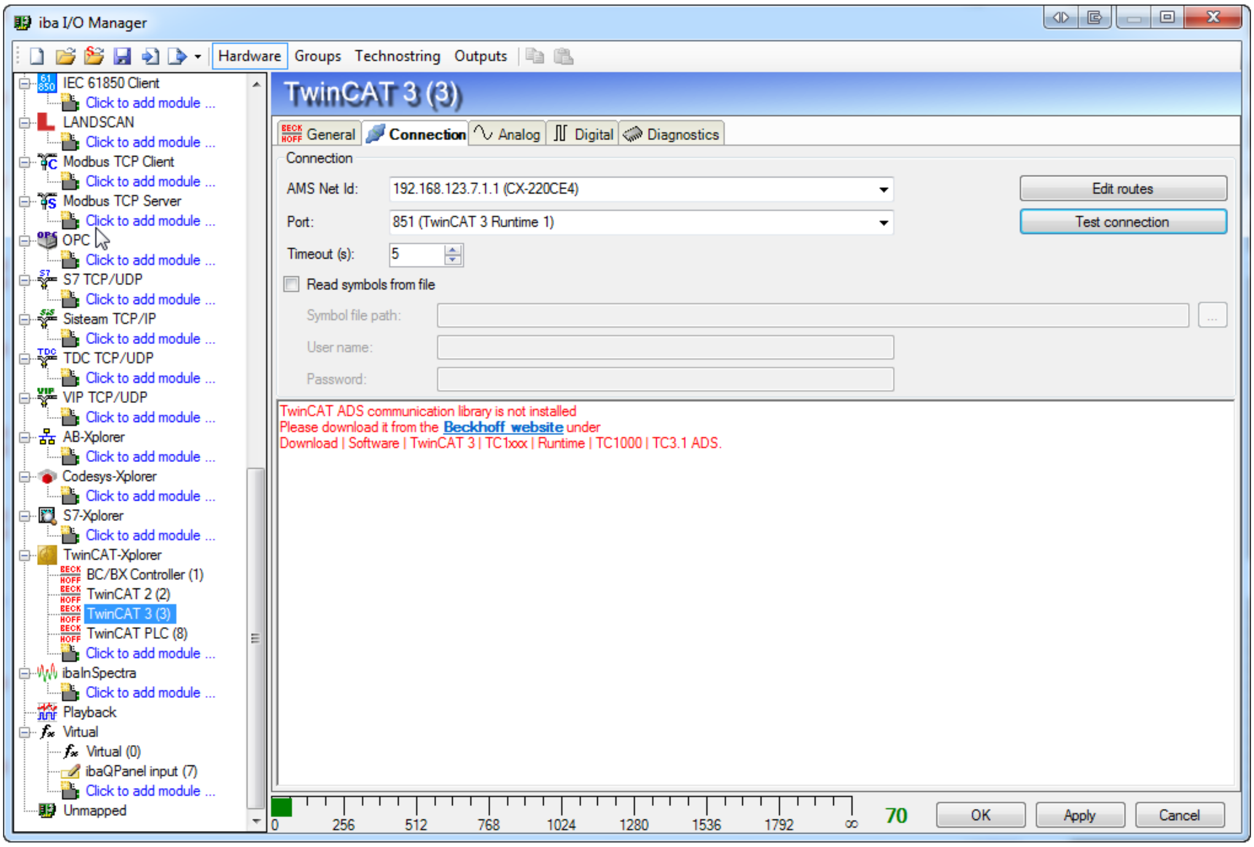
Name : TwinCAT PLC

Module type :

TwinCAT PLC
BC/BX Controller

There are 2 types of modules that you can add to the TwinCAT-Xplorer interface:

- TwinCAT PLC: Use this when you want to measure from an industrial PC or embedded PC (CX series).
- BC/BX Controller: Use this when you want to measure from a bus controller.



Connection

AMS Net Id: 192.168.123.7.1.1 (CX-220CE4)

Port: 851 (TwinCAT 3 Runtime 1)

Timeout (s): 5

□ Read symbols from file

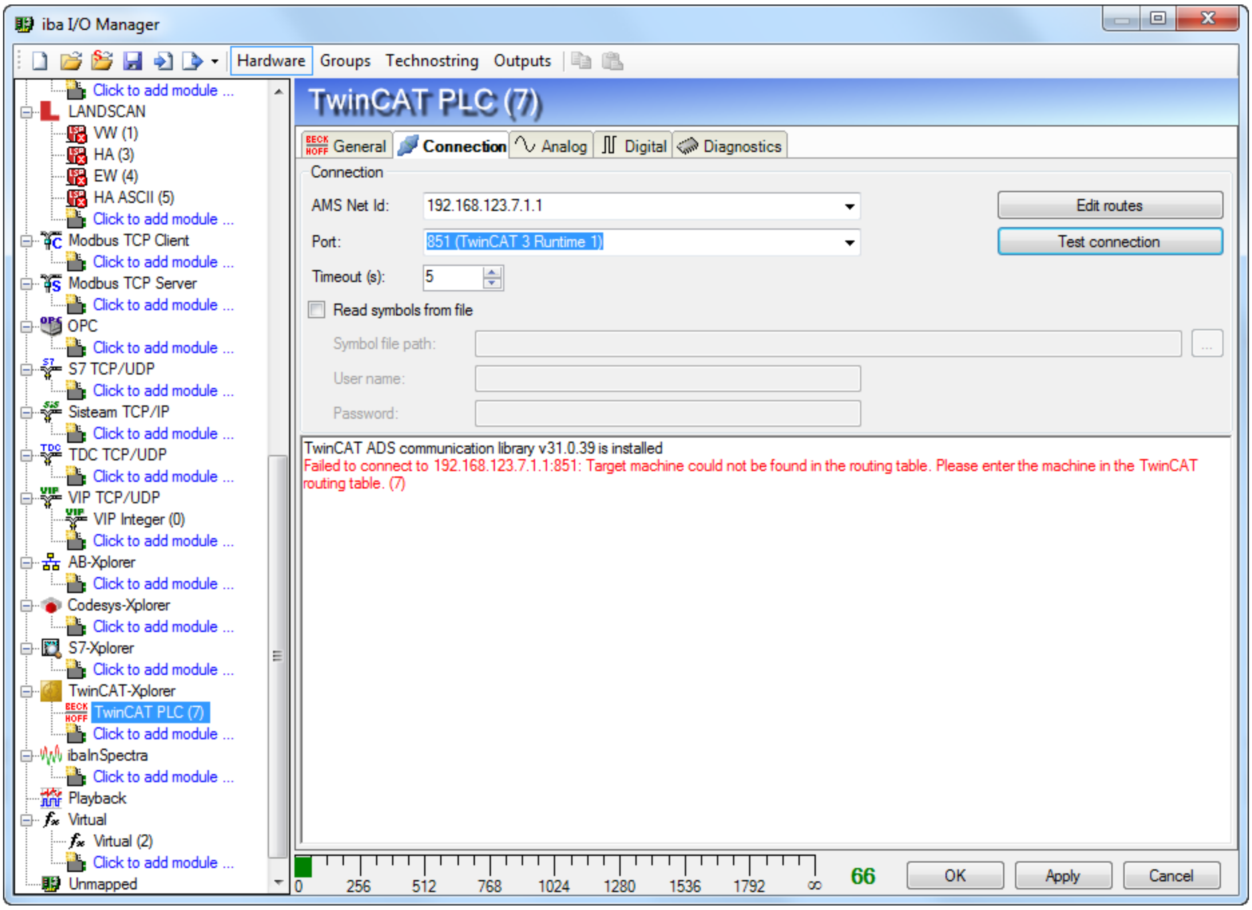
Symbol file path:

User name:

Password:

TwinCAT ADS communication library is not installed
Please download it from the Beckhoff website under
Download | Software | TwinCAT 3 | TC1xxx | Runtime | TC1000 | TC3.1 ADS.

ibaPDA uses the TwinCAT ADS communication library of Beckhoff to connect to the PLCs. This library is not included in the ibaPDA installation. If TwinCAT v2 or v3 is installed on the PC where the ibaPDA service is running then the library will be available. If TwinCAT is not installed then you will have to download the TwinCAT 3 ADS runtime from the Beckhoff website (http://www.beckhoff.com). Navigate to Dowload | Software | TwinCAT 3 | TC1xxx | Runtime | TC1000 | TC3.1 ADS. Start the installer and select full install.



Connection

AMS Net Id: 192.168.123.7.1.1

Port: 851 (TwinCAT 3 Runtime 1)

Timeout (s): 5

□ Read symbols from file

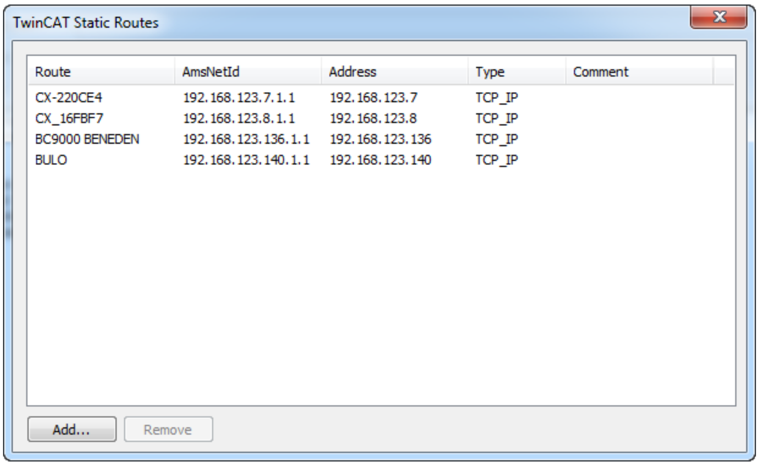
```
TwinCAT ADS communication library v31.0.39 is installed
Failed to connect to 192.168.123.7.1.1:851: Target machine could not be found in the routing table. Please enter the machine in the TwinCAT routing table. (7)
```

On the connection tab you have to configure the address of the PLC you want to connect to. In ADS the address consists of an AMS Net Id and a port number. The AMS Net Id is the address of an ADS router. It is an extension of the IP address. It consists of 6 bytes. Usually the first 4 bytes are the IP address and the last 2 bytes are 1. The port number determines the ADS device that is connected to the router. There are predefined port numbers for the TwinCAT runtimes:

- 800: for BC/BX bus controllers
- 801, 811, 821, 831: for the 4 possible TwinCAT 2 runtimes
- 851, 852, 853, 854: for the 4 possible TwinCAT 3 runtimes

On the ibaPDA server PC there is also an ADS router installed when the TwinCAT ADS library is installed. An entry to the remote PLC needs to be entered in the routing table of this ADS router. Also an entry for the ibaPDA server PC's AMS Net Id needs to be entered in the routing table of the remote ADS router. This can be done by the ADS router configuration tool. The tool can be started via the *Edit routes* button when you are on the ibaPDA server PC.

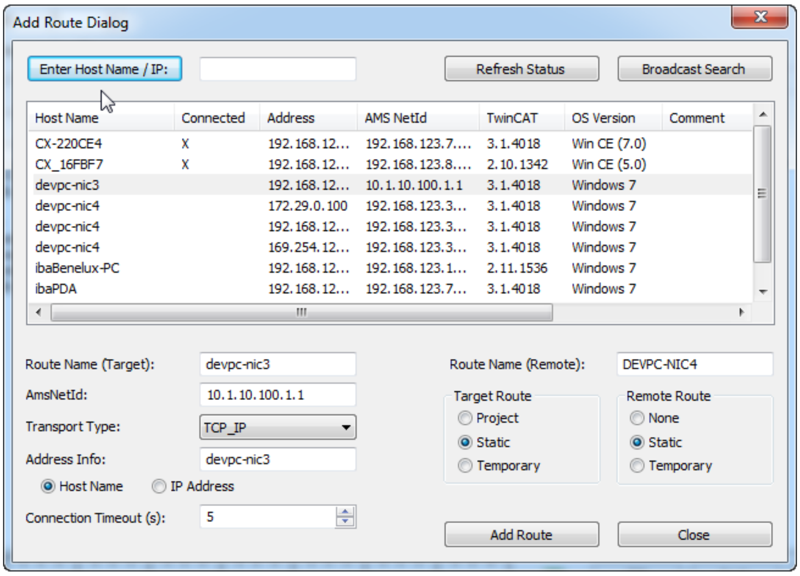
This will open the following dialog when TwinCAT 3 or the TwinCAT 3 ADS runtime is installed.



TwinCAT Static Routes

Route	AmsNetId	Address	Type	Comment
CX-220CE4	192.168.123.7.1.1	192.168.123.7	TCP_IP	
CX_16FBF7	192.168.123.8.1.1	192.168.123.8	TCP_IP	
BC9000 BENEDEN	192.168.123.136.1.1	192.168.123.136	TCP_IP	
BULO	192.168.123.140.1.1	192.168.123.140	TCP_IP	

Click the *Add…* button to add a new route.



Add Route Dialog

Host Name	Connected	Address	AMS NetId	TwinCAT	OS Version	Comment
CX-220CE4	X	192.168.12...	192.168.123.7....	3.1.4018	Win CE (7.0)	
CX_16FBF7	X	192.168.12...	192.168.123.8....	2.10.1342	Win CE (5.0)	
devpc-nic3		192.168.12...	10.1.10.100.1.1	3.1.4018	Windows 7	
devpc-nic4		172.29.0.100	192.168.123.3...	3.1.4018	Windows 7	
devpc-nic4		192.168.12...	192.168.123.3...	3.1.4018	Windows 7	
devpc-nic4		169.254.12...	192.168.123.3...	3.1.4018	Windows 7	
ibaBenelux-PC		192.168.12...	192.168.123.1...	2.11.1536	Windows 7	
ibaPDA		192.168.12...	192.168.123.7...	3.1.4018	Windows 7	

Route Name (Target): devpc-nic3

AmsNetId: 10.1.10.100.1.1

Transport Type: TCP_IP

Address Info: devpc-nic3

● Host Name ○ IP Address

Connection Timeout (s): 5

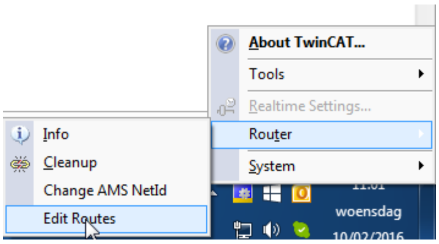
Route Name (Remote): DEVPC-NIC4

Target Route: ○ Project ● Static ○ Temporary

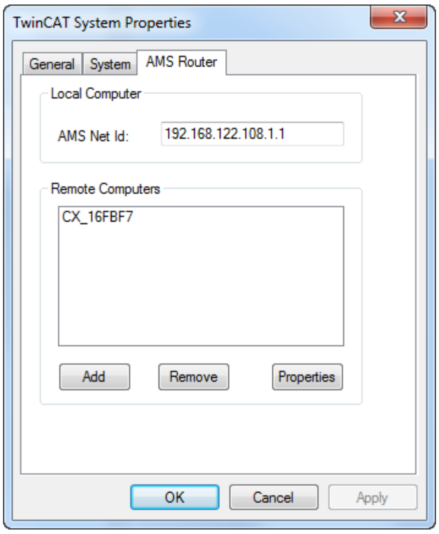
Remote Route: ○ None ● Static ○ Temporary

Use the *Broadcast Search* button to look on the local network for compatible TwinCAT PLCs. Then either select a found PLC or enter the *Route Name*, *AmsNetId* and *Address Info* manually. Select *Static* for *Target Route*. Select *Static* for *Remote Route* when connecting to a TwinCAT PLC. Select *None* for *Remote Route* when connecting to a bus controller. Finally click the *Add Route* button to add the route into the routing table. The router will try to connect to the remote ADS router and might require a user name and password. If the connection was successful then an X will appear in the *Connected* column.

You can also open the configuration tool via the context menu on the TwinCAT system tray icon. Use the command *Router* -> *Edit Routes*.



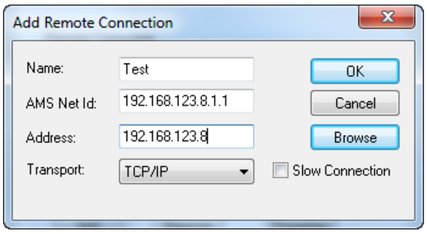
When TwinCAT 2 is installed then the following dialog will be opened.



AMS Net Id: 192.168.122.108.1.1

CX_16FBF7

Click the *Add* button to add a new route.



Name: Test

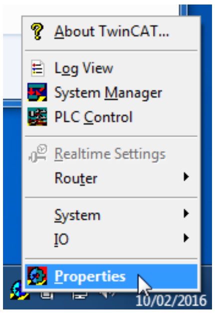
AMS Net Id: 192.168.123.8.1.1

Address: 192.168.123.8

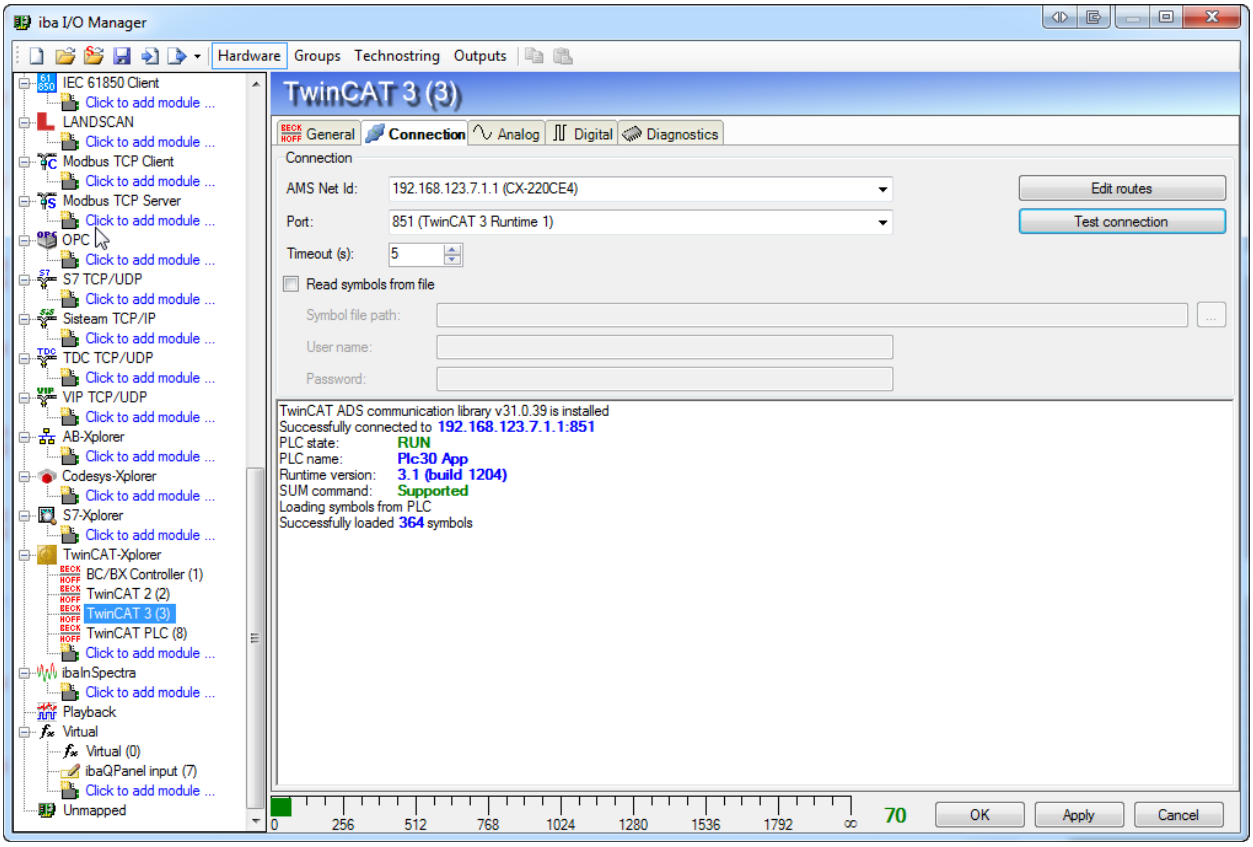
Transport: TCP/IP

☐ Slow Connection

You have to manually enter the *Name*, *AMS Net Id*, *Address* fields. Click the *OK* button to add the route into the routing table.



You can also open the configuration tool via the context menu on the TwinCAT system tray icon. Use the command *Properties*.



Connection

AMS Net Id: 192.168.123.7.1.1 (CX-220CE4)

Port: 851 (TwinCAT 3 Runtime 1)

Timeout (s): 5

□ Read symbols from file

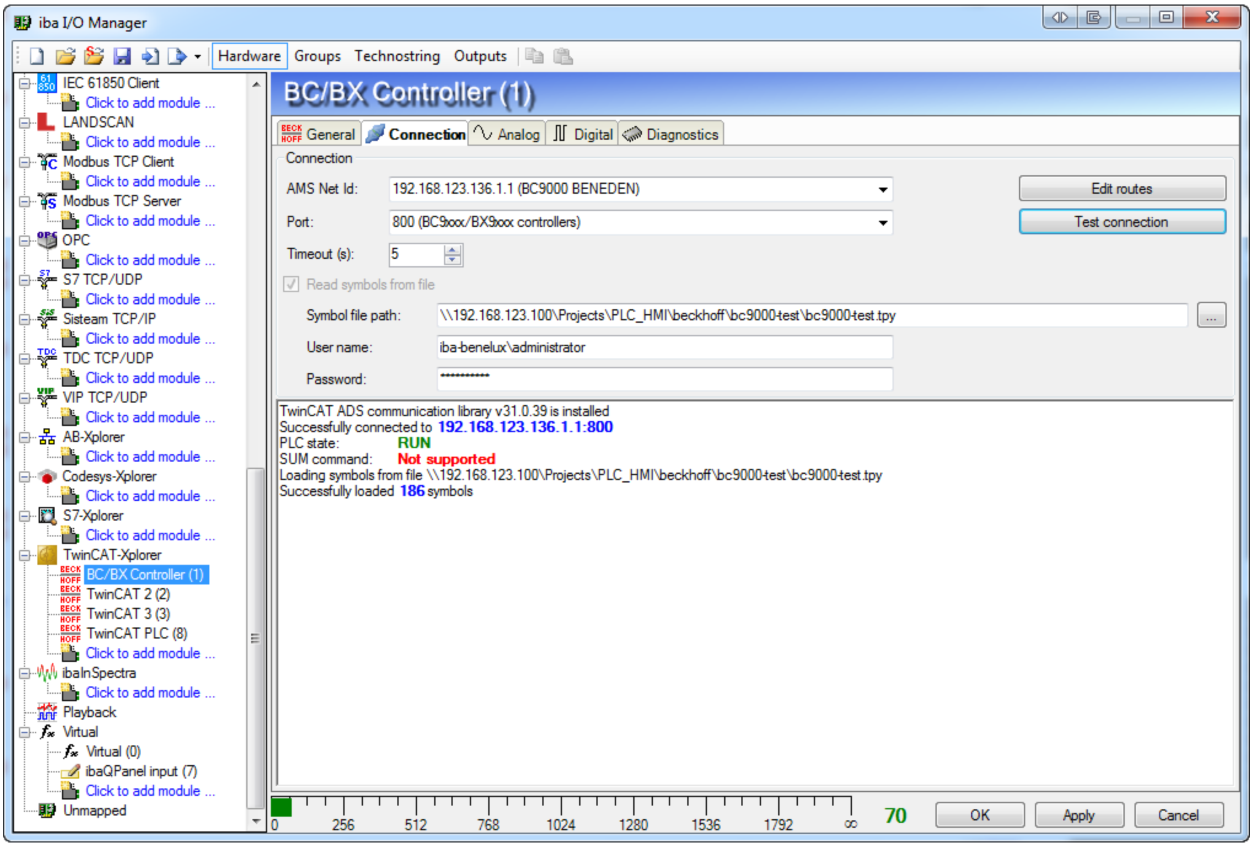
Symbol file path:

User name:

Password:

```
TwinCAT ADS communication library v31.0.39 is installed
Successfully connected to 192.168.123.7.1.1:851
PLC state:        RUN
PLC name:         Plc30 App
Runtime version:  3.1 (build 1204)
SUM command:      Supported
Loading symbols from PLC
Successfully loaded 364 symbols
```

Once the routing tables are correct you can connect to the PLC in ibaPDA. Use the *Test connection* button on the *Connection* tab. When you test the connection you get some information about the connected PLC like its current state, its name, the runtime version and whether it supports the SUM command. The SUM command is used to send multiple ADS read commands inside of a single ADS SUM command. This speeds up the response time a lot. On most TwinCAT PLCs the symbols are stored on the PLC itself. If this is the case then ibaPDA will load the symbols from the PLC during the connection test.



Connection

AMS Net Id: 192.168.123.136.1.1 (BC9000 BENEDEN)

Port: 800 (BC9xxx/BX9xxx controllers)

Timeout (s): 5

☑ Read symbols from file

Symbol file path: \\192.168.123.100\Projects\PLC_HMI\beckhoff\bc9000-test\bc9000-test.tpy

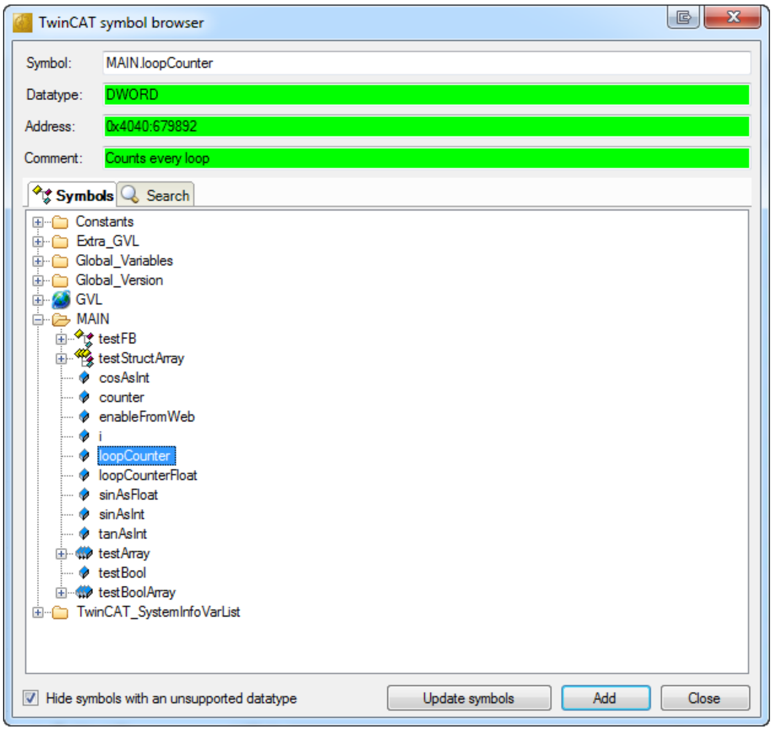
User name: iba-benelux\administrator

Password: ***********

```
TwinCAT ADS communication library v31.0.39 is installed
Successfully connected to 192.168.123.136.1.1:800
PLC state:          RUN
SUM command:        Not supported
Loading symbols from file \\192.168.123.100\Projects\PLC_HMI\beckhoff\bc9000-test\bc9000-test.tpy
Successfully loaded 186 symbols
```

If the symbols are not stored on the PLC, this is the case for bus controllers, then a symbol file needs to be specified. The symbol file is a .tpy file that is generated by the TwinCAT compiler. It is normally located in the same directory as the project file. It is the ibaPDA service that is going to access the symbol file. This means that if the symbol file is located on a network share then you need to specify the UNC path and proper user credentials for it.

On the general tab click the *Select symbols* hyperlink to open the symbol browser.



Symbol: MAIN.loopCounter

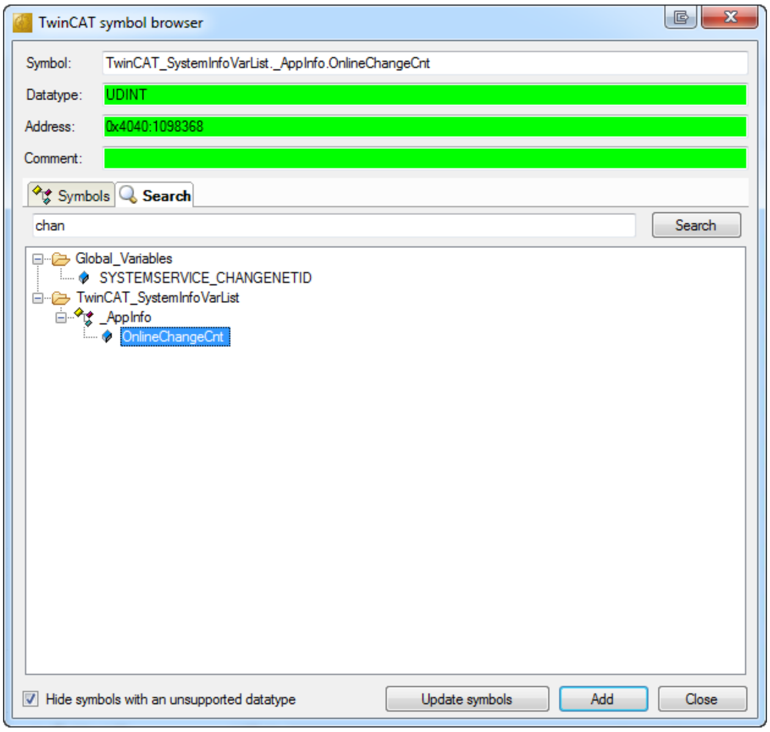
Datatype: DWORD

Address: 0x4040:679892

Comment: Counts every loop

☑ Hide symbols with an unsupported datatype

The symbol browser shows all the symbols that were loaded from the PLC or imported from the symbol file. You can select single or multiple symbols in the tree. Click the *Add* button to add them to the corresponding analog or digital signal grid. If you selected a single symbol then the next symbol will be selected after you clicked the *Add* button. This allows you to hit add multiple times in order to add consecutive symbols. You can also double click a symbol to add it to the signal grid. Use the *Update symbols* button to read the symbols again from the PLC or to read the symbol file again.



Symbol: TwinCAT_SystemInfoVarList._AppInfo.OnlineChangeCnt
Datatype: UDINT
Address: 0x4040:1098368
Comment:

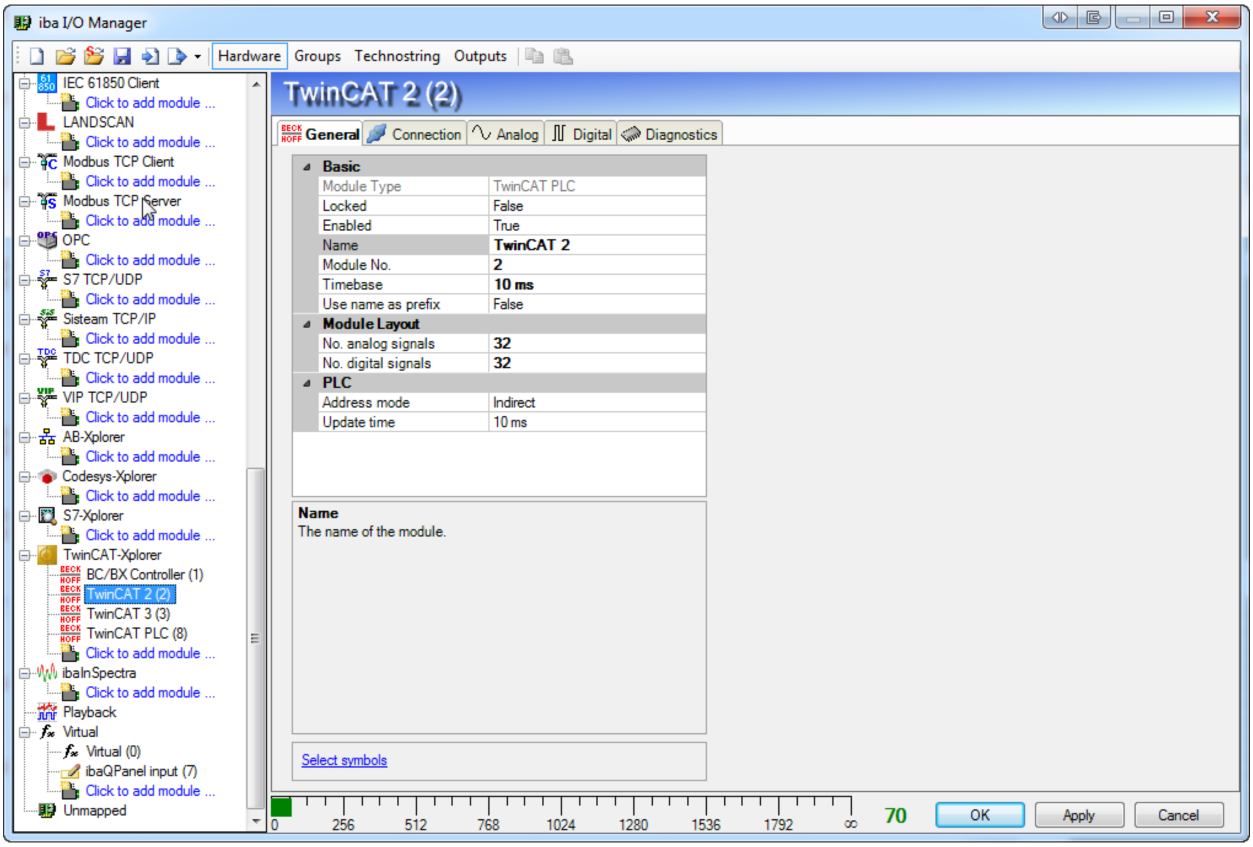
chan

☑ Hide symbols with an unsupported datatype

On the search tab you can search symbols by name. The search result tree works in the same way as the complete symbol tree.

The data type, address and comment of the selected symbol are also shown in the browser. The address consist of an Index Group (in hex) followed by an Index Offset (in decimal). The address of a symbol can change when a project is recompiled or when the project is changed and extra symbols are added or symbols are removed. ibaPDA can read the data for a symbol via the symbol address or via the symbol name. When ibaPDA reads via the symbol name then it first requests a handle to the symbol from the PLC. This handle then points to the actual address of the symbol. When a new version of the project is loaded on the PLC then the addresses that the handles point to are updated automatically by the PLC. This way ibaPDA will always read the correct data for the symbol when it reads via the handle. This addressing mode is called *Indirect*. The other addressing mode is called *Direct*. In direct mode ibaPDA uses the symbol address directly to read data for a symbol. In this case ibaPDA could read incorrect data after a project download to the PLC. You can configure the addressing mode that ibaPDA will use on the *General* tab of the module. It is recommended to use the *Indirect* address mode. Bus controllers don't have symbol information on them so they only support *Direct* address mode.

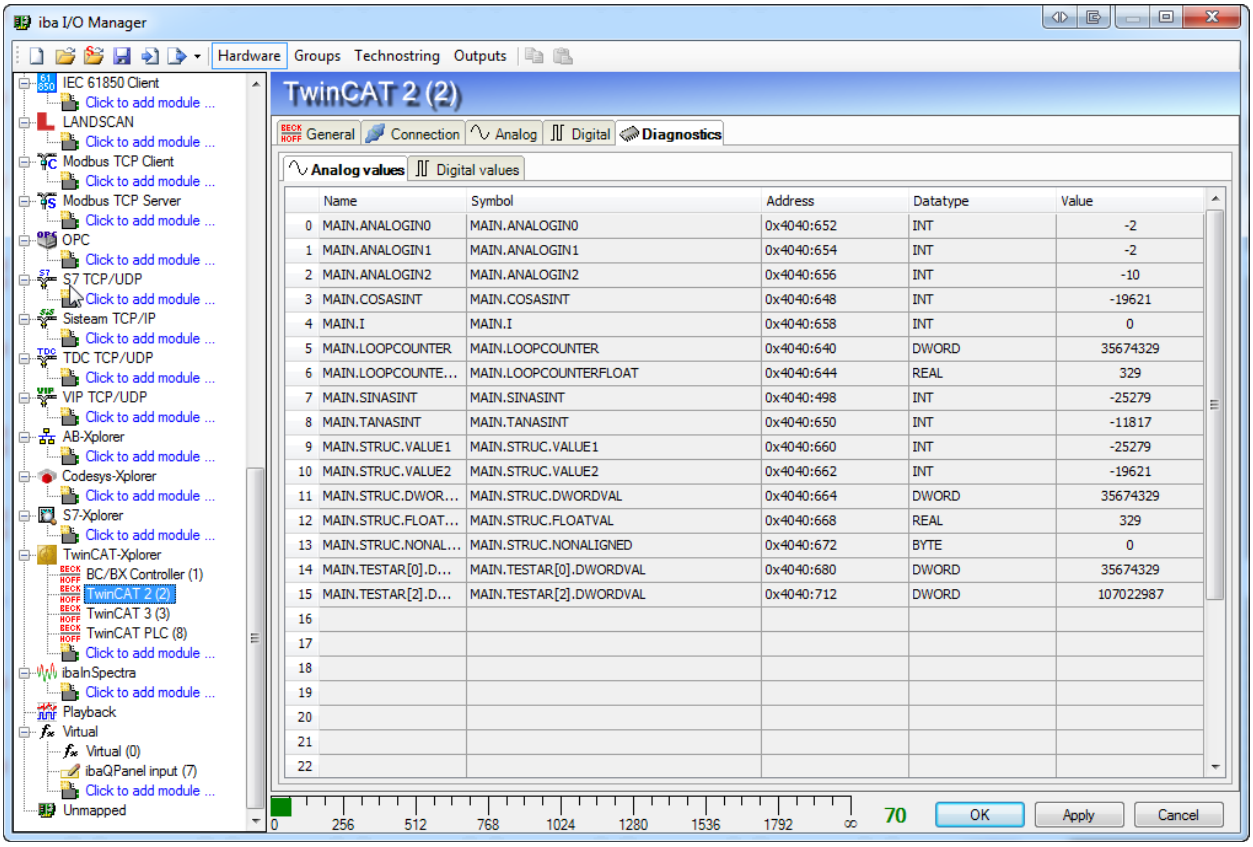
ibaPDA is able to detect that a symbol download occurs on a bus controller. It will then periodically check if the symbol file has updated. When the symbol file is updated then ibaPDA will reload it and update the addresses of the requested symbols. From then on ibaPDA will use the correct addresses again for the symbols and the data will be read correctly.



Basic	
Module Type	TwinCAT PLC
Locked	False
Enabled	True
Name	**TwinCAT 2**
Module No.	**2**
Timebase	**10 ms**
Use name as prefix	False
Module Layout	
No. analog signals	**32**
No. digital signals	**32**
PLC	
Address mode	Indirect
Update time	10 ms

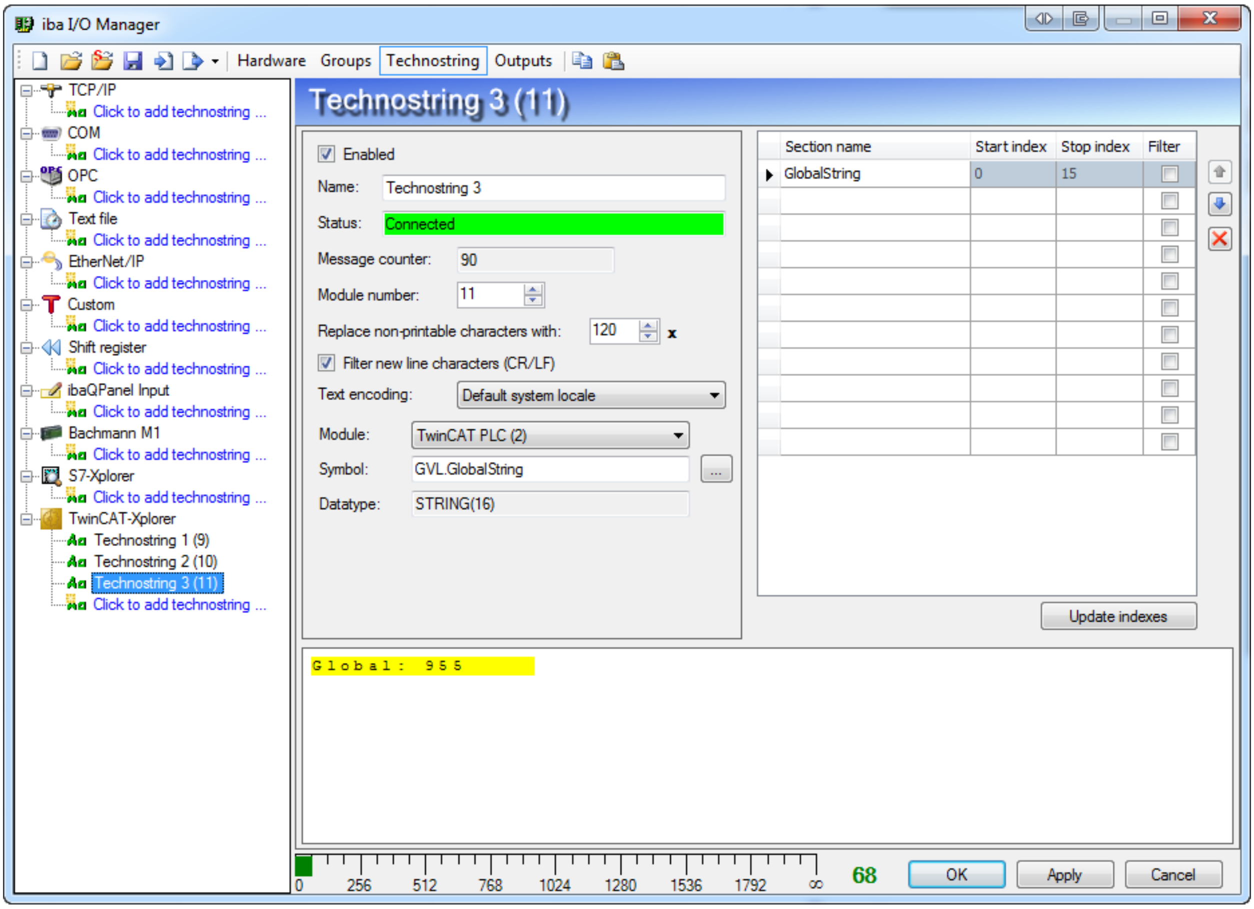
Name
The name of the module.

On the general tab of the module you can also configure the update time. This determines how fast ibaPDA tries to retrieve data from the PLC. The actual update time might be higher if the PLC is unable to deliver the data fast enough. You can check the connections table on the TwinCAT-Xplorer interface to see how fast the data is actually coming in.



	Name	Symbol	Address	Datatype	Value
0	MAIN.ANALOGIN0	MAIN.ANALOGIN0	0x4040:652	INT	-2
1	MAIN.ANALOGIN1	MAIN.ANALOGIN1	0x4040:654	INT	-2
2	MAIN.ANALOGIN2	MAIN.ANALOGIN2	0x4040:656	INT	-10
3	MAIN.COSASINT	MAIN.COSASINT	0x4040:648	INT	-19621
4	MAIN.I	MAIN.I	0x4040:658	INT	0
5	MAIN.LOOPCOUNTER	MAIN.LOOPCOUNTER	0x4040:640	DWORD	35674329
6	MAIN.LOOPCOUNTE...	MAIN.LOOPCOUNTERFLOAT	0x4040:644	REAL	329
7	MAIN.SINASINT	MAIN.SINASINT	0x4040:498	INT	-25279
8	MAIN.TANASINT	MAIN.TANASINT	0x4040:650	INT	-11817
9	MAIN.STRUC.VALUE1	MAIN.STRUC.VALUE1	0x4040:660	INT	-25279
10	MAIN.STRUC.VALUE2	MAIN.STRUC.VALUE2	0x4040:662	INT	-19621
11	MAIN.STRUC.DWOR...	MAIN.STRUC.DWORDVAL	0x4040:664	DWORD	35674329
12	MAIN.STRUC.FLOAT...	MAIN.STRUC.FLOATVAL	0x4040:668	REAL	329
13	MAIN.STRUC.NONAL...	MAIN.STRUC.NONALIGNED	0x4040:672	BYTE	0
14	MAIN.TESTAR[0].D...	MAIN.TESTAR[0].DWORDVAL	0x4040:680	DWORD	35674329
15	MAIN.TESTAR[2].D...	MAIN.TESTAR[2].DWORDVAL	0x4040:712	DWORD	107022987
16					
17					
18					
19					
20					
21					
22					

When you have selected all the symbols you want to measure and set the update time then you can apply the configuration. IbaPDA will then make a connection to the PLC and do periodic reads. The module has a fifth tab called *Diagnostics* that shows the values of all the signals that are currently being read from the PLC. The grey rows are signals that are inactive.


iba I/O Manager
Hardware
Groups
Technostring
Outputs
TCP/IP
Click to add technostring ...
COM
Click to add technostring ...
OPC
Click to add technostring ...
Text file
Click to add technostring ...
EtherNet/IP
Click to add technostring ...
Custom
Click to add technostring ...
Shift register
Click to add technostring ...
ibaQPanel Input
Click to add technostring ...
Bachmann M1
Click to add technostring ...
S7-Xplorer
Click to add technostring ...
TwinCAT-Xplorer
Technostring 1 (9)
Technostring 2 (10)
Technostring 3 (11)
Click to add technostring ...
Technostring 3 (11)
Update indexes
0
256
512
768
1024
1280
1536
1792
∞
68
OK
Apply
Cancel

☑ Enabled

Name: Technostring 3

Status: Connected

Message counter: 90

Module number: 11

Replace non-printable characters with: 120 x

☑ Filter new line characters (CR/LF)

Text encoding: Default system locale

Module: TwinCAT PLC (2)

Symbol: GVL.GlobalString

Datatype: STRING(16)

Section name	Start index	Stop index	Filter
GlobalString	0	15	□

```
Global: 955
```

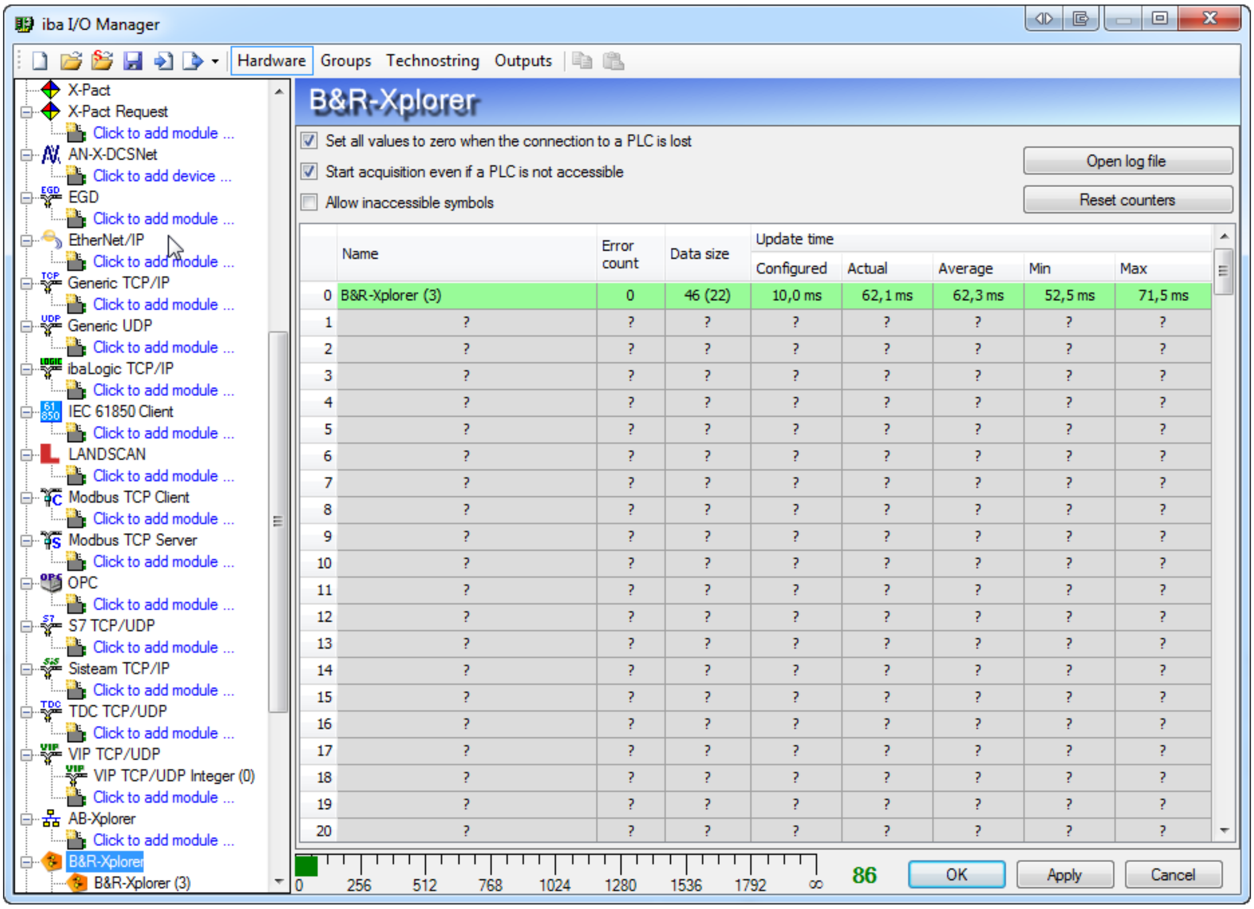
Symbols with data type STRING or WSTRING can be measured as technostring. The TwinCAT-Xplorer technostring uses the connection made by a TwinCAT-Xplorer module to read the string value. This means that the update time of the module is used to read the data. This also means that the response time of the module will be influenced by all technostrings linked to the module. The linked TwinCAT-Xplorer module doesn’t need to have any analog or digital signals configured.

Once you have selected a TwinCAT-Xplorer module you can browse for a symbol with data type STRING or WSTRING. The symbol browser will only show the symbols with compatible data types. Once you have selected a string symbol a technostring section will be added with the maximum size of the symbol and with the same name as the symbol. If you want you can define additional sections to split the received string into multiple parts.

The string value will only be read when the acquisition is running. So when you just added a new technostring you won’t see a preview.

2 B&R-Xplorer

The B&R-Xplorer interface in ibaPDA is used to measure data from B&R PLCs. The interface supports B&R industrial PCs and control systems like the X20 system. It is an Xplorer interface which means that the data is cyclically read by ibaPDA instead of being sent by the PLC. ibaPDA uses the B&R PVI library to communicate with the PLCs.



X-Pact
X-Pact Request
Click to add module ...
AN-X-DCSNet
Click to add device ...
EGD
Click to add module ...
EtherNet/IP
Click to add module ...
Generic TCP/IP
Click to add module ...
Generic UDP
Click to add module ...
ibaLogic TCP/IP
Click to add module ...
IEC 61850 Client
Click to add module ...
LANDSCAN
Click to add module ...
Modbus TCP Client
Click to add module ...
Modbus TCP Server
Click to add module ...
OPC
Click to add module ...
S7 TCP/UDP
Click to add module ...
Sisteam TCP/IP
Click to add module ...
TDC TCP/UDP
Click to add module ...
VIP TCP/UDP
VIP TCP/UDP Integer (0)
Click to add module ...
AB-Xplorer
Click to add module ...
B&R-Xplorer
B&R-Xplorer (3)

☑ Set all values to zero when the connection to a PLC is lost

☑ Start acquisition even if a PLC is not accessible

□ Allow inaccessible symbols

	Name	Error count	Data size	Update time				
				Configured	Actual	Average	Min	Max
0	B&R-Xplorer (3)	0	46 (22)	10,0 ms	62,1 ms	62,3 ms	52,5 ms	71,5 ms
1	?	?	?	?	?	?	?	?
2	?	?	?	?	?	?	?	?
3	?	?	?	?	?	?	?	?
4	?	?	?	?	?	?	?	?
5	?	?	?	?	?	?	?	?
6	?	?	?	?	?	?	?	?
7	?	?	?	?	?	?	?	?
8	?	?	?	?	?	?	?	?
9	?	?	?	?	?	?	?	?
10	?	?	?	?	?	?	?	?
11	?	?	?	?	?	?	?	?
12	?	?	?	?	?	?	?	?
13	?	?	?	?	?	?	?	?
14	?	?	?	?	?	?	?	?
15	?	?	?	?	?	?	?	?
16	?	?	?	?	?	?	?	?
17	?	?	?	?	?	?	?	?
18	?	?	?	?	?	?	?	?
19	?	?	?	?	?	?	?	?
20	?	?	?	?	?	?	?	?

The B&R-Xplorer interface shows a table of the available connections. Per B&R-Xplorer license you get 16 connections. A maximum of 240 connections is allowed. This means that maximum 15 licenses can be used. Each connection corresponds with a row in the table. The row is green when the connection is ok and data is being read. The row is red when the connection could not be established. The row is grey when there is no connection configured. The PVI manager only sends data to ibaPDA when the data changes. The update time is the time between consecutive updates for the same signal. So the update time will depend on how fast the data changes in the PLC and on how fast the PVI manager can read data from the PLC. The table shows the actual, average, minimum and maximum of the update time. It also shows the configured update time. This determines the polling frequency of the PVI manager. The data size shows how many bytes are required to read all signals. The number in brackets shows how many signals are being read. You can use the “Reset counters” button to clear the counters for all connections.

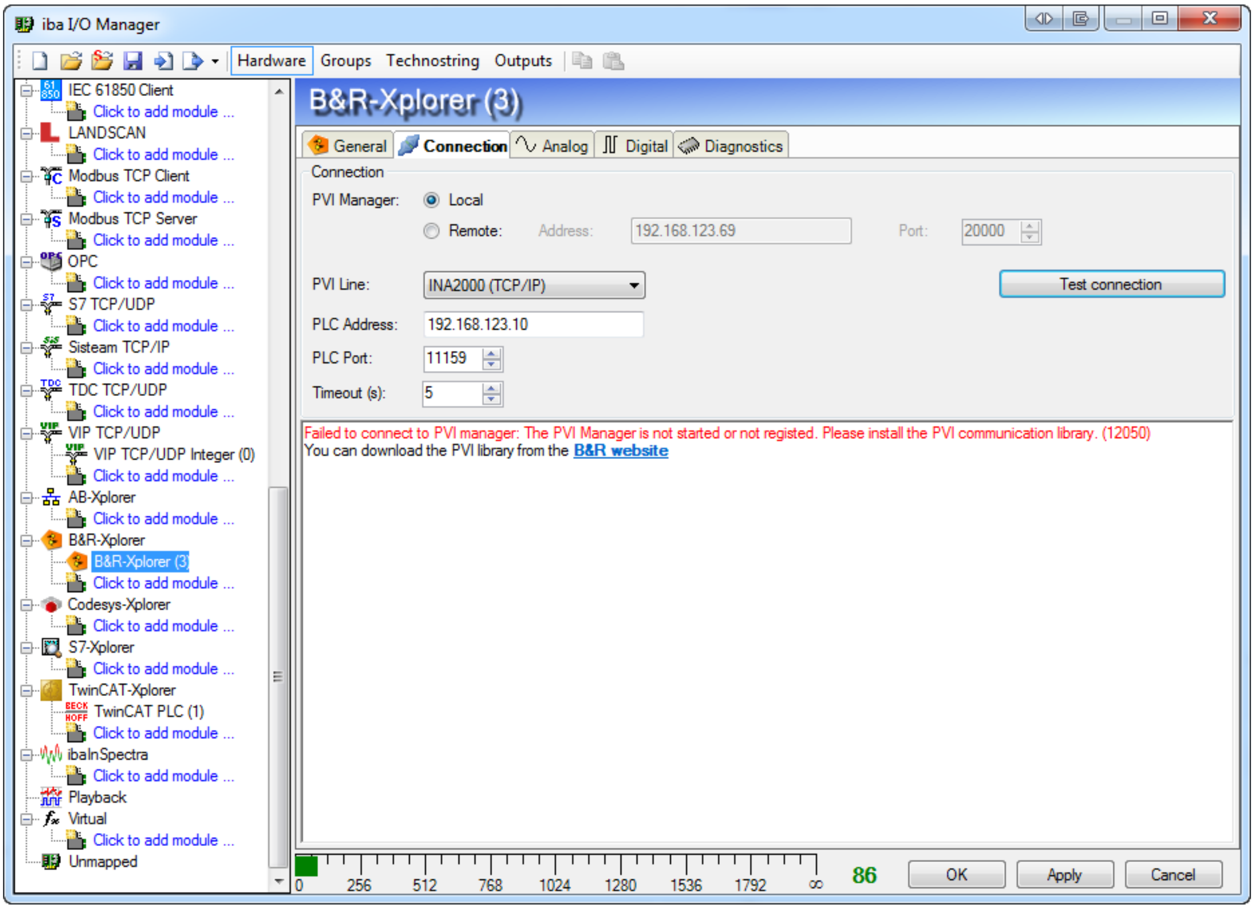
On the interface you can also decide how to handle some error conditions:

- 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 read a symbol that is no longer available in the PLC then the PLC will return an error. If the option “Allow inaccessible symbols” 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.

There is only one type of module that you can add to the B&R-Xplorer interface: B&R-Xplorer.



Connection

PVI Manager: ● Local

○ Remote: Address: 192.168.123.69 Port: 20000

PVI Line: INA2000 (TCP/IP)

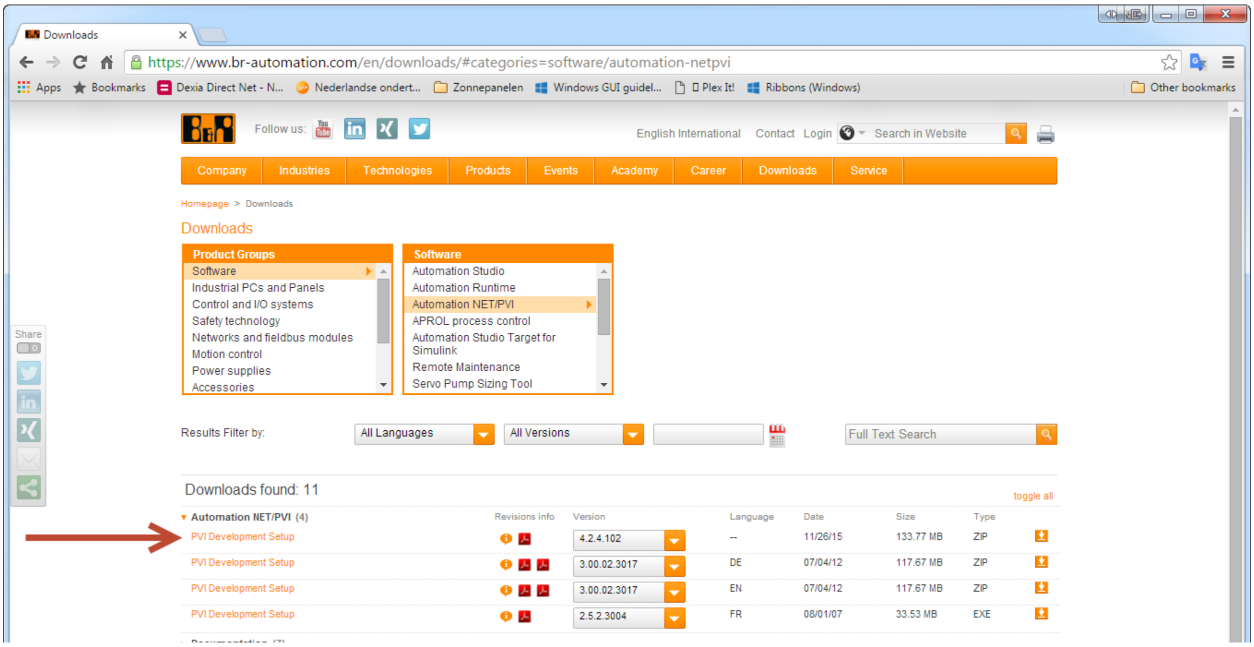
PLC Address: 192.168.123.10

PLC Port: 11159

Timeout (s): 5

Failed to connect to PVI manager: The PVI Manager is not started or not registed. Please install the PVI communication library. (12050)
You can download the PVI library from the B&R website

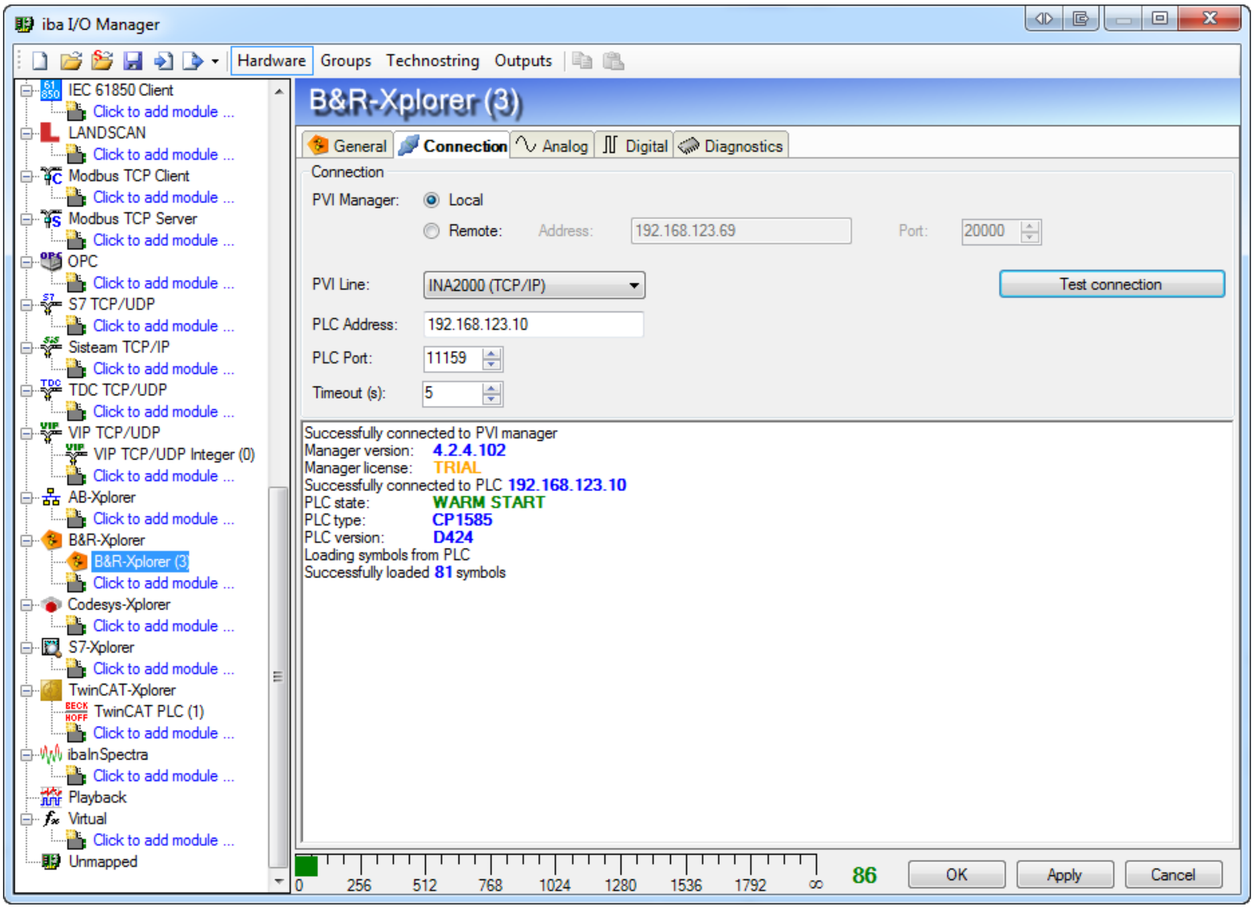
ibaPDA uses the B&R PVI communication library to connect to the PLCs. This library is not included in the ibaPDA installation. If B&R Automation Studio is installed on the PC where the ibaPDA service is running then the library will be available. If Automation Studio is not installed then you will have to download the B&R PVI runtime from the B&R website (http://www.br-automation.com). Navigate to Dowloads | Software | Automation NET/PVI | PVI Development Setup. Start the installer and use the default settings.


Downloads
https://www.br-automation.com/en/downloads/#categories=software/automation-netpvi
Apps Bookmarks Dexia Direct Net - N... Nederlandse ondert... Zonnepanelen Windows GUI guidel... Plex It! Ribbons (Windows) Other bookmarks
Follow us:
English International Contact Login Search in Website
Company Industries Technologies Products Events Academy Career Downloads Service
Homepage > Downloads
Downloads
Product Groups
Software
Industrial PCs and Panels
Control and I/O systems
Safety technology
Networks and fieldbus modules
Motion control
Power supplies
Accessories
Software
Automation Studio
Automation Runtime
Automation NET/PVI
APROL process control
Automation Studio Target for Simulink
Remote Maintenance
Servo Pump Sizing Tool
Results Filter by:
All Languages
All Versions
Full Text Search
Share
toggle all

Downloads found: 11

▾ Automation NET/PVI (4)

	Revisions info	Version	Language	Date	Size	Type
PVI Development Setup		4.2.4.102	--	11/26/15	133.77 MB	ZIP
PVI Development Setup		3.00.02.3017	DE	07/04/12	117.67 MB	ZIP
PVI Development Setup		3.00.02.3017	EN	07/04/12	117.67 MB	ZIP
PVI Development Setup		2.5.2.3004	FR	08/01/07	33.53 MB	EXE



Connection

PVI Manager: ● Local
○ Remote: Address: 192.168.123.69 Port: 20000

PVI Line: INA2000 (TCP/IP)

PLC Address: 192.168.123.10

PLC Port: 11159

Timeout (s): 5

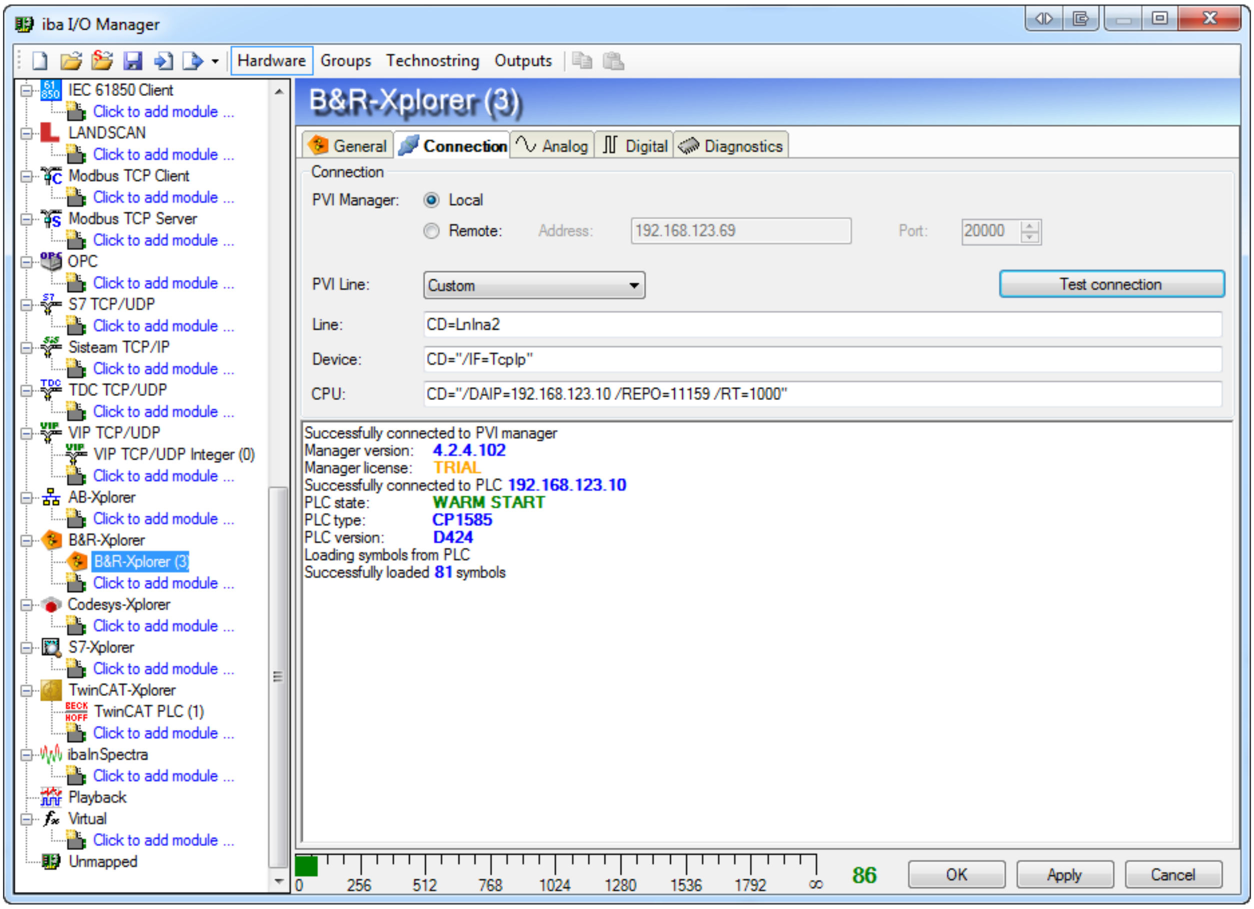
```
Successfully connected to PVI manager
Manager version:   4.2.4.102
Manager license:   TRIAL
Successfully connected to PLC 192.168.123.10
PLC state:         WARM START
PLC type:          CP1585
PLC version:       D424
Loading symbols from PLC
Successfully loaded 81 symbols
```

On the connection tab you have to configure the PVI manager you want to use to connect to the PLC. You can use a local PVI manager or a remote one. If the PVI manager is running on non-B&R hardware then you require a license from B&R to use it.

The next step is configuring the PVI line to use. The PVI line determines which protocol the PVI manager will use to connect to the PLC. There are 3 options:

- INA2000 (TCP/IP): This interface uses the INA2000 protocol over TCP and UDP. It is supported by control generations SGC, SG3 and SG4. The automation runtime version needs to be 2.10 or higher.
- ANSL: This interface is the successor to the INA2000 protocol. It also uses TCP and UDP. It is only supported by control generations SG4. The automation runtime version needs to be 4.08 or higher.
- Custom: In this mode you can configure the PVI line yourself by entering the connection description strings for the line, device and CPU.

In case of the INA2000 and ANSL lines you have to configure the IP address or host name of the PLC. You also have to configure the port number. The timeout determines how long a connection attempt will take.



Connection

PVI Manager: ● Local ○ Remote: Address: 192.168.123.69 Port: 20000

PVI Line: Custom

Line: CD=LnIna2

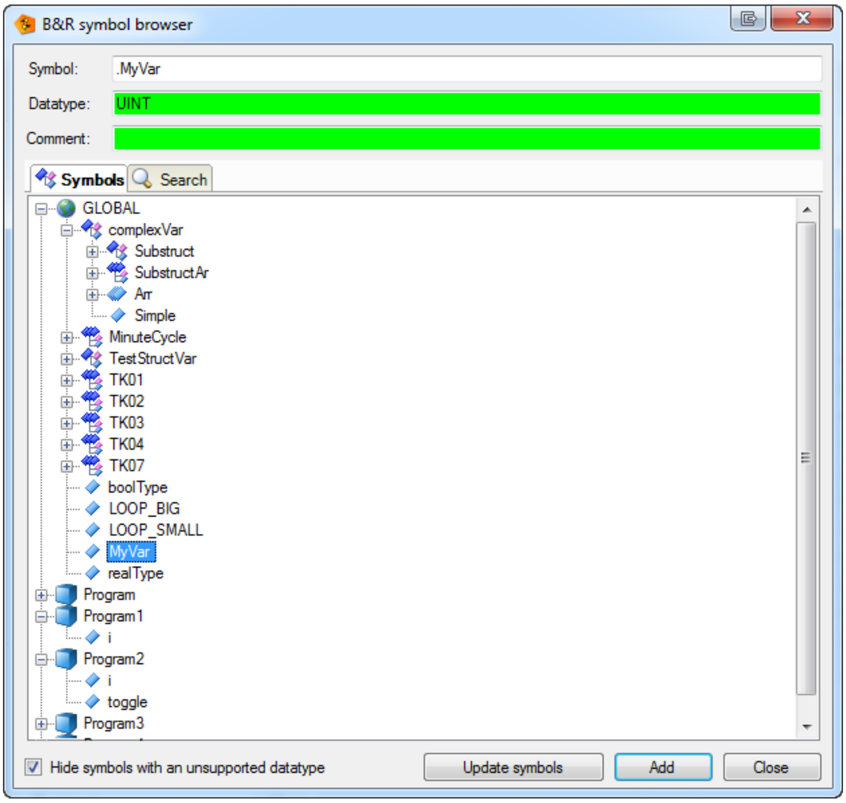
Device: CD="/IF=TcpIp"

CPU: CD="/DAIP=192.168.123.10 /REPO=11159 /RT=1000"

```
Successfully connected to PVI manager
Manager version:   4.2.4.102
Manager license:   TRIAL
Successfully connected to PLC 192.168.123.10
PLC state:         WARM START
PLC type:          CP1585
PLC version:       D424
Loading symbols from PLC
Successfully loaded 81 symbols
```

Use the *Test connection* button on the *Connection* tab to try the connection to the PLC. When you test the connection you get some information about the connected PVI manager and PLC. ibaPDA will also load the symbols that are available on the PLC.

On the general tab click the *Select symbols* hyperlink to open the symbol browser.



B&R symbol browser

Symbol: .MyVar

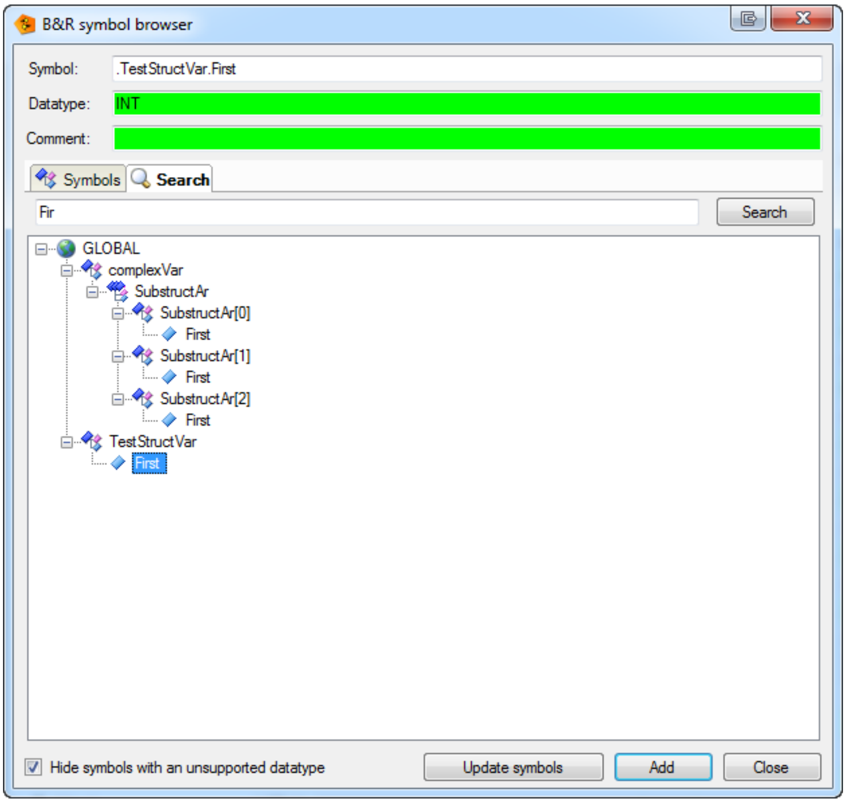
Datatype: UINT

Comment:

```
GLOBAL
  complexVar
    Substruct
    SubstructAr
    Arr
    Simple
  MinuteCycle
  TestStructVar
  TK01
  TK02
  TK03
  TK04
  TK07
  boolType
  LOOP_BIG
  LOOP_SMALL
  MyVar
  realType
Program
Program1
  i
Program2
  i
  toggle
Program3
```

☑ Hide symbols with an unsupported datatype

The symbol browser shows all the symbols that were loaded from the PLC. You can select single or multiple symbols in the tree. Click the *Add* button to add them to the corresponding analog or digital signal grid. If you selected a single symbol then the next symbol will be selected after you clicked the *Add* button. This allows you to hit add multiple times in order to add consecutive symbols. You can also double click a symbol to add it to the signal grid. Use the *Update symbols* button to read the symbols again from the PLC.



B&R symbol browser

Symbol: .TestStructVar.First

Datatype: INT

Comment:

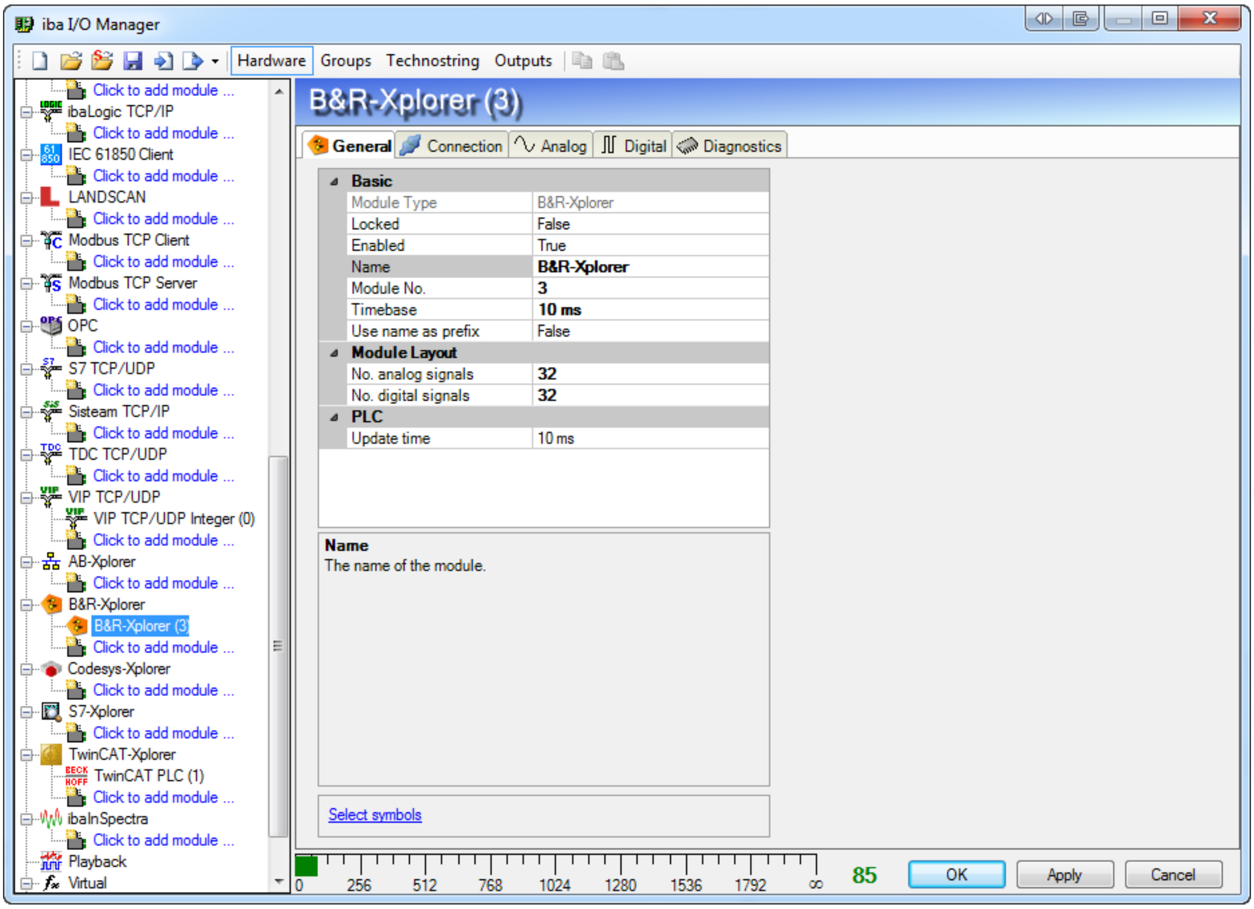
Fir

```
GLOBAL
  complexVar
    SubstructAr
      SubstructAr[0]
        First
      SubstructAr[1]
        First
      SubstructAr[2]
        First
  TestStructVar
    First
```

☑ Hide symbols with an unsupported datatype

On the search tab you can search symbols by name. The search result tree works in the same way as the complete symbol tree.

The data type and comment of the selected symbol are also shown in the browser

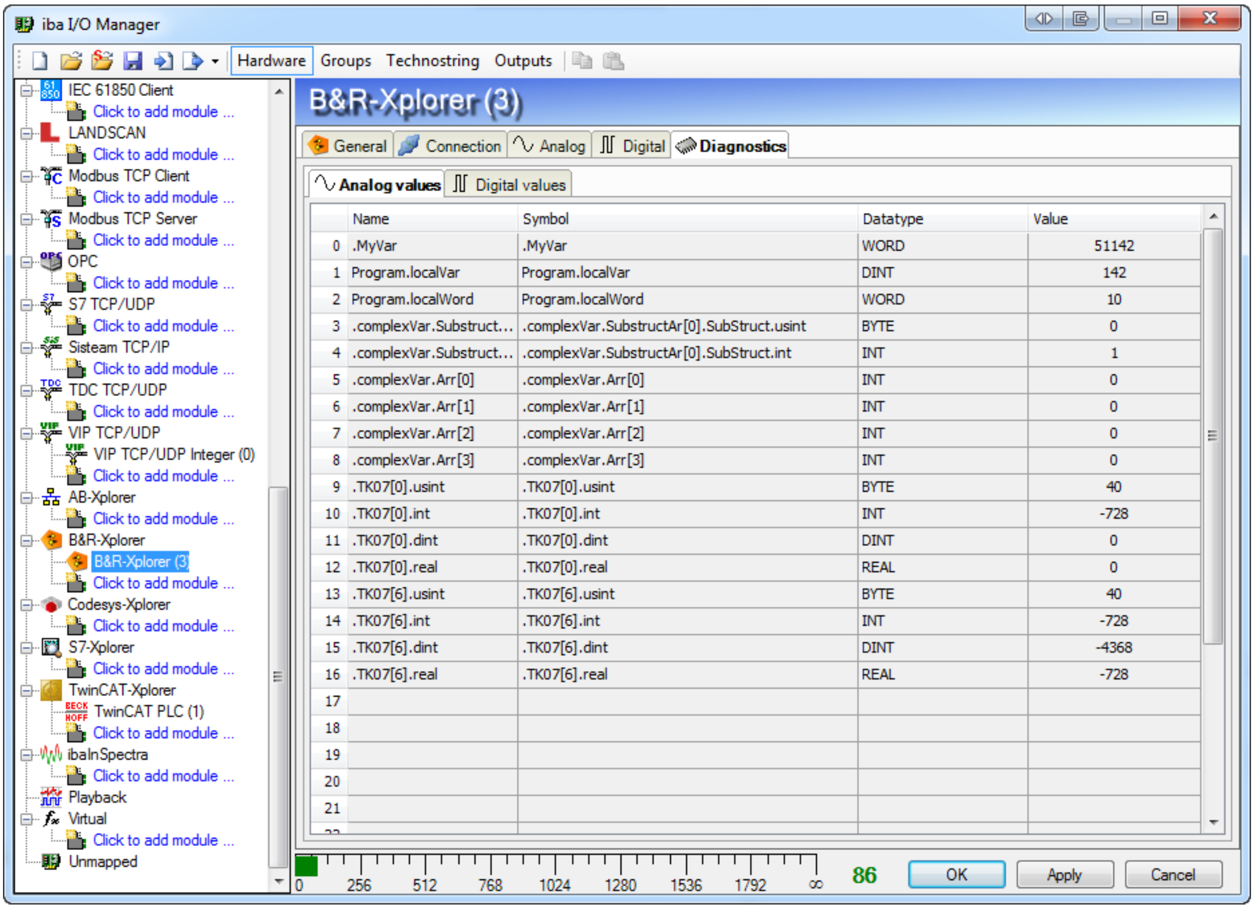


Basic	
Module Type	B&R-Xplorer
Locked	False
Enabled	True
Name	**B&R-Xplorer**
Module No.	**3**
Timebase	**10 ms**
Use name as prefix	False
Module Layout	
No. analog signals	**32**
No. digital signals	**32**
PLC	
Update time	10 ms

Name
The name of the module.

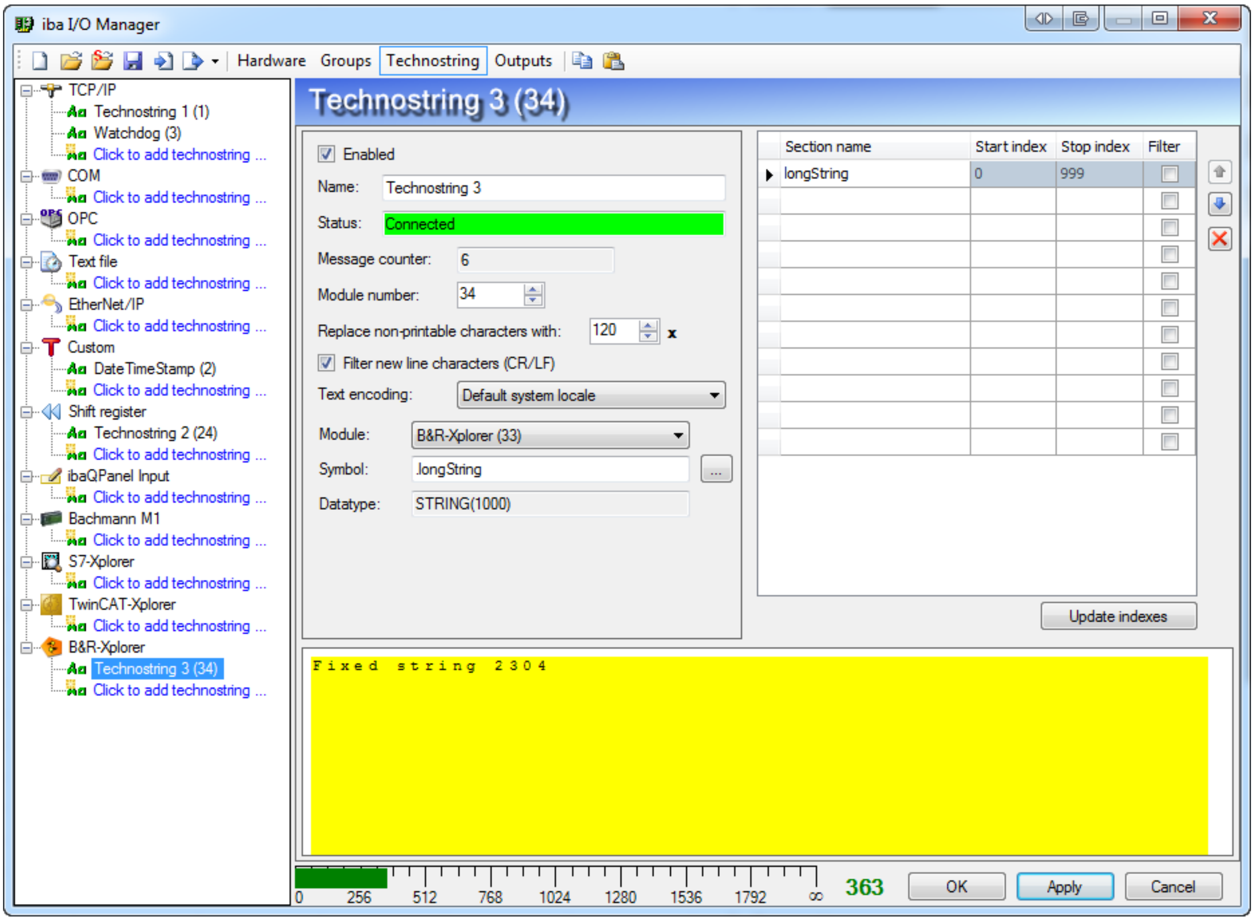
Select symbols

On the general tab of the module you can configure the update time. This determines how fast the PVI manager tries to retrieve data from the PLC. The actual update time might be higher if the PLC is unable to deliver the data fast enough. You can check the connections table on the B&R-Xplorer interface to see how fast the data is actually coming in.



	Name	Symbol	Datatype	Value
0	.MyVar	.MyVar	WORD	51142
1	Program.localVar	Program.localVar	DINT	142
2	Program.localWord	Program.localWord	WORD	10
3	.complexVar.Substruct...	.complexVar.SubstructAr[0].SubStruct.usint	BYTE	0
4	.complexVar.Substruct...	.complexVar.SubstructAr[0].SubStruct.int	INT	1
5	.complexVar.Arr[0]	.complexVar.Arr[0]	INT	0
6	.complexVar.Arr[1]	.complexVar.Arr[1]	INT	0
7	.complexVar.Arr[2]	.complexVar.Arr[2]	INT	0
8	.complexVar.Arr[3]	.complexVar.Arr[3]	INT	0
9	.TK07[0].usint	.TK07[0].usint	BYTE	40
10	.TK07[0].int	.TK07[0].int	INT	-728
11	.TK07[0].dint	.TK07[0].dint	DINT	0
12	.TK07[0].real	.TK07[0].real	REAL	0
13	.TK07[6].usint	.TK07[6].usint	BYTE	40
14	.TK07[6].int	.TK07[6].int	INT	-728
15	.TK07[6].dint	.TK07[6].dint	DINT	-4368
16	.TK07[6].real	.TK07[6].real	REAL	-728
17				
18				
19				
20				
21				

When you have selected all the symbols you want to measure and set the update time then you can apply the configuration. IbaPDA will then make a connection to the PLC and do periodic reads. The module has a fifth tab called *Diagnostics* that shows the values of all the signals that are currently being read from the PLC. The grey rows are signals that are inactive.


iba I/O Manager
Hardware
Groups
Technostring
Outputs
TCP/IP
Technostring 1 (1)
Watchdog (3)
Click to add technostring ...
COM
Click to add technostring ...
OPC
Click to add technostring ...
Text file
Click to add technostring ...
EtherNet/IP
Click to add technostring ...
Custom
DateTimeStamp (2)
Click to add technostring ...
Shift register
Technostring 2 (24)
Click to add technostring ...
ibaQPanel Input
Click to add technostring ...
Bachmann M1
Click to add technostring ...
S7-Xplorer
Click to add technostring ...
TwinCAT-Xplorer
Click to add technostring ...
B&R-Xplorer
Technostring 3 (34)
Click to add technostring ...
Update indexes
0
256
512
768
1024
1280
1536
1792
∞
363
OK
Apply
Cancel

Technostring 3 (34)

☑ Enabled

Name: Technostring 3

Status: Connected

Message counter: 6

Module number: 34

Replace non-printable characters with: 120 x

☑ Filter new line characters (CR/LF)

Text encoding: Default system locale

Module: B&R-Xplorer (33)

Symbol: .longString

Datatype: STRING(1000)

Section name	Start index	Stop index	Filter
longString	0	999	□

```
Fixed string 2304
```

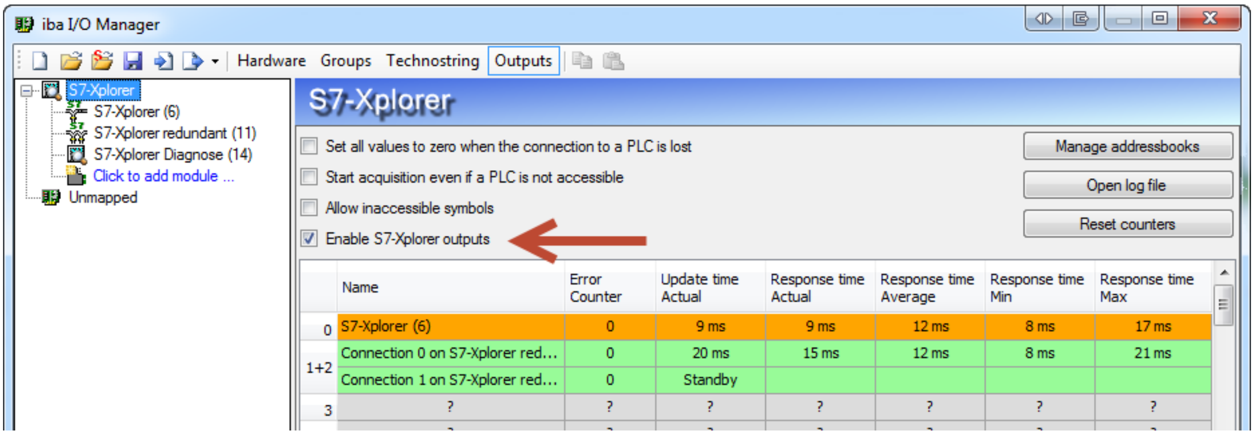
Symbols with data type STRING or WSTRING can be measured as technostring. The B&R-Xplorer technostring uses the connection made by a B&R-Xplorer module to read the string value. This means that the update time of the module is used to read the data. This also means that the actual update time of the module will be influenced by all technostrings linked to the module. The linked B&R-Xplorer module doesn't need to have any analog or digital signals configured.

Once you have selected a B&R-Xplorer module you can browse for a symbol with data type STRING or WSTRING. The symbol browser will only show the symbols with compatible data types. Once you have selected a string symbol a technostring section will be added with the maximum size of the symbol and with the same name as the symbol. If you want you can define additional sections to split the received string into multiple parts.

The string value will only be read when the acquisition is running. So when you just added a new technostring you won't see a preview.

3 S7-Xplorer outputs

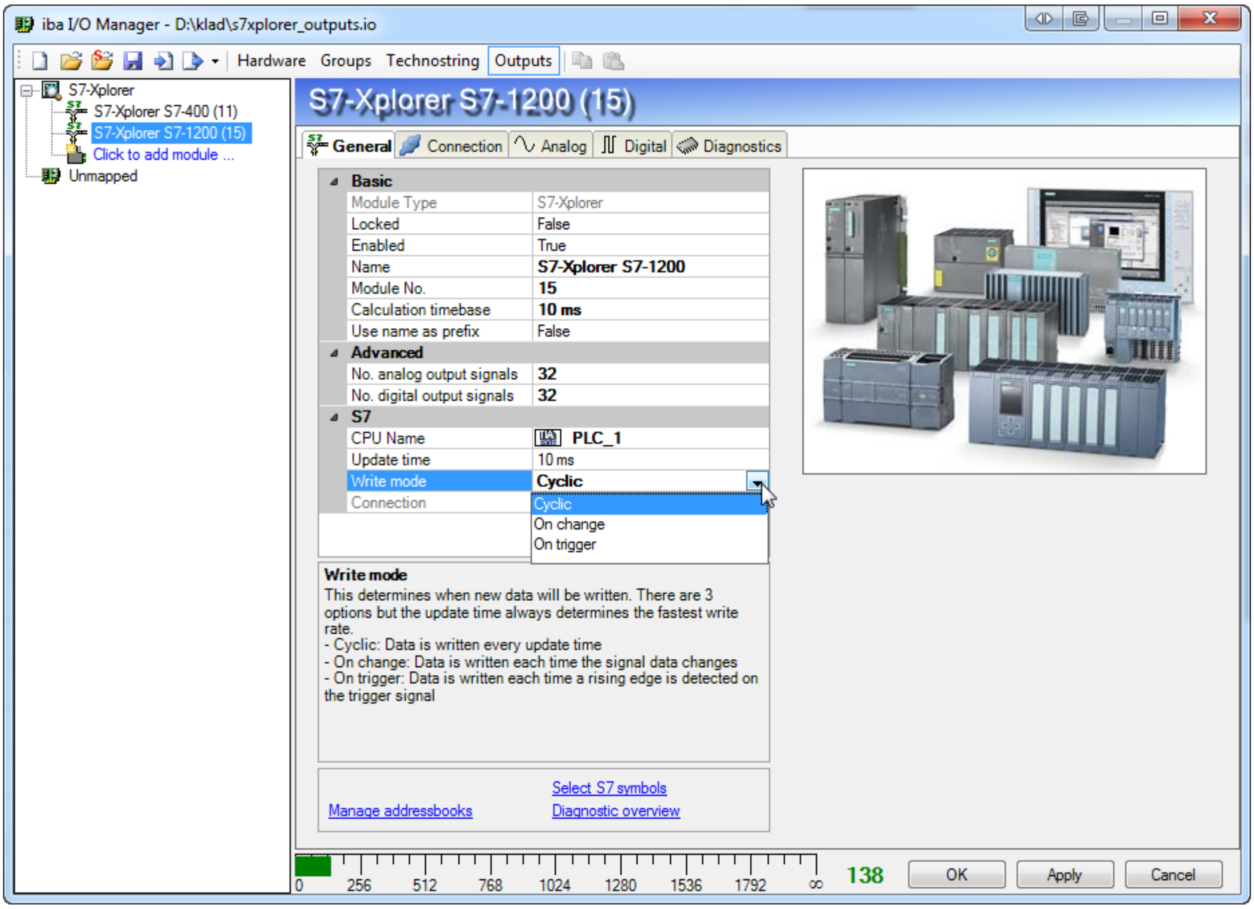
The S7-Xplorer module can now also write back data to a connected PLC. Since this is a potentially dangerous feature you first have to enable it on the S7-Xplorer interface.



□ Set all values to zero when the connection to a PLC is lost
□ Start acquisition even if a PLC is not accessible
□ Allow inaccessible symbols
☑ Enable S7-Xplorer outputs

	Name	Error Counter	Update time Actual	Response time Actual	Response time Average	Response time Min	Response time Max
0	S7-Xplorer (6)	0	9 ms	9 ms	12 ms	8 ms	17 ms
1+2	Connection 0 on S7-Xplorer red...	0	20 ms	15 ms	12 ms	8 ms	21 ms
	Connection 1 on S7-Xplorer red...	0	Standby				
3	?	?	?	?	?	?	?

Select the *Outputs* section in the I/O manager. On the general tab of the S7-Xplorer module you can configure the number of analog and digital output signals.



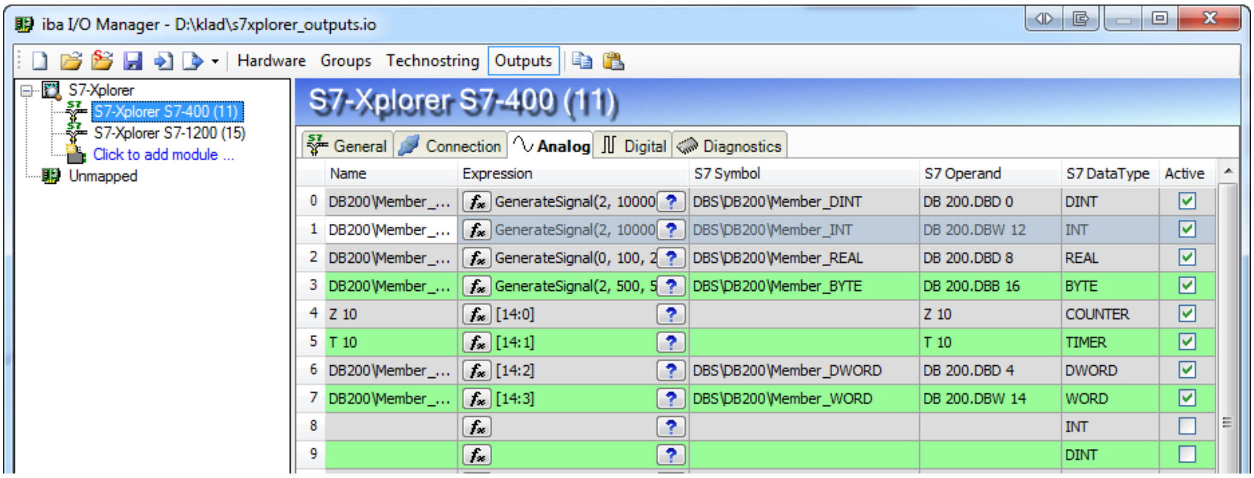
Basic	
Module Type	S7-Xplorer
Locked	False
Enabled	True
Name	**S7-Xplorer S7-1200**
Module No.	**15**
Calculation timebase	**10 ms**
Use name as prefix	False
Advanced	
No. analog output signals	**32**
No. digital output signals	**32**
S7	
CPU Name	**PLC_1**
Update time	10 ms
Write mode	**Cyclic**
Connection	

Write mode
This determines when new data will be written. There are 3 options but the update time always determines the fastest write rate.
- Cyclic: Data is written every update time
- On change: Data is written each time the signal data changes
- On trigger: Data is written each time a rising edge is detected on the trigger signal

There is a single connection used for both reading and writing data from/to the PLC. So the more you write to the PLC the slower the data will be read. You can determine when data is written to the PLC by setting the *Write mode* property. These are the possible modes:

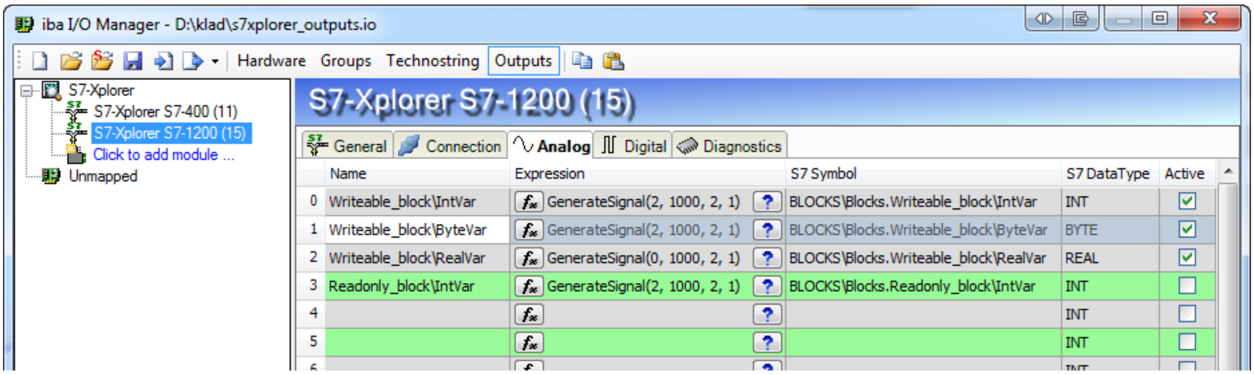
- Cyclic: Data is written every update time. So for every read also a write is done.
- On change: Data is only written when one of the values of the output signals has changed.

- On trigger: Data is only written when a rising edge on the trigger signal is detected. The values of the output signals are taken at the trigger timestamp.



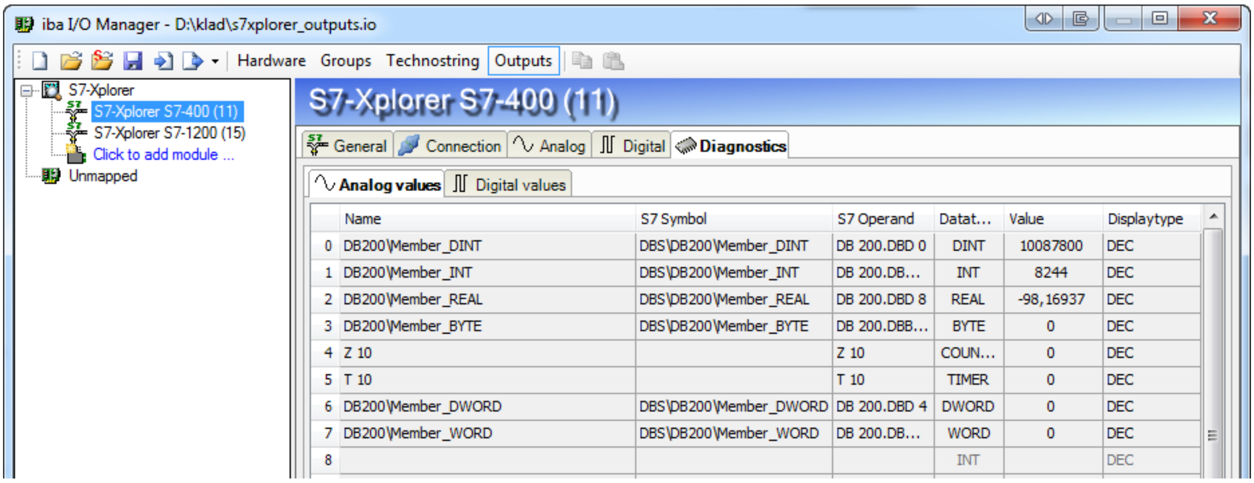
	Name	Expression	S7 Symbol	S7 Operand	S7 DataType	Active
0	DB200\Member_...	GenerateSignal(2, 10000	DBS\DB200\Member_DINT	DB 200.DBD 0	DINT	☑
1	DB200\Member_...	GenerateSignal(2, 10000	DBS\DB200\Member_INT	DB 200.DBW 12	INT	☑
2	DB200\Member_...	GenerateSignal(0, 100, 2	DBS\DB200\Member_REAL	DB 200.DBD 8	REAL	☑
3	DB200\Member_...	GenerateSignal(2, 500, 5	DBS\DB200\Member_BYTE	DB 200.DBB 16	BYTE	☑
4	Z 10	[14:0]		Z 10	COUNTER	☑
5	T 10	[14:1]		T 10	TIMER	☑
6	DB200\Member_...	[14:2]	DBS\DB200\Member_DWORD	DB 200.DBD 4	DWORD	☑
7	DB200\Member_...	[14:3]	DBS\DB200\Member_WORD	DB 200.DBW 14	WORD	☑
8					INT	☐
9					DINT	☐

On the analog and digital tabs you can configure the S7 operands or S7 symbols that need to be written to. When using S7 operands you also have to configure the data type. In the expression column you configure the value that will be written to the PLC. You can enter any expression you want. It will be calculated using the calculation timebase on the general tab. The calculation timebase is the same as the sample timebase of the input signals.



	Name	Expression	S7 Symbol	S7 DataType	Active
0	Writeable_block\IntVar	GenerateSignal(2, 1000, 2, 1)	BLOCKS\Blocks.Writeable_block\IntVar	INT	☑
1	Writeable_block\ByteVar	GenerateSignal(2, 1000, 2, 1)	BLOCKS\Blocks.Writeable_block\ByteVar	BYTE	☑
2	Writeable_block\RealVar	GenerateSignal(0, 1000, 2, 1)	BLOCKS\Blocks.Writeable_block\RealVar	REAL	☑
3	Readonly_block\IntVar	GenerateSignal(2, 1000, 2, 1)	BLOCKS\Blocks.Readonly_block\IntVar	INT	☐
4				INT	☐
5				INT	☐

Writing is also supported when using the TCP/IP S7-1x00 connection mode. In this case you have to select the S7 symbols you want to write to.



	Name	S7 Symbol	S7 Operand	Datat...	Value	Displaytype
0	DB200\Member_DINT	DBS\DB200\Member_DINT	DB 200.DBD 0	DINT	10087800	DEC
1	DB200\Member_INT	DBS\DB200\Member_INT	DB 200.DB...	INT	8244	DEC
2	DB200\Member_REAL	DBS\DB200\Member_REAL	DB 200.DBD 8	REAL	-98,16937	DEC
3	DB200\Member_BYTE	DBS\DB200\Member_BYTE	DB 200.DBB...	BYTE	0	DEC
4	Z 10		Z 10	COUN...	0	DEC
5	T 10		T 10	TIMER	0	DEC
6	DB200\Member_DWORD	DBS\DB200\Member_DWORD	DB 200.DBD 4	DWORD	0	DEC
7	DB200\Member_WORD	DBS\DB200\Member_WORD	DB 200.DB...	WORD	0	DEC
8				INT		DEC

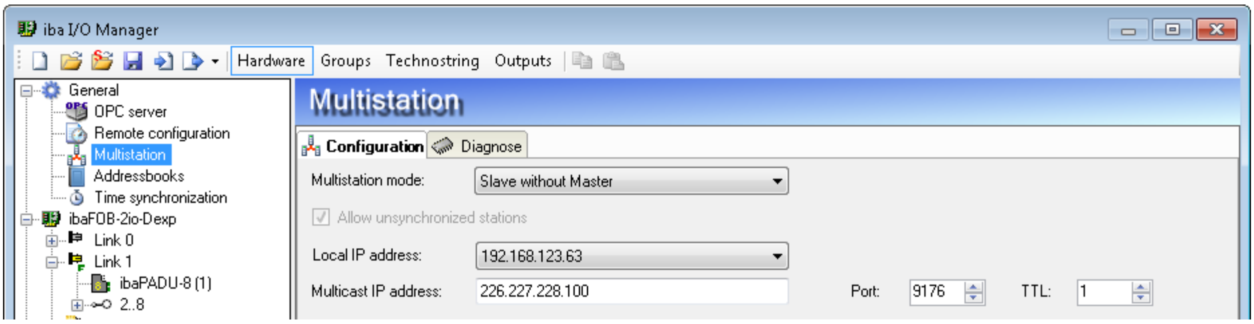
The diagnostics tab shows the last values that have been written to the PLC.

4 Multistation: unsynchronized stations

The multistation system allows the synchronous recording of data on multiple ibaPDA stations. The different ibaPDA stations are coupled via a FO (fiber optic) connection and via the network. They start and stop simultaneously and they can exchange triggers. The data in the dat files generated by the different ibaPDA stations is perfectly aligned.

This multistation system has been extended in ibaPDA v6.36.0 with the concept of unsynchronized stations. Unsynchronized stations are only connected to the other stations via the network. They don't have a FO connection. This means that they perform the data acquisition asynchronous from each other. They don't start and stop together. The only thing they can do is exchange triggers.

Unsynchronized stations can be configured in *Master* and in *Slave* mode. There is also a new mode called *Slave without master* that only has unsynchronized stations.



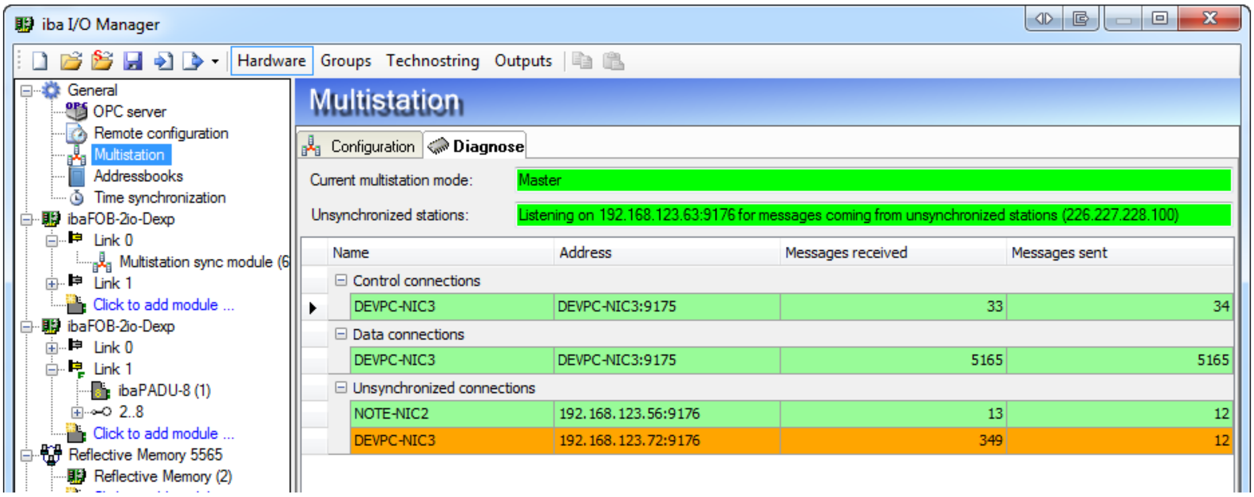
Multistation mode: Slave without Master

☑ Allow unsynchronized stations

Local IP address: 192.168.123.63

Multicast IP address: 226.227.228.100 Port: 9176 TTL: 1

The unsynchronized stations communicate with each other via UDP multicast. You first have to configure which network card you want to use to communicate with the unsynchronized stations. You do this by selecting the local IP address. The list contains all the IP addresses configured on the ibaPDA server. You have to configure the *multicast IP address* and the *port* number that the stations should use to exchange multicast messages. The *TTL* (Time To Live) parameter determines how many routers the multicast messages can pass.



Current multistation mode: Master

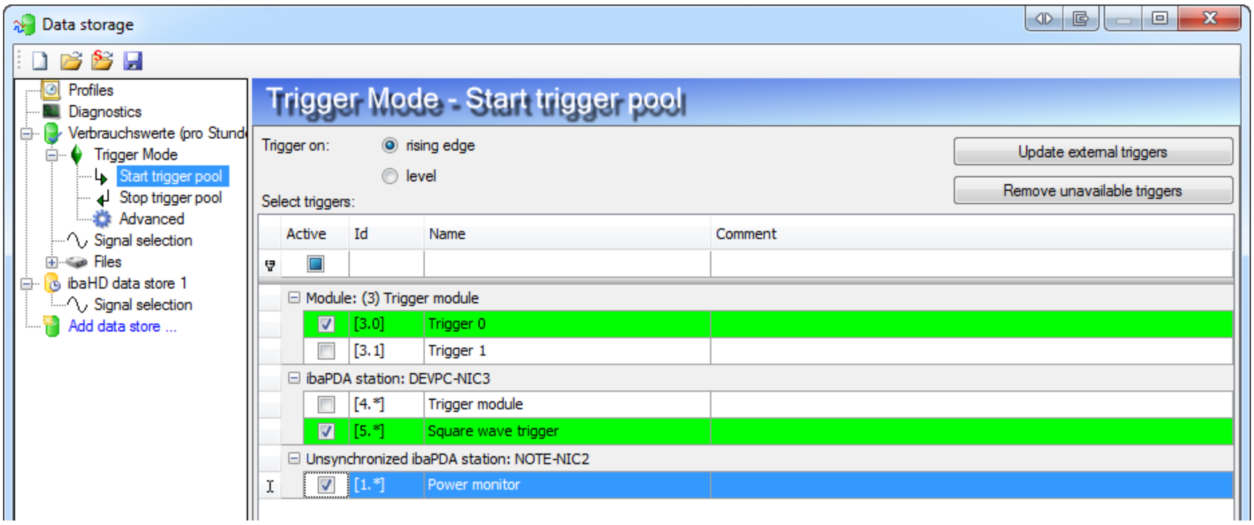
Unsynchronized stations: Listening on 192.168.123.63:9176 for messages coming from unsynchronized stations (226.227.228.100)

Name	Address	Messages received	Messages sent
Control connections			
DEVPC-NIC3	DEVPC-NIC3:9175	33	34
Data connections			
DEVPC-NIC3	DEVPC-NIC3:9175	5165	5165
Unsynchronized connections			
NOTE-NIC2	192.168.123.56:9176	13	12
DEVPC-NIC3	192.168.123.72:9176	349	12

On the Diagnose tab you can see all connections that are made. There are 3 types:

- Control connections: These are connections between the master and its synchronized slave stations. They are used to start/stop the acquisition synchronized and to exchange trigger definitions.
- Data connections: These are connections between synchronized stations (both master and slave) that are used to exchange trigger events. These connections only exist when the acquisition is running.

- Unsynchronized connections: These are connections between unsynchronized stations. They are used to exchange trigger definitions and trigger events. The unsynchronized connections have a green background when they are actively being used to exchange trigger events. They have an orange background when they are not being used to exchange trigger events because e.g. there is already a data connection to the same station or the trigger pools in the data stores aren’t configured to react on triggers from that station.



Trigger on: ● rising edge ○ level

Select triggers:

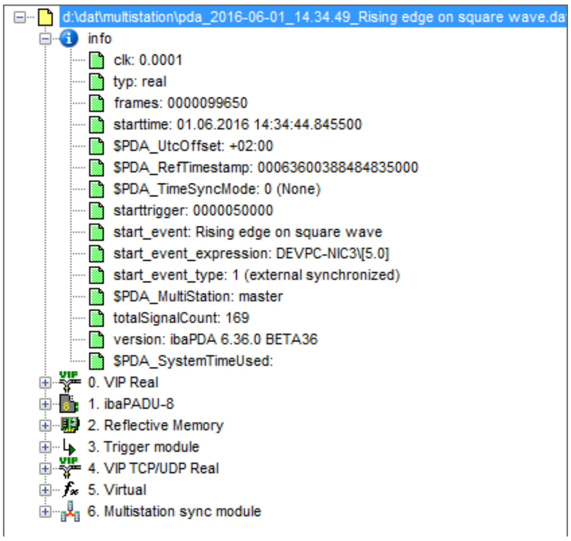
Active	Id	Name	Comment
■			
Module: (3) Trigger module			
☑	[3.0]	Trigger 0	
□	[3.1]	Trigger 1	
ibaPDA station: DEVPC-NIC3			
□	[4.*]	Trigger module	
☑	[5.*]	Square wave trigger	
Unsynchronized ibaPDA station: NOTE-NIC2			
☑	[1.*]	Power monitor	

In the trigger pool configuration of a data store you can see the list of available triggers. There are 3 types of triggers:

- Local triggers: These are the triggers coming from the trigger modules of the I/O configuration of the local ibaPDA station.
- Synchronized external triggers: These are triggers coming from global trigger modules configured in the I/O configurations of the other synchronized ibaPDA stations.
- Unsynchronized external triggers: These are triggers coming from global trigger modules configured in the I/O configurations of the other unsynchronized ibaPDA stations.

The triggers of the synchronized stations are automatically updated when the acquisition is started. The unsynchronized stations and their triggers are not updated automatically. You have to manually click the *Update external triggers* button to let ibaPDA search for unsynchronized stations and their trigger definitions.

When a trigger event occurs on a station that has unsynchronized stations support enabled then it will send the event to the multicast address so that all other stations are able to receive it. The source station doesn’t need to know all the receiver stations. The event message contains the name of the event, its ID (=module number and signal number) and the absolute UTC time when the event occurred. The unsynchronized stations that receive such an event compare the received absolute UTC time with their own UTC system time to synchronize the event with their own acquired data. It is very important that the system time of all unsynchronized stations is accurate. The more accurate it is the better the trigger event can be synchronized to the local data.



```
d:\dat\multistation\pda_2016-06-01_14.34.49_Rising edge on square wave.da
  info
    clk: 0.0001
    typ: real
    frames: 0000099650
    starttime: 01.06.2016 14:34:44.845500
    $PDA_UtcOffset: +02:00
    $PDA_RefTimestamp: 00063600388484835000
    $PDA_TimeSyncMode: 0 (None)
    starttrigger: 0000050000
    start_event: Rising edge on square wave
    start_event_expression: DEVPC-NIC3\[5.0]
    start_event_type: 1 (external synchronized)
    $PDA_MultiStation: master
    totalSignalCount: 169
    version: ibaPDA 6.36.0 BETA36
    $PDA_SystemTimeUsed:
  0. VIP Real
  1. ibaPADU-8
  2. Reflective Memory
  3. Trigger module
  4. VIP TCP/UDP Real
  5. Virtual
  6. Multistation sync module
```

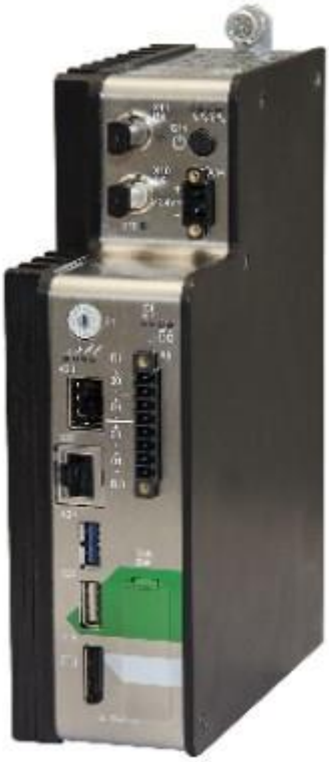
There is a new infofield added to the dat file. It is called start_event_type or stop_event_type for stop triggers. It determines what type of trigger was used to trigger this dat file. The possible values are:

- 0 (local): This is a local trigger.
- 1 (external synchronized): This is a synchronized external trigger.
- 2 (external unsynchronized): This is an unsynchronized external trigger whose timestamp was able to be synchronized to the local data.
- 3 (external unsynchronized invalid time): This is an unsynchronized external trigger whose timestamp differed more than 5 seconds from the current UTC system time when it was received. A dat file has been generated but the trigger timestamp was changed to the current UTC system time.

The new unsynchronized station feature can also be used as a fallback mechanism for regular synchronized stations in case of a failure. In order for this to work all stations in a multistation system have to have unsynchronized station support enabled. When a slave station starts without a master connection then it will automatically switch to slave without master mode. In this mode it will still try to exchange trigger events with the other stations. When a slave starts and there is no FO connection to the master then it will also automatically switch to slave without master mode.

5 ibaDAQ-S

The ibaDAQ-S is a new type of central unit for the modular ibaPADU-S system. The main difference with other ibaPADU-S central units is that it is an integrated measurement system running ibaPDA server.



In essence, the ibaDAQ-S is a fanless PC running Windows 7 Embedded so when attaching a monitor using the DisplayPort (X26) you will see the standard Windows desktop.

5.1 Hardware

The following connectors, LEDs are currently available on the ibaDAQ-S:

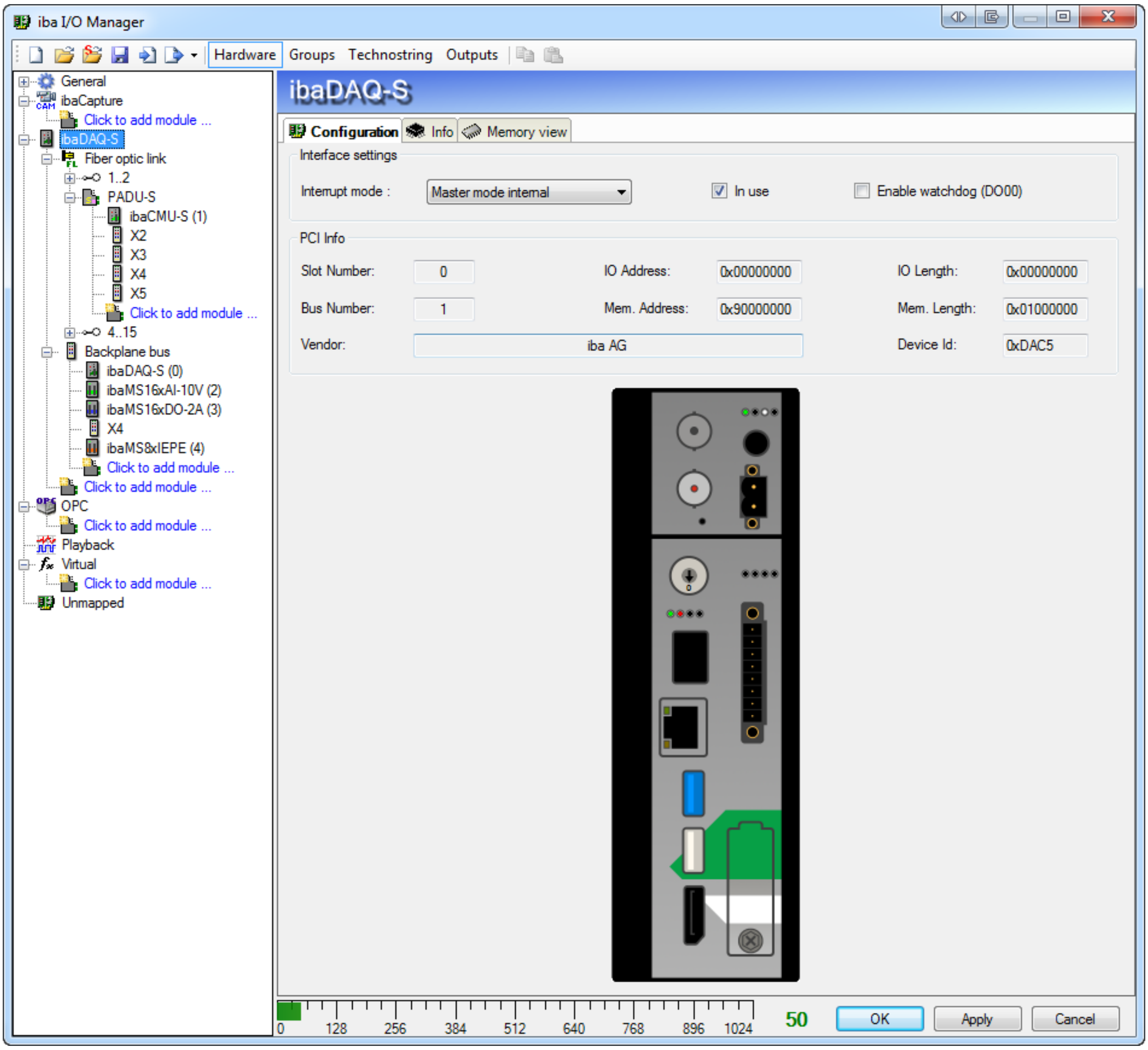
- X10: transmitconnector for Fiber optic link
- X11: receive connector for Fiber optic link
- Run LED (green): this LED blinks from the moment the ibaDAQ-S is powered up
- PDA LED (orange): will be illuminated when the ibaPDA service is running. The LED will blink when S10 is pressed during power up (see also S10).
- Flex LED (white): this is illuminated when the Fiber optic link is configured and available. It will blink when 3 Mbit/32 Mbit input is available
- Error LED (red): in case of an FPGA error, this LED will start blinking. Contact iba support in case it does.
- S11: power button with similar functionality to that of the power button on a computer
- X14: 24V DC power supply input

- S10: when pressed during power up, automatic start of the ibaPDA server acquisition and the automatic shutdown functionality (see also DI00) will be disabled. The button should be pressed until the orange PDA LED starts blinking. The button can be released afterwards.
  When pressed when the system is running, pushing this button for more than 3 seconds will cause a system reboot after closing all applications (including the ibaPDA server acquisition) properly
- S1: controls the IP address of the network adapter related to X22
  - 0: The IP address can be configured using the ibaPDA client
  - 1: Get IP address from DHCP server
  - 2: IP address is set to 192.168.1.1 with subnet mask 255.255.255.0
  - 3-F: Currently unused
- X5: connector for 2 digital inputs and 2 digital outputs. The status of each input or output can be checked using the 4 LEDs above the connector.
  - DI00: the first of two digital inputs on the ibaDAQ-S module. This input can also be used to automatically shutdown the ibaDAQ-S (see also 5.2) (e.g. this input could be triggered by a UPS module in case of a power supply failure).
  - DI01: the second digital input
  - DO00: the first of two digital outputs on the ibaDAQ-S module. This output can also be configured to generate a signal when a watchdog alarm is generated (see also 5.2).
  - DO01: the second digital output
- Lic LED: shows the status of the ibaPDA license
  - Off: undefined
  - Red: no dongle/license
  - Yellow: a temporary/demo license is used
  - Green: license OK
- Dat LED: shows the status of the Data store manager
  - Off: no data stores configured
  - Red: at least one data store not recording
  - Orange: all data stores waiting for trigger
  - Green: all data stores recording or waiting
- PDA1 LED: the first of two LEDs that can be configured in ibaPDA using digital signals (see also 5.2)
- PDA2 LED: the second of two LEDs that can be configured in ibaPDA using digital signals (see also 5.2)
- X20: currently unused SFP port
- X22: RJ45 1 Gigabit Ethernet connector
- X24: USB 3.0 connector

- X25: USB 2.0 connector
- X26: DisplayPort for connecting an external monitor
- X30: backup battery to preserve time and date information when no power is available

5.2 Configuration in ibaPDA

Where other ibaPADU-S modules need to be configured on an external computer running ibaPDA server, here only an ibaPDA client is required to configure the ibaDAQ-S hosting its own ibaPDA server. This client can either be running on a separate computer (provided that the ibaDAQ-S is connected through a network) or on the ibaDAQ-S itself (when a monitor, keyboard and mouse are connected).



Interface settings

Interrupt mode : Master mode internal ☑ In use □ Enable watchdog (DO00)

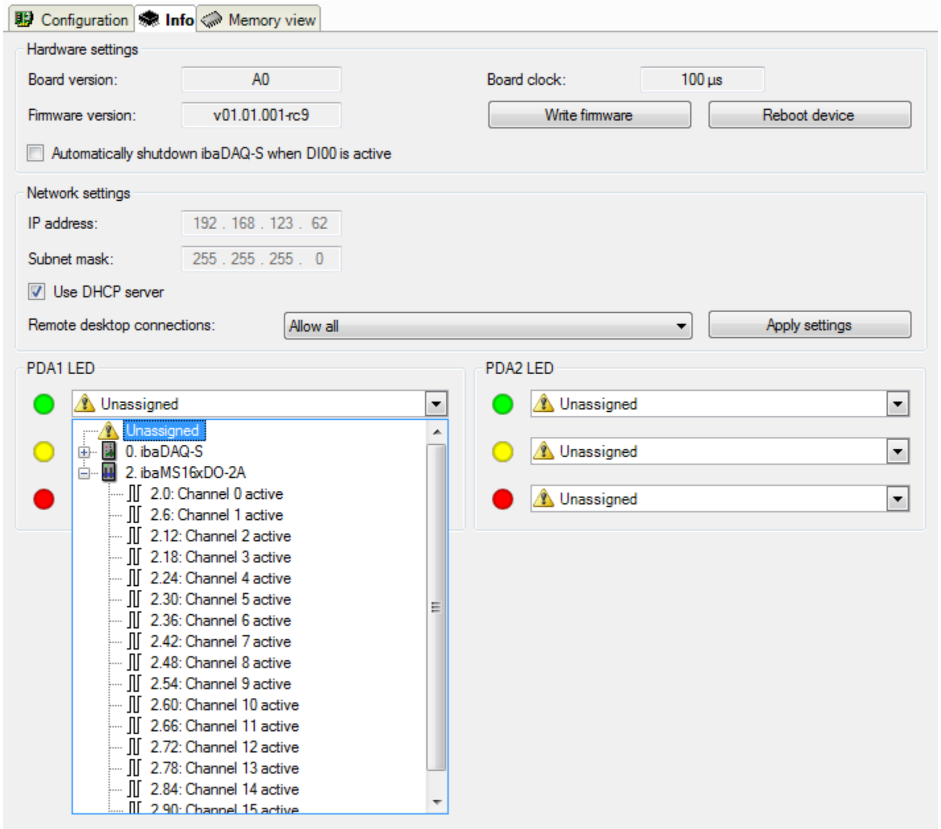
PCI Info

Slot Number:	0	IO Address:	0x00000000	IO Length:	0x00000000
Bus Number:	1	Mem. Address:	0x90000000	Mem. Length:	0x01000000
Vendor:	iba AG			Device Id:	0xDAC5

When opening the I/O Manager in the ibaPDA client, the ibaDAQ-S interface should appear. The ibaDAQ-S interface contains two nodes: the Fiber optic link and the Backplane bus. The Fiber optic link represents an integrated ibaFOB-io-D board in the ibaDAQ-S hardware and is also configured as such. The Backplane bus node always contains the ibaDAQ-S module itself, apart from other modules connected to the bus. These modules can either be added manually, or by using the Autodetect feature when right-clicking on the Backplane bus node.

The Configuration tab of the ibaDAQ-S interface node is similar to that of an ibaFOB-D board with a few exceptions:

- Since no extra iba PCI boards can be added to the ibaDAQ-S, the Interrupt mode cannot be set to Slave mode: the ibaDAQ-S is always in Master mode (internal or external)
- The Enable watchdog (DO00) checkbox: the ibaDAQ-S also implements the watchdog functionality of ibaFOB-D boards. However, when this checkbox is activated, the first digital output on the ibaDAQ-S module (DO00) will be asserted in case of a watchdog alarm
- The schematic representation of the ibaDAQ-S: as is the case for ibaFOB-D boards, this drawing shows the status of the LEDs on the ibaDAQ-S module as well as the position of the rotary switch (S1). The state of the LEDs of the RJ45 connector are not shown.



Hardware settings

Board version: A0

Board clock: 100 µs

Firmware version: v01.01.001-rc9

□ Automatically shutdown ibaDAQ-S when DI00 is active

Network settings

IP address: 192 . 168 . 123 . 62

Subnet mask: 255 . 255 . 255 . 0

☑ Use DHCP server

Remote desktop connections: Allow all

PDA1 LED

● Unassigned

- Unassigned
- 0. ibaDAQ-S
- 2. ibaMS16xDO-2A
  - 2.0: Channel 0 active
  - 2.6: Channel 1 active
  - 2.12: Channel 2 active
  - 2.18: Channel 3 active
  - 2.24: Channel 4 active
  - 2.30: Channel 5 active
  - 2.36: Channel 6 active
  - 2.42: Channel 7 active
  - 2.48: Channel 8 active
  - 2.54: Channel 9 active
  - 2.60: Channel 10 active
  - 2.66: Channel 11 active
  - 2.72: Channel 12 active
  - 2.78: Channel 13 active
  - 2.84: Channel 14 active
  - 2.90: Channel 15 active

PDA2 LED

● Unassigned

● Unassigned

● Unassigned

The Info tab shows more specific ibaDAQ-S settings. The hardware board version is displayed as well as the current firmware version. The ibaDAQ-S firmware can be updated here by clicking the Write firmware button and selecting the appropriate iba-file in the dialog that will appear. Note that after a firmware update, the DAQ-S should be powered off and on manually so the FPGA can completely reset.

When the checkbox Automatically shutdown ibaDAQ-S when DI00 is active is enabled, a high input signal on DI00 will cause the ibaDAQ-S to automatically shutdown. This can be useful when, for example, a UPS asserts an alarm signal when a power failure has occurred. The ibaDAQ-S can then perform a graceful shutdown.

The Network settings group allows the user to manually configure the IP address of the network adapter connected to X22. The IP address can only be configured manually in the ibaPDA client when rotary switch S1 is set to 0.

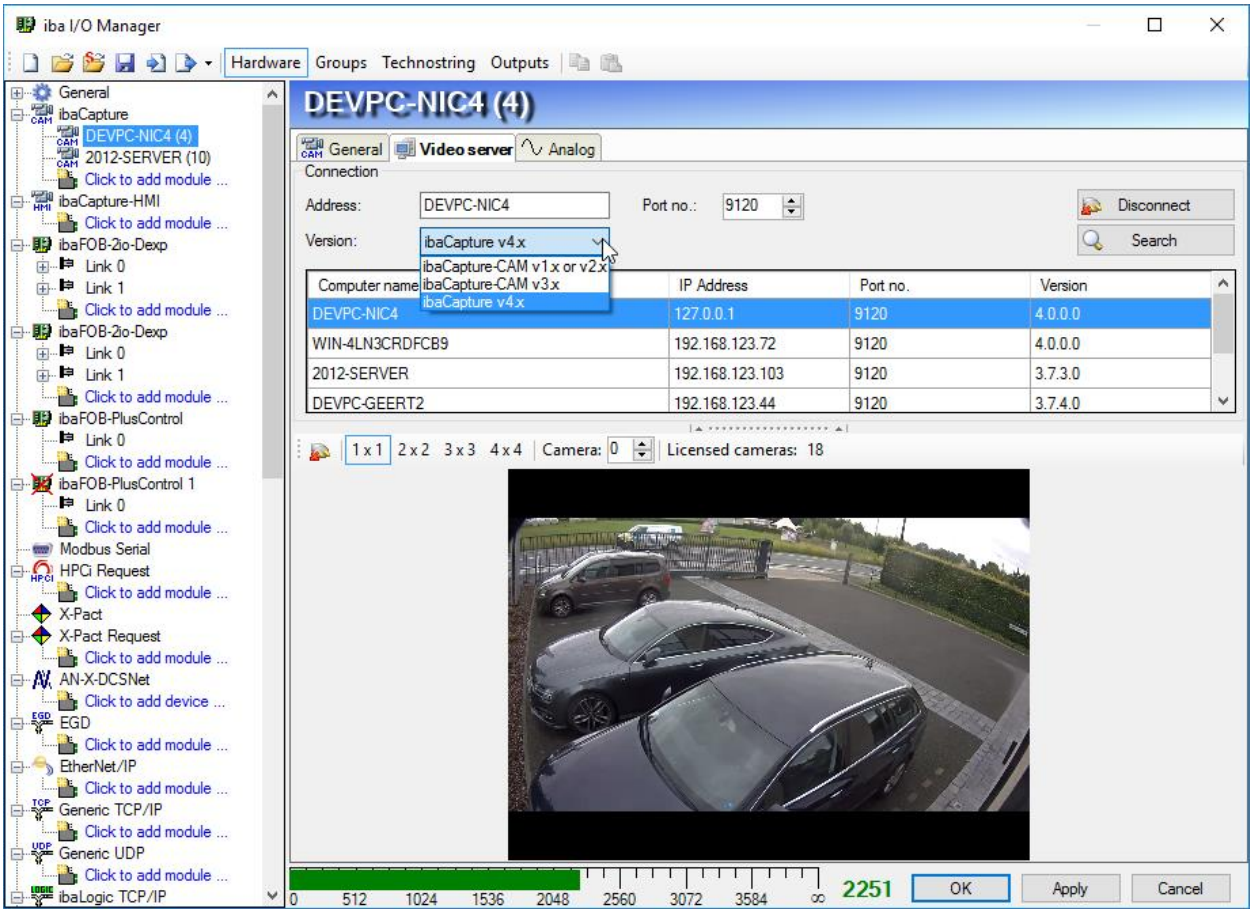
Apart from that, the Remote desktop settings can also be configured here.

Settings are only applied after the user manually clicks the Apply settings button.

Finally, LEDs PDA1 and PDA2 can be configured here. Each LED can be green, orange, red or off with each color controlled by a digital signal. The red color has priority over orange, which has priority over green. In case the three signals for an LED are not asserted, inactive or unassigned, the LED will be off.

6 ibaCapture v4 support

IbaPDA supports all versions of ibaCapture(-CAM). On the video server connection tab you can manually select the version. But for your convenience it will be automatically set when you connect to the video server.



Address: DEVPC-NIC4

Port no.: 9120

Version: ibaCapture v4.x

Computer name	IP Address	Port no.	Version
DEVPC-NIC4	127.0.0.1	9120	4.0.0.0
WIN-4LN3CRDFCB9	192.168.123.72	9120	4.0.0.0
2012-SERVER	192.168.123.103	9120	3.7.3.0
DEVPC-GEERT2	192.168.123.44	9120	3.7.4.0

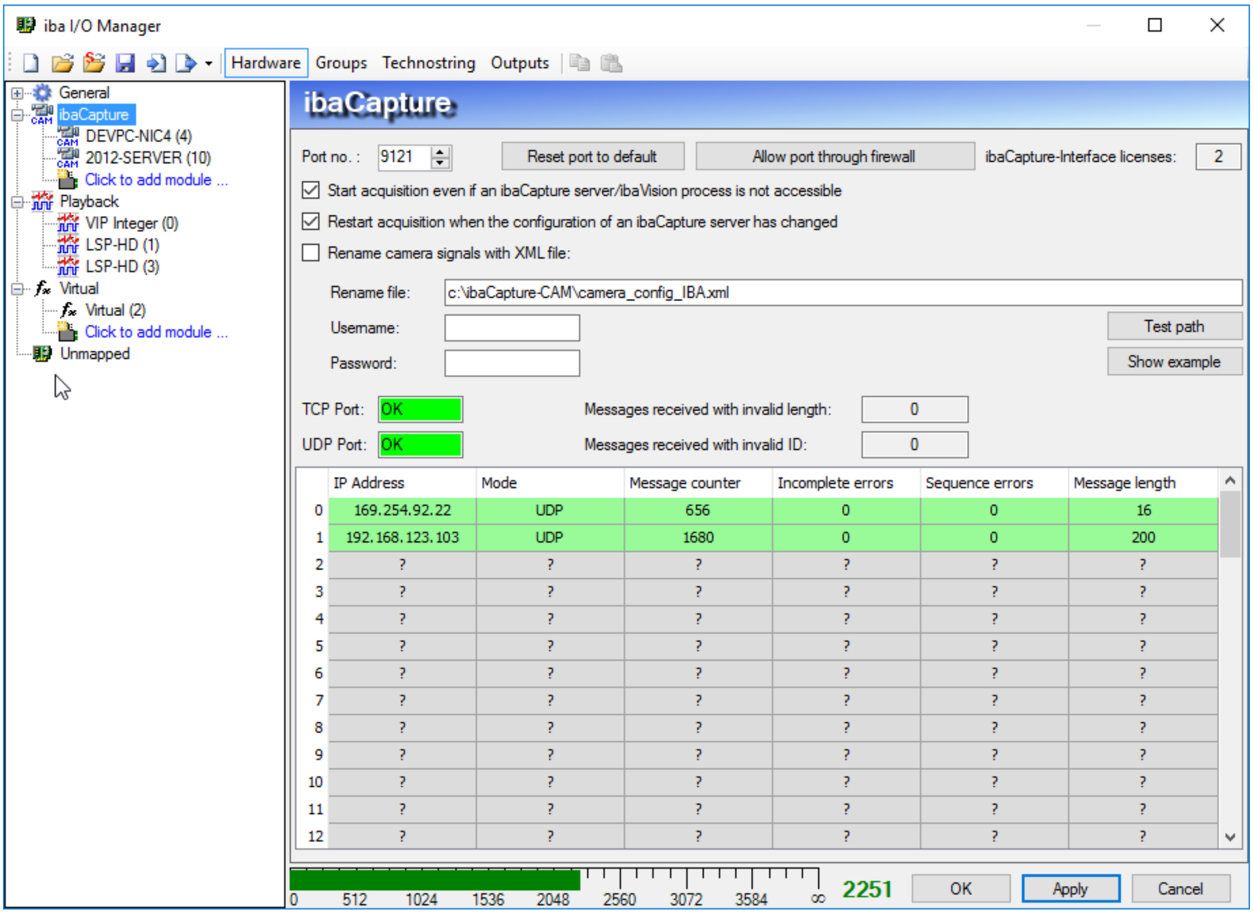
Camera: 0

Licensed cameras: 18

There are some differences between ibaCapture-CAM v3 and ibaCapture v4 that have an influence on ibaPDA.

First of all the maximum number of cameras has been increased from 16 to 64. In ibaPDA you can configure the number of cameras on the general tab of the ibaCapture module. The minimum value is 16. When you connect to the video server in the I/O manager then ibaPDA will automatically set the number of cameras to the correct value.

Secondly the license for the connection between ibaPDA and ibaCapture (*ibaCapture-Interface-PDA* license) is now checked by the ibaCapture v4 server instead of the ibaPDA server. IbaPDA server still checks the license for ibaCapture-CAM v1, v2 and v3 servers. This also means that the ibaCapture interface is always available in the I/O manager even when there is no ibaCapture-Interface-PDA license on the ibaPDA dongle.



Port no. : 9121

ibaCapture-Interface licenses: 2

☑ Start acquisition even if an ibaCapture server/ibaVision process is not accessible

☑ Restart acquisition when the configuration of an ibaCapture server has changed

☐ Rename camera signals with XML file:

Rename file: c:\ibaCapture-CAM\camera_config_IBA.xml

Username:

Password:

TCP Port: OK

UDP Port: OK

Messages received with invalid length: 0

Messages received with invalid ID: 0

	IP Address	Mode	Message counter	Incomplete errors	Sequence errors	Message length
0	169.254.92.22	UDP	656	0	0	16
1	192.168.123.103	UDP	1680	0	0	200
2	?	?	?	?	?	?
3	?	?	?	?	?	?
4	?	?	?	?	?	?
5	?	?	?	?	?	?
6	?	?	?	?	?	?
7	?	?	?	?	?	?
8	?	?	?	?	?	?
9	?	?	?	?	?	?
10	?	?	?	?	?	?
11	?	?	?	?	?	?
12	?	?	?	?	?	?

Thirdly ibaPDA can now get a notification from the ibaCapture v4 server that the camera configuration has changed. If ibaPDA detects a change in camera name or a change in index for one of its active cameras then the acquisition can be automatically restarted. You have to enable the option *“Restart acquisition when the configuration of an ibaCapture server has changed”* on the ibaCapture interface.

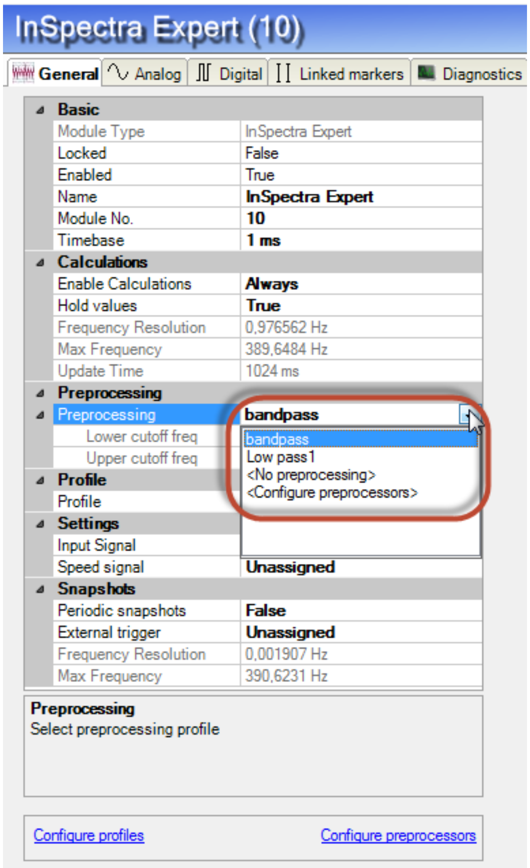
Finally the size of the synchronization messages has changed. For ibaCapture-CAM v3 servers the size was fixed 200 bytes. For ibaCapture v4 servers the size is 8 + 8 * the number of cameras bytes.

7 InSpectra Expert preprocessing

Since ibaPDA 6.35.5, the InSpectra Expert module allows preprocessing the input signal with a custom FIR or IIR filter. The following filter methods are available:

- Low pass
- High pass
- Band pass
- Band stop

You can select and manage the filters in the general TAB of the module:

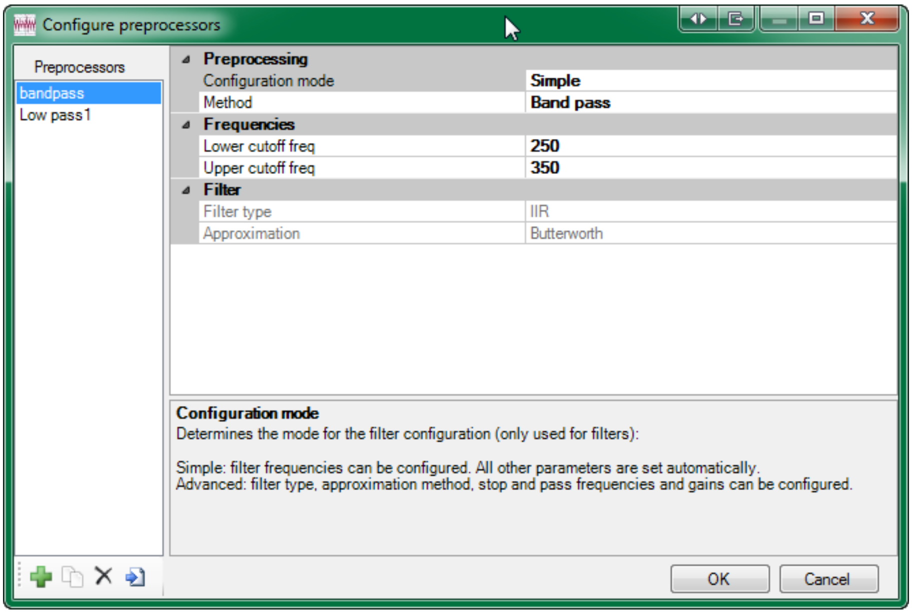


InSpectra Expert (10)

Basic	
Module Type	InSpectra Expert
Locked	False
Enabled	True
Name	**InSpectra Expert**
Module No.	**10**
Timebase	**1 ms**
Calculations	
Enable Calculations	**Always**
Hold values	**True**
Frequency Resolution	0,976562 Hz
Max Frequency	389,6484 Hz
Update Time	1024 ms
Preprocessing	
Preprocessing	**bandpass**
Lower cutoff freq	
Upper cutoff freq	
Profile	
Profile	
Settings	
Input Signal	
Speed signal	**Unassigned**
Snapshots	
Periodic snapshots	**False**
External trigger	**Unassigned**
Frequency Resolution	0,001907 Hz
Max Frequency	390,6231 Hz

Preprocessing
Select preprocessing profile

By clicking <Configure preprocessors> in the drop down, you can access the preprocessing manager. In the left part you can create, delete or rename filters which you can reuse in other InSpectra Expert modules.



Preprocessing	
Configuration mode	**Simple**
Method	**Band pass**
Frequencies	
Lower cutoff freq	**250**
Upper cutoff freq	**350**
Filter	
Filter type	IIR
Approximation	Butterworth

Configuration mode
Determines the mode for the filter configuration (only used for filters):

Simple: filter frequencies can be configured. All other parameters are set automatically.
Advanced: filter type, approximation method, stop and pass frequencies and gains can be configured.

You can create your own filters or import a filter designed by ibaAnalyzer.

- Importing a filter from ibaAnalyzer:

  ibaAnalyzer has a filter design tool. You can access it via “Setup” -> “Digital filter design…”. You can export the generated filter to a .fil file. In the ibaPDA preprocessing manager, you can import the filter using the right most button at the bottom.

- Creating your own filter:

  Two filter configuration modes are supported:

  - Simple:
    - Low/high pass: one cutoff frequency needs to be specified
    - Band pass/stop: two cutoff frequencies need to be specified
  - Advanced:
    - Low/high pass: two (frequency, amplitude) spectrum points need to be specified
    - Band pass/stop: four (frequency, amplitude) spectrum points need to be specified

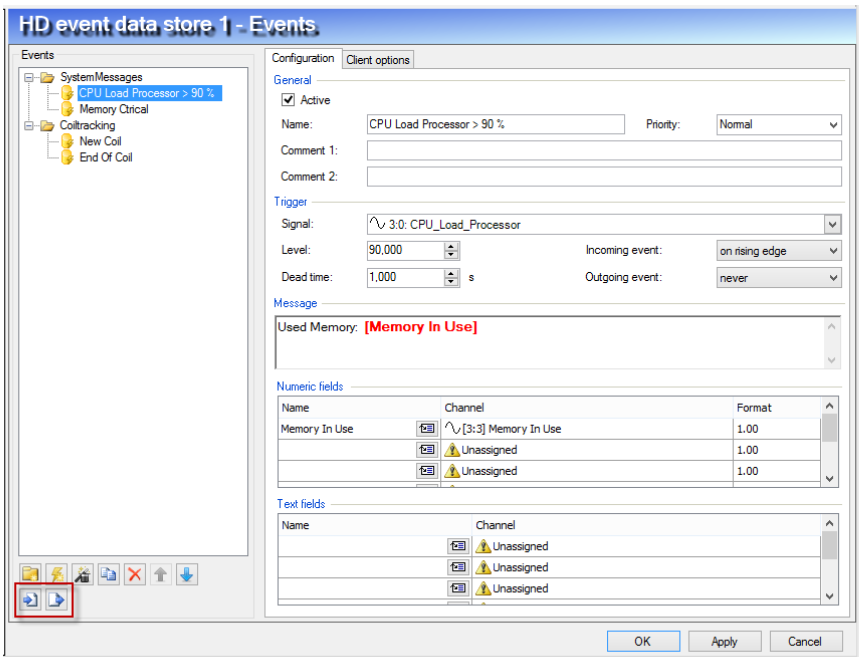
In case of simple mode, the cutoff frequencies are automatically converted to spectrum points. Based on the given spectrum points and the timebase of the input signal, the actual filter coefficients are calculated at run-time when starting the acquisition. The resulting filter coefficients are stored temporarily in your installation directory: “iba\ibaPDA\Server\temp\filters”. They are stored as Matlab files so you can analyse them with Matlab easily. The number in the filename corresponds with the InSpectra module number.

In case of an IIR filter, one can choose the approximation type (Butterworth, Chebyshev, Elliptic, Inv-Chebyshev). In case of a FIR filter, a window can be specified (default is rectangular).

8 ibaHD event store improvements

8.1 Export and import of ibaHD Server event data store configuration

The export of the actual event data store configuration will generate a .txt file with tab separation. The exported .txt file could be opened with any text editor or imported to MS Excel for editing purposes.



☑ Active

Name: CPU Load Processor > 90 %

Priority: Normal

Comment 1:

Comment 2:

Signal: 3:0: CPU_Load_Processor

Level: 90,000

Incoming event: on rising edge

Dead time: 1,000 s

Outgoing event: never

Used Memory: **[Memory In Use]**

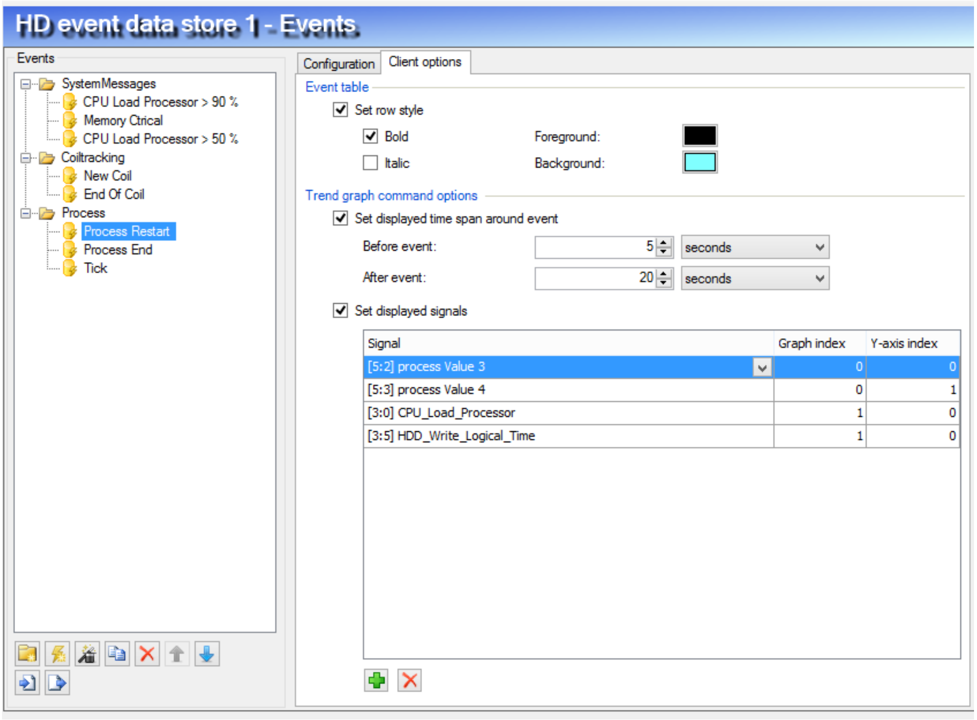
Name	Channel	Format
Memory In Use	[3:3] Memory In Use	1.00
	Unassigned	1.00
	Unassigned	1.00

Name	Channel
	Unassigned
	Unassigned
	Unassigned

The import of an event data store configuration from a .txt file will replace the existing one.

8.2 Client options for ibaHD Events

Client options are optional settings for each configured ibaHD Event. The options affect the trendview in case the user selects or clicks a row at the event table. The options can be set for each event individually. The settings are part of the event data store configuration and are present in all connected ibaPDA Clients.



HD event data store 1 - Events

Events

- SystemMessages
  - CPU Load Processor > 90 %
  - Memory Ctrical
  - CPU Load Processor > 50 %
- Coiltracking
  - New Coil
  - End Of Coil
- Process
  - Process Restart
  - Process End
  - Tick

Configuration | Client options

Event table

☑ Set row style

☑ Bold Foreground: ■

☐ Italic Background: □

Trend graph command options

☑ Set displayed time span around event

Before event: 5 seconds

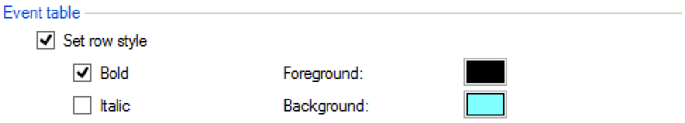
After event: 20 seconds

☑ Set displayed signals

Signal	Graph index	Y-axis index
[5:2] process Value 3	0	0
[5:3] process Value 4	0	1
[3:0] CPU_Load_Processor	1	0
[3:5] HDD_Write_Logical_Time	1	0

8.2.1 Table row style

For each event you can configure how it will look by default in the event table. You can set the font style (bold, italic). You can also configure the foreground and background colors of the event table row.



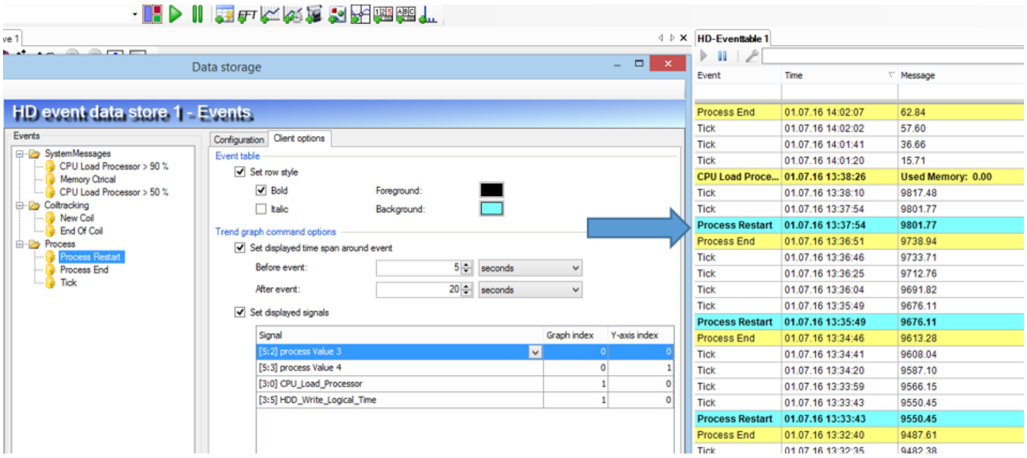
Event table

☑ Set row style

☑ Bold Foreground: ■

☐ Italic Background: □

Styles defined in event table properties will override styles configured in event data store configuration.



Data storage

HD event data store 1 - Events

Events

- SystemMessages
  - CPU Load Processor > 90 %
  - Memory Ctrical
  - CPU Load Processor > 50 %
- Coiltracking
  - New Coil
  - End Of Coil
- Process
  - Process Restart
  - Process End
  - Tick

Configuration | Client options

Event table

☑ Set row style

☑ Bold Foreground: ■

☐ Italic Background: □

Trend graph command options

☑ Set displayed time span around event

Before event: 5 seconds

After event: 20 seconds

☑ Set displayed signals

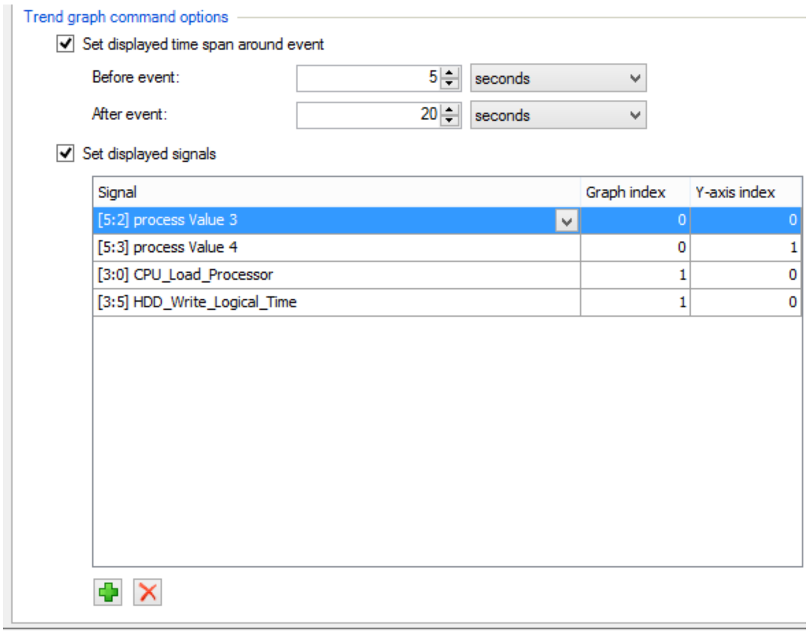
Signal	Graph index	Y-axis index
[5:2] process Value 3	0	0
[5:3] process Value 4	0	1
[3:0] CPU_Load_Processor	1	0
[3:5] HDD_Write_Logical_Time	1	0

HD-Eventtable 1

Event	Time	Message
Process End	01.07.16 14:02:07	62.84
Tick	01.07.16 14:02:02	57.60
Tick	01.07.16 14:01:41	36.66
Tick	01.07.16 14:01:20	15.71
CPU Load Proce...	**01.07.16 13:38:26**	**Used Memory: 0.00**
Tick	01.07.16 13:38:10	9817.48
Tick	01.07.16 13:37:54	9801.77
Process Restart	**01.07.16 13:37:54**	**9801.77**
Process End	01.07.16 13:36:51	9738.94
Tick	01.07.16 13:36:46	9733.71
Tick	01.07.16 13:36:25	9712.76
Tick	01.07.16 13:36:04	9691.82
Tick	01.07.16 13:35:49	9676.11
Process Restart	**01.07.16 13:35:49**	**9676.11**
Process End	01.07.16 13:34:46	9613.28
Tick	01.07.16 13:34:41	9608.04
Tick	01.07.16 13:34:20	9587.10
Tick	01.07.16 13:33:59	9566.15
Tick	01.07.16 13:33:43	9550.45
Process Restart	**01.07.16 13:33:43**	**9550.45**
Process End	01.07.16 13:32:40	9487.61
Tick	01.07.16 13:32:35	9482.38

8.2.2 Trend graph command options

Clicking on events in the event table can command trend views to jump to the timestamp of the event and pause the trend graph. You can also configure a list of ibaHD signals that are linked to the event.



Trend graph command options

☑ Set displayed time span around event

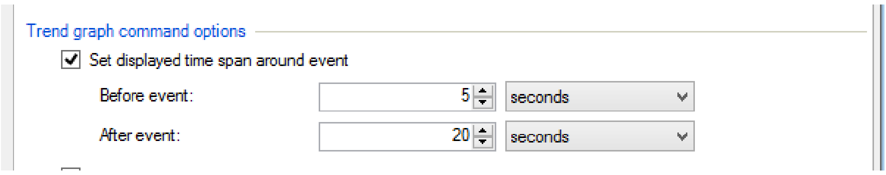
Before event: 5 seconds

After event: 20 seconds

☑ Set displayed signals

Signal	Graph index	Y-axis index
[5:2] process Value 3	0	0
[5:3] process Value 4	0	1
[3:0] CPU_Load_Processor	1	0
[3:5] HDD_Write_Logical_Time	1	0

8.2.2.1 Set displayed time span around event



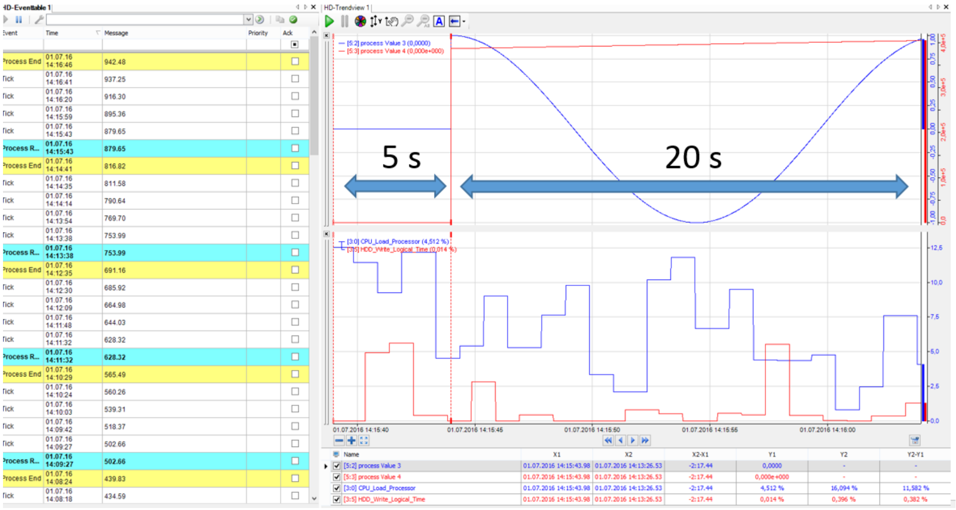
Trend graph command options

☑ Set displayed time span around event

Before event: 5 seconds

After event: 20 seconds

The commanded trendview will open with selected time span around the selected event.

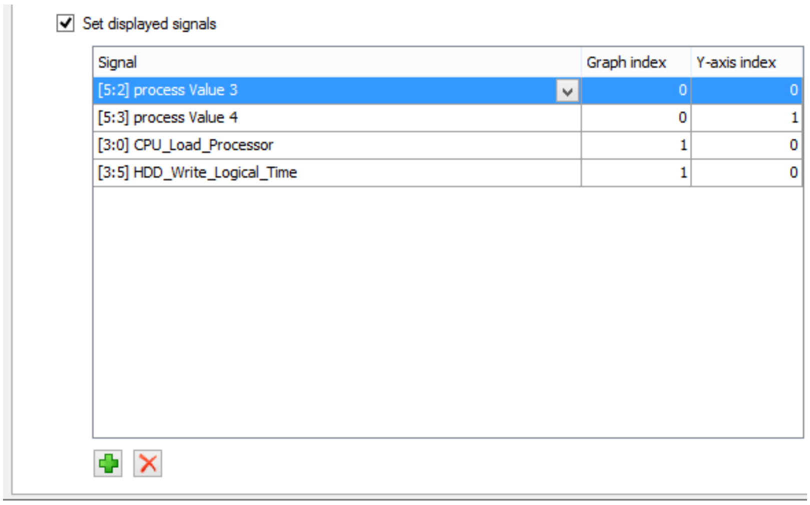


Event	Time	Message	Priority	Ack
				▣
Process End	01.07.16 14:16:46	942.48		☐
Tick	01.07.16 14:16:41	937.25		☐
Tick	01.07.16 14:16:20	916.30		☐
Tick	01.07.16 14:15:59	895.36		☐
Tick	01.07.16 14:15:43	879.65		☐
Process R...	01.07.16 14:15:43	879.65		☐
Process End	01.07.16 14:14:41	816.82		☐
Tick	01.07.16 14:14:35	811.58		☐
Tick	01.07.16 14:14:14	790.64		☐
Tick	01.07.16 14:13:54	769.70		☐
Tick	01.07.16 14:13:38	753.99		☐
Process R...	01.07.16 14:13:38	753.99		☐
Process End	01.07.16 14:12:35	691.16		☐
Tick	01.07.16 14:12:30	685.92		☐
Tick	01.07.16 14:12:09	664.98		☐
Tick	01.07.16 14:11:48	644.03		☐
Tick	01.07.16 14:11:32	628.32		☐
Process R...	01.07.16 14:11:32	628.32		☐
Process End	01.07.16 14:10:29	565.49		☐
Tick	01.07.16 14:10:24	560.26		☐
Tick	01.07.16 14:10:03	539.31		☐
Tick	01.07.16 14:09:42	518.37		☐
Tick	01.07.16 14:09:27	502.66		☐
Process R...	01.07.16 14:09:27	502.66		☐
Process End	01.07.16 14:08:24	439.83		☐
Tick	01.07.16 14:08:18	434.59		☐

Name	X1	X2	X2-X1	Y1	Y2	Y2-Y1
☑ [5:2] process Value 3	01.07.2016 14:15:43.98	01.07.2016 14:13:26.53	-2:17.44	0,0000	-	-
☑ [5:3] process Value 4	01.07.2016 14:15:43.98	01.07.2016 14:13:26.53	-2:17.44	0,000e+000	-	-
☑ [3:0] CPU_Load_Processor	01.07.2016 14:15:43.98	01.07.2016 14:13:26.53	-2:17.44	4,512 %	16,094 %	11,582 %
☑ [3:5] HDD_Write_Logical_Time	01.07.2016 14:15:43.98	01.07.2016 14:13:26.53	-2:17.44	0,014 %	0,396 %	0,382 %

When there is no time span defined in the event configuration then the trend graph will just put the event timestamp in the middle of the X-axis and it won’t change the size of the X-axis.

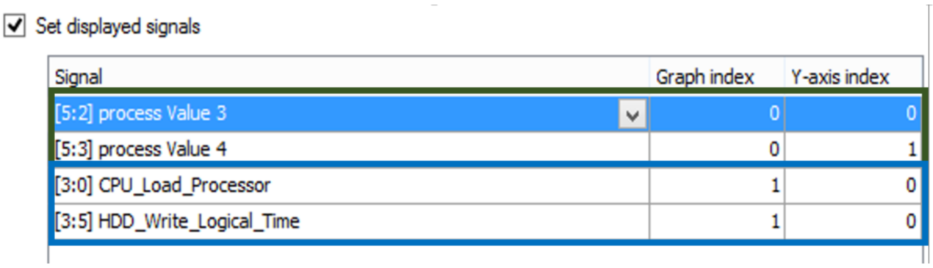
8.2.2.2 Set displayed signals



☑ Set displayed signals

Signal	Graph index	Y-axis index
[5:2] process Value 3	0	0
[5:3] process Value 4	0	1
[3:0] CPU_Load_Processor	1	0
[3:5] HDD_Write_Logical_Time	1	0

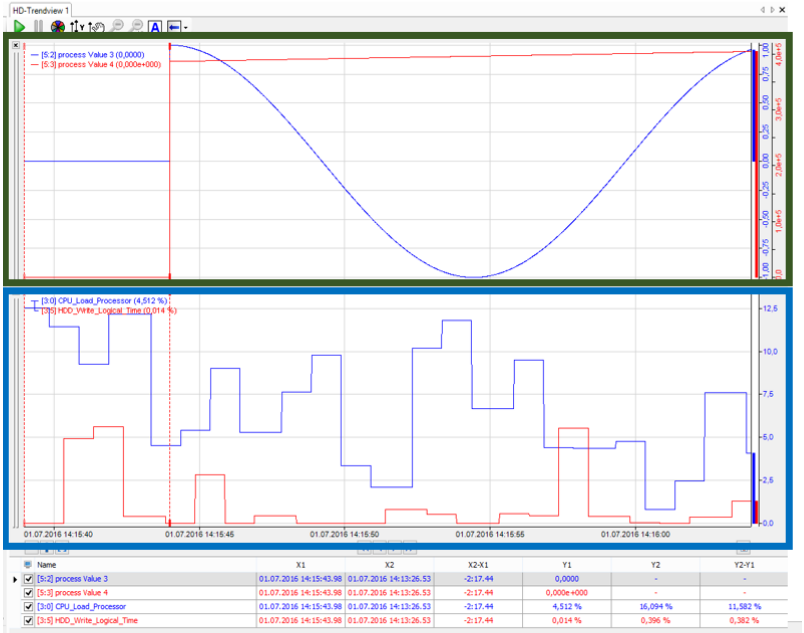
The commanded trendview will open with the configured signals for the display. Events can be combined with related signals. E.g. Events related to a machine state are going to display signals of the machine in a trendview.



☑ Set displayed signals

Signal	Graph index	Y-axis index
[5:2] process Value 3	0	0
[5:3] process Value 4	0	1
[3:0] CPU_Load_Processor	1	0
[3:5] HDD_Write_Logical_Time	1	0

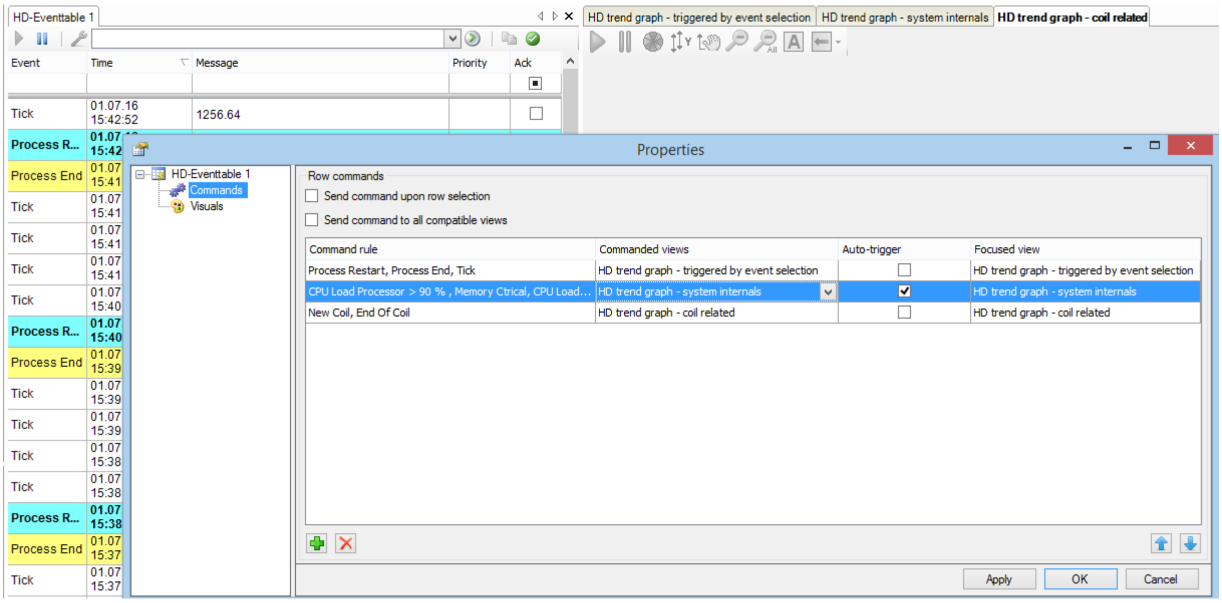
With the example configuration above the commanded trendview is going to display 2 trend graphs. Each graph will show 2 signals from the connected HD Server. Each signal in the first graph will have an own Y-axis and in the second graph they will be using the same Y-axis.



Name	X1	X2	X2-X1	Y1	Y2	Y2-Y1
☑ [5:2] process Value 3	01.07.2016 14:15:43.98	01.07.2016 14:13:26.53	-2:17.44	0,0000	-	-
☑ [5:3] process Value 4	01.07.2016 14:15:43.98	01.07.2016 14:13:26.53	-2:17.44	0,000e+000	-	-
☑ [3:0] CPU_Load_Processor	01.07.2016 14:15:43.98	01.07.2016 14:13:26.53	-2:17.44	4,512 %	16,094 %	11,582 %
☑ [3:5] HDD_Write_Logical_Time	01.07.2016 14:15:43.98	01.07.2016 14:13:26.53	-2:17.44	0,014 %	0,396 %	0,382 %

Use ➕ ✕ to add or remove signals from the list of displayed signals.

8.3 Improved command configuration in ibaHD event table



Properties

HD-Eventtable 1
- Commands
- Visuals

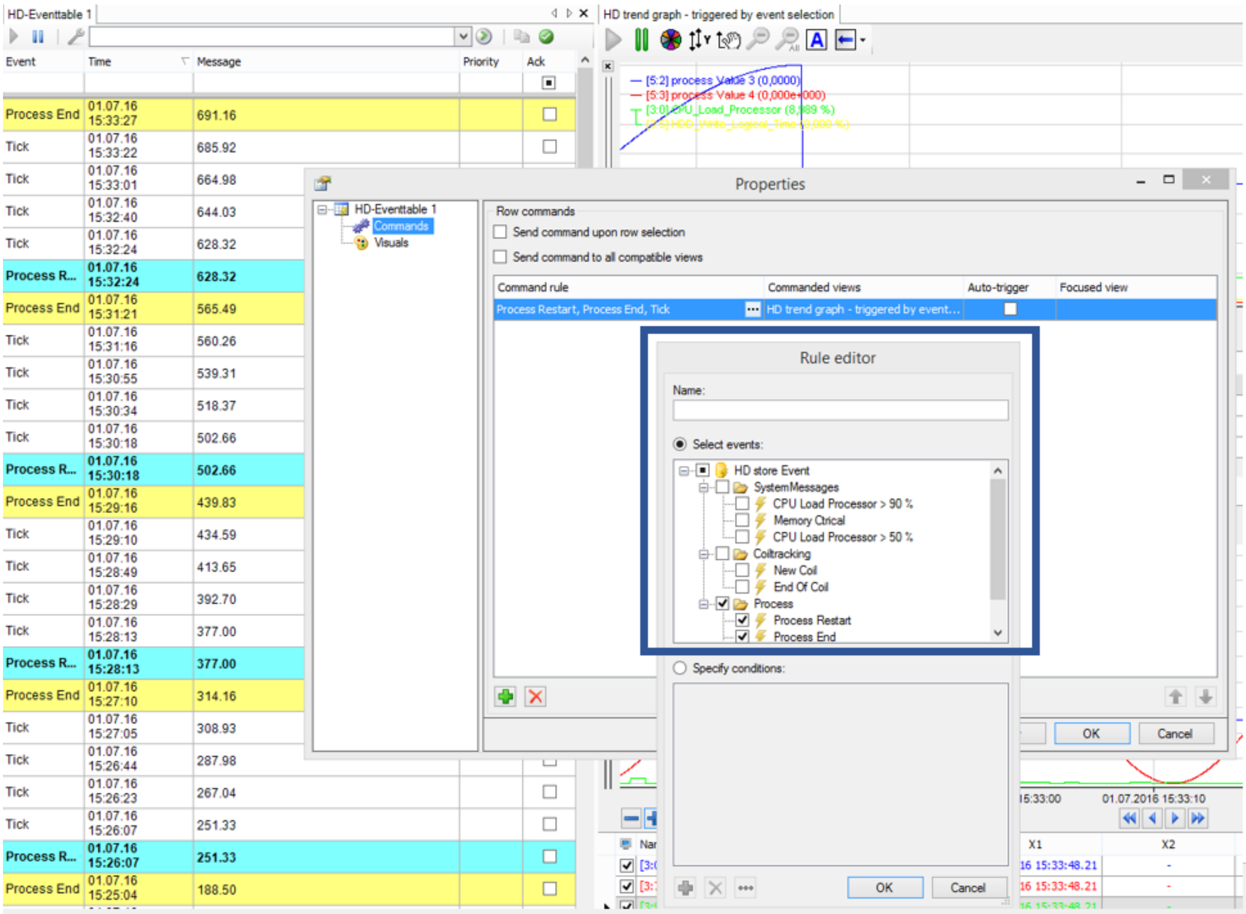
Row commands

□ Send command upon row selection

□ Send command to all compatible views

Command rule	Commanded views	Auto-trigger	Focused view
Process Restart, Process End, Tick	HD trend graph - triggered by event selection	□	HD trend graph - triggered by event selection
CPU Load Processor > 90 % , Memory Ctrical, CPU Load…	HD trend graph - system internals	☑	HD trend graph - system internals
New Coil, End Of Coil	HD trend graph - coil related	□	HD trend graph - coil related

The commands are presented as a table in the event table properties. The rule editor of the ibaHD event table has been modified. You can now easily select one or more events for a command via the event signal tree.



Properties

Row commands

□ Send command upon row selection

□ Send command to all compatible views

Command rule	Commanded views	Auto-trigger	Focused view
Process Restart, Process End, Tick	HD trend graph - triggered by event…	□	

Rule editor

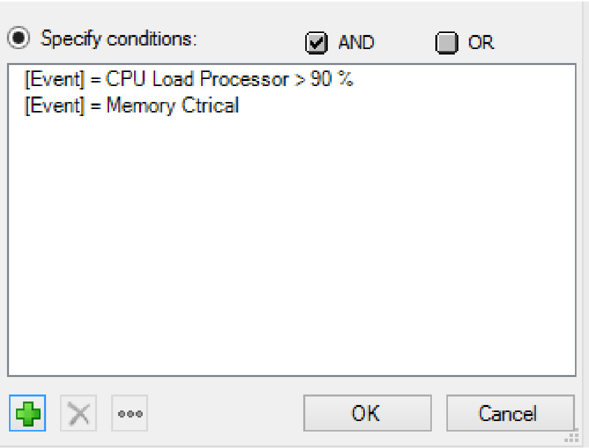
Name:

● Select events:

- ■ HD store Event
  - □ SystemMessages
    - □ CPU Load Processor > 90 %
    - □ Memory Ctrical
    - □ CPU Load Processor > 50 %
  - □ Coiltracking
    - □ New Coil
    - □ End Of Coil
  - ☑ Process
    - ☑ Process Restart
    - ☑ Process End

○ Specify conditions:

Rules can now be combined with *AND* and *OR* conditions.

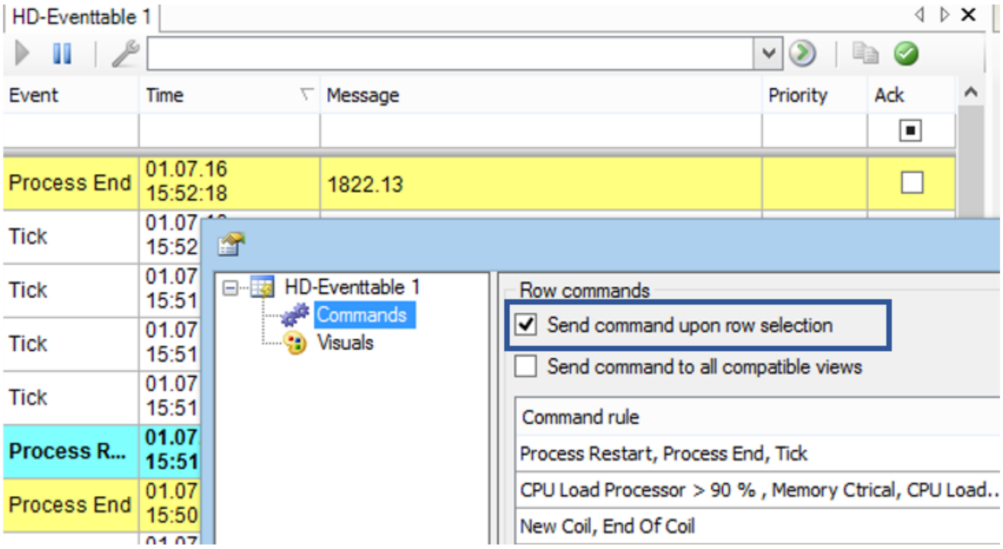


● Specify conditions: ☑ AND □ OR

[Event] = CPU Load Processor > 90 %
[Event] = Memory Ctrical

8.4 Send command upon row selection in ibaHD event table

Normally the event table commands are executed when you doubleclick a row in the event table. When you enable the *“Send command upon row selection”* option then the command will be executed as soons as you select the row, either by clicking it or by navigating to it via the arrow keys on the keyboard.



Event	Time	Message	Priority	Ack
				▣
Process End	01.07.16 15:52:18	1822.13		□
Tick	01.07 15:52			
Tick	01.07 15:51			
Tick	01.07 15:51			
Tick	01.07 15:51			
Process R...	01.07 15:51			
Process End	01.07 15:50			

Row commands

☑ Send command upon row selection

□ Send command to all compatible views

Command rule
Process Restart, Process End, Tick
CPU Load Processor > 90 % , Memory Ctrical, CPU Load..
New Coil, End Of Coil