ibaPDA-Interface-ABB-Xplorer

PLC-Xplorer data interface for ABB AC800 controllers

Manual

Issue 1.3

Measurement Systems for Industry and Energy

www.iba-ag.com

Manufacturer

iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

Contacts

Main office	+49 911 97282-0
Support	+49 911 97282-14
Engineering	+49 911 97282-13
E-mail	iba@iba-ag.com
Web	www.iba-ag.com

The content of this publication has been checked for compliance with the described hardware and software. Nevertheless, discrepancies cannot be ruled out, and we do not provide guarantee for complete conformity. However, the information furnished in this publication is updated regularly. Required corrections are contained in the following regulations or can be downloaded on the Internet.

The current version is available for download on our web site www.iba-ag.com.

Version	Date	Revision	Author	Version SW
1.3	08-2025	Editorial update ibaPDA v8.11	nm	8.0.3

Contents

1 About this documentation

This documentation describes the function and application of the software interface *ibaPDA-Interface-ABB-Xplorer*.

Other documentation

This documentation is a supplement to the *ibaPDA* manual. Information about all the other characteristics and functions of *ibaPDA* can be found in the *ibaPDA* manual or in the online help.

1.1 Target group and previous knowledge

This documentation is aimed at qualified professionals who are familiar with handling electrical and electronic modules as well as communication and measurement technology. A person is regarded as professional if he/she is capable of assessing safety and recognizing possible consequences and risks on the basis of his/her specialist training, knowledge and experience and knowledge of the standard regulations.

This documentation in particular addresses persons who are concerned with the configuration, test, commissioning or maintenance of Programmable Logic Controllers of the supported products. For the handling *ibaPDA-Interface-ABB-Xplorer* the following basic knowledge is required and/or useful:

- Windows operating system
- Basic knowledge of *ibaPDA*
- Knowledge of configuration and operation of the relevant control system

1.2 Notations

In this manual, the following notations are used:

Action	**Notation**
Menu command	Menu *Logic diagram*
Calling the menu command	*Step 1 – Step 2 – Step 3 – Step x* Example: Select the menu *Logic diagram – Add – New function block*.
Keys	<Key name> Example: <Alt>; <F1>
Press the keys simultaneously	<Key name> + <Key name> Example: <Alt> + <Ctrl>
Buttons	<Key name> Example: <OK>; <Cancel>
Filenames, paths	`Filename`, `Path` Example: `Test.docx`

1.3 Used symbols

If safety instructions or other notes are used in this manual, they mean:

Danger!

The non-observance of this safety information may result in an imminent risk of death or severe injury:

- Observe the specified measures.

Warning!

The non-observance of this safety information may result in a potential risk of death or severe injury!

- Observe the specified measures.

Caution!

The non-observance of this safety information may result in a potential risk of injury or material damage!

- Observe the specified measures

Note

A note specifies special requirements or actions to be observed.

Tip

Tip or example as a helpful note or insider tip to make the work a little bit easier.

Other documentation

Reference to additional documentation or further reading.

2 System requirements

The following system requirements are necessary when using the ABB-Xplorer data interface:

- *ibaPDA* v8.0.0 or higher
- Base license for *ibaPDA* + license for *ibaPDA-Interface-PLC-Xplorer* or *ibaPDA-Interface-ABB-Xplorer*
- With more than 16 connections you need additional *one-step-up-Interface-ABB-Xplorer* licenses for each additional 16 connections.

Note

ℹ The *ibaPDA-Interface-PLC-Xplorer* license contains, among others, the license for this interface.

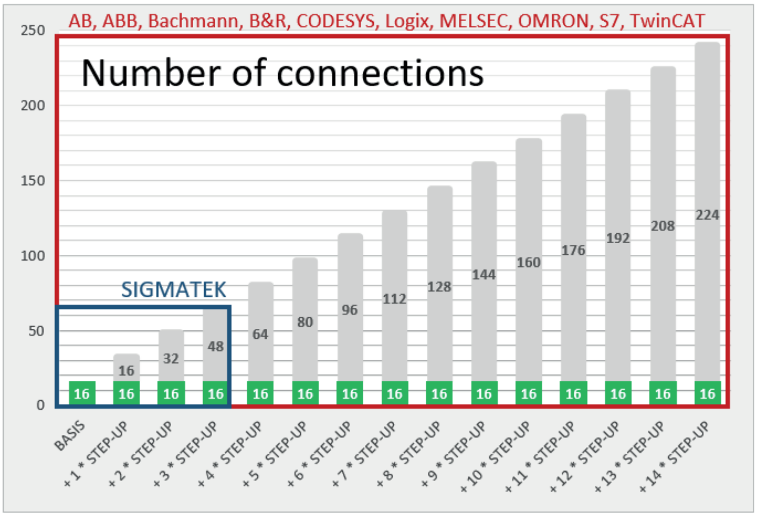
For further requirements for the used computer hardware and the supported operating systems, refer to the *ibaPDA* documentation.

License information

Order no.	Product name	Description
31.001042	ibaPDA-Interface-PLC-Xplorer	Extension license for *ibaPDA* system. All additional Xplorer data interfaces are added. (Complete specifications at www.iba-ag.com)
31.000009	ibaPDA-Interface-ABB-Xplorer	Extension license for an *ibaPDA* system adding the data interface: + ABB-Xplorer (16 connections)
31.100009	one-step-up-Interface-ABB-Xplorer	Extension license for 16 additional ABB-Xplorer connections (max. 14 extension licenses)

Note

To use more than 16 data connections per interface, you can purchase the one-step-up-... extension licenses separately for each interface. Up to 16 further connections to PLCs can be established on each one-step-up-license. Up to 240 connections can be configured and used per data interface with the multiple purchase or multiple release of these licenses (up to 15 in total).
Exception of SIGMATEK: Here, only up to 4 licenses (64 connections) can be activated.



Consider the limitation of the number of signals by the *ibaPDA* base license.

3 PLC-Xplorer data interface for AC800 controllers

The ABB-Xplorer interface is suitable for measurement data acquisition with *ibaPDA* to ABB AC800M and AC800PEC controllers.

The data is cyclically read by *ibaPDA* instead of being sent by the PLC.

In the ABB AC800 controller, no programming work is necessary for establishing a connection between *ibaPDA* and a controller with defined IP address and for sending the respective signals. For transmitting measurement data, no additional software from ABB is necessary.

3.1 System topologies

The connections to the controllers can be established via the computer's standard Ethernet ports.

No further software is necessary for operation.

Note

It is recommended carrying out the TCP/IP communication on a separate network segment to exclude a mutual influence by other network components.

3.2 Configuration and engineering of the AC800 controller

For the configuration of AC800 controllers, you have to define variables that are accessible from *ibaPDA* as MMS access variables in the ABB Compact Control Builder.

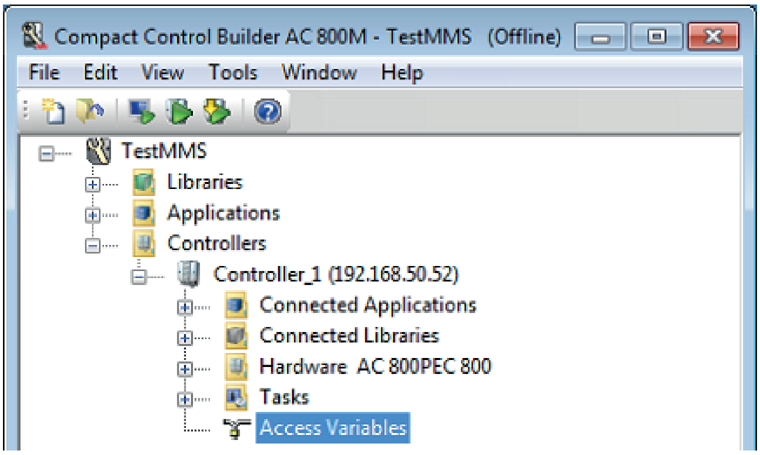
Note

As a standard, the MMS connection to your PLC is enabled as soon as the Ethernet connection is established. If this is not the case, you have to configure the connection independently. You can find more detailed information in the manufacturer’s manual.

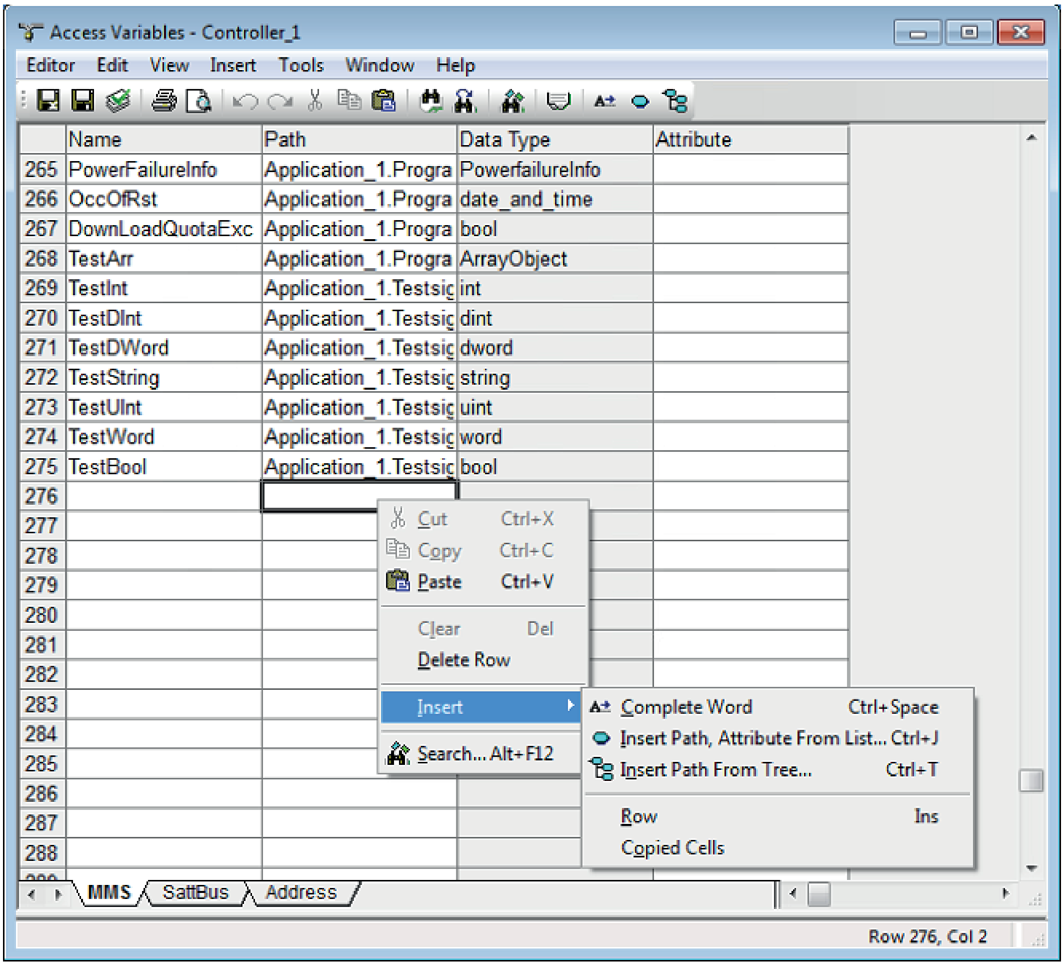
3.2.1 Access variables in the ABB Compact Control Builder

Edit the access variables in the ABB Compact Control Builder according to the following steps.

1. Open the controller in use and select *Access Variables*.



2. Select the *MMS* tab.

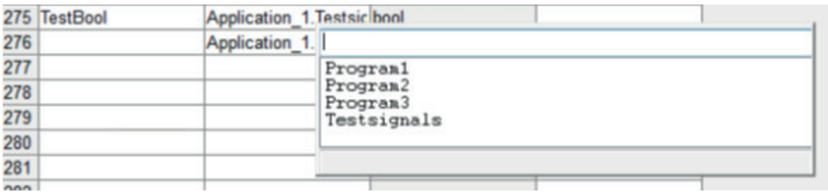

Access Variables - Controller_1
Editor Edit View Insert Tools Window Help
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Clear Del
Delete Row
Insert
Search... Alt+F12
Complete Word Ctrl+Space
Insert Path, Attribute From List... Ctrl+J
Insert Path From Tree... Ctrl+T
Row Ins
Copied Cells
MMS
SattBus
Address

	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				
277				
278				
279				
280				
281				
282				
283				
284				
285				
286				
287				
288				

Row 276, Col 2

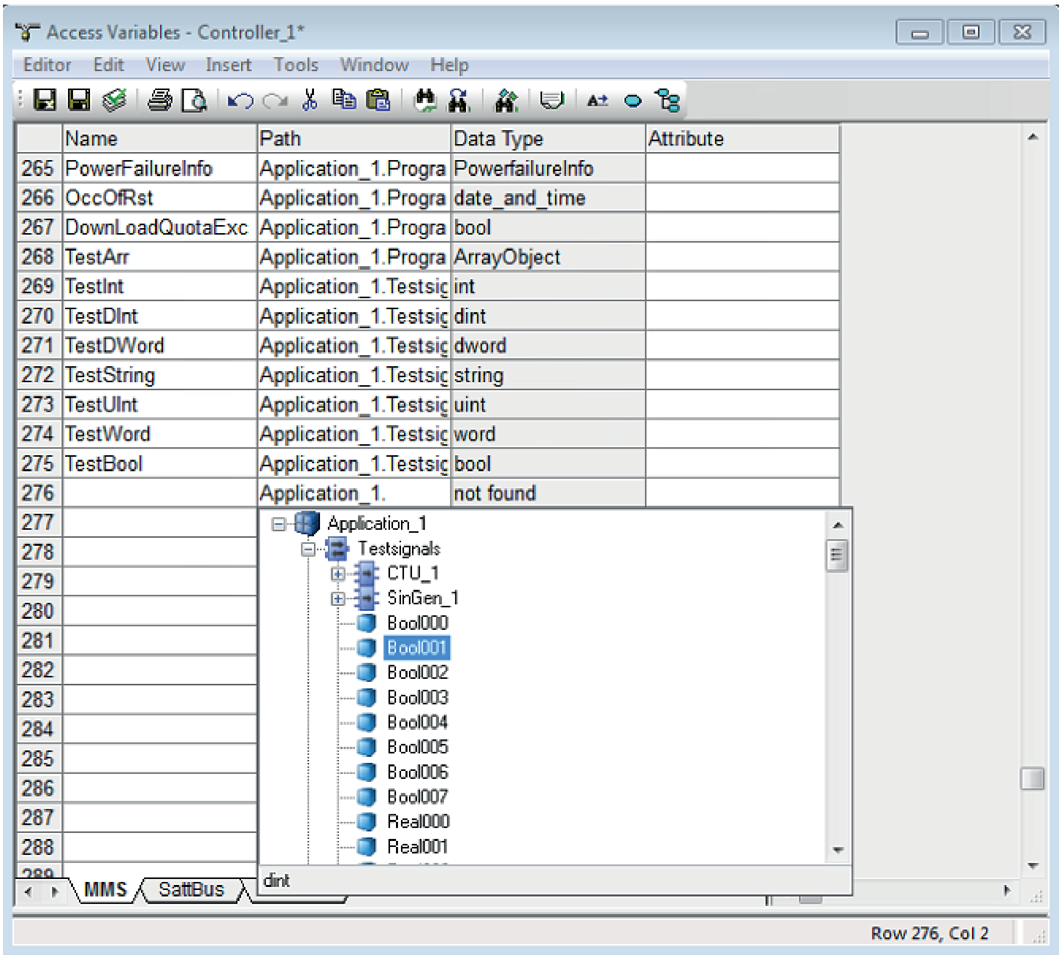
3. Define a name for the access variable and select the corresponding path to the variable within the program.

   To do this, either type the path name and enable the auto-fill function (Complete Word, Insert Path from List) or use the variable browser (Insert Path From Tree).



275	TestBool	Application_1.Testsic bool
276		Application_1.
277		
278		
279		
280		
281		

```
Program1
Program2
Program3
Testsignals
```



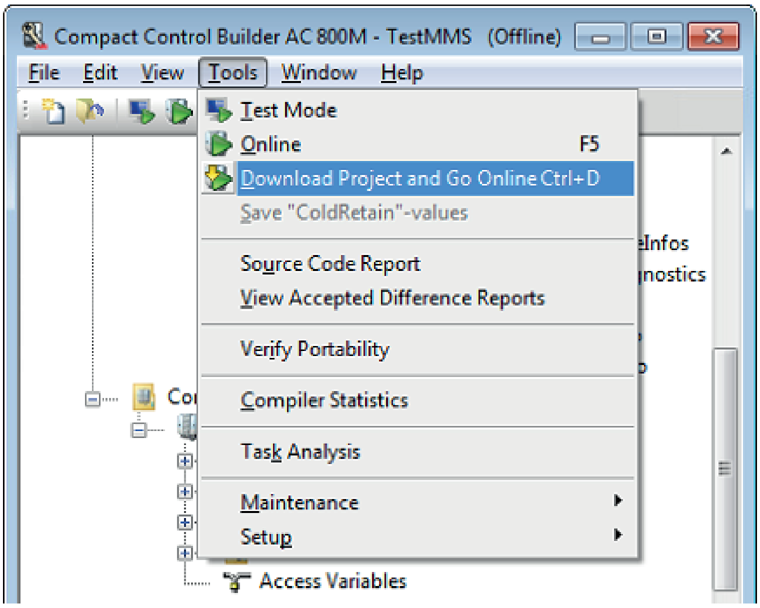
	Name	Path	Data Type	Attribute
265	PowerFailureInfo	Application_1.Progra	PowerfailureInfo	
266	OccOfRst	Application_1.Progra	date_and_time	
267	DownLoadQuotaExc	Application_1.Progra	bool	
268	TestArr	Application_1.Progra	ArrayObject	
269	TestInt	Application_1.Testsig	int	
270	TestDInt	Application_1.Testsig	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	
277				
278				
279				
280				
281				
282				
283				
284				
285				
286				
287				
288				

```
Application_1
  Testsignals
    CTU_1
    SinGen_1
    Bool000
    Bool001
    Bool002
    Bool003
    Bool004
    Bool005
    Bool006
    Bool007
    Real000
    Real001
```

dint

Row 276, Col 2

4. Once all variables are defined, select *Tools – Download Project and Go Online* in the main menu to refresh the program in the PLC.



5. Now use the <Test> button in the *Connection* tab in the I/O Manager of *ibaPDA* to refresh the address book. Select the new variables in the ABB MMS symbol browser and insert them into your *ibaPDA* project.

3.3 Configuration and engineering ibaPDA

The engineering for *ibaPDA* is described in the following. If all system requirements are fulfilled, *ibaPDA* displays the *ABB-Xplorer* interface in the interface tree of the I/O Manager.

3.3.1 General interface settings

If the Xplorer interface is selected in the tree, you can see an overview of diagnostics information on the configured connections between *ibaPDA* and the controllers.

The interface has the following features and configuration options.

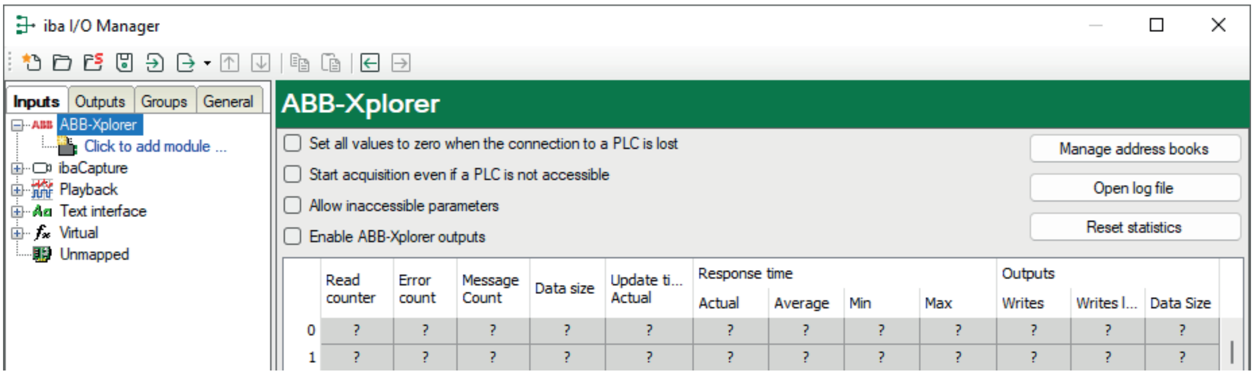


ABB-Xplorer

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

☐ Start acquisition even if a PLC is not accessible

☐ Allow inaccessible parameters

☐ Enable ABB-Xplorer outputs

	Read counter	Error count	Message Count	Data size	Update ti... Actual	Response time				Outputs		
						Actual	Average	Min	Max	Writes	Writes l...	Data Size
0	?	?	?	?	?	?	?	?	?	?	?	?
1	?	?	?	?	?	?	?	?	?	?	?	?

Set all values to zero when the connection to a PLC is lost
If this option is enabled, all measured values of the PLC are set to zero as soon as the connection is lost. If this option is disabled, *ibaPDA* keeps the last valid measured value in memory when the connection is lost.

Start acquisition even if a PLC is not accessible
If this option is enabled, the acquisition starts even if the controller is not accessible. A warning is prompted in the validation dialog, not an error. If the system was started without a connection to the controller, *ibaPDA* periodically tries to connect to the PLC.

Allow inaccessible parameters
Enable this option to start acquisition even if parameters are not accessible. These inaccessible parameters are prompted as warnings in the validation dialog box, not as errors. This can only occur if the address book is not up-to-date.

If you do not enable this option, measurement does not start when inaccessible parameters are present.

Enable ABB-Xplorer outputs
Enable this option for activating the output modules. With the ABB-Xplorer outputs you can write directly on ABB controllers. See also ↗ *Output module*, page 21.

Connection table
For each connection, the table shows the connection status, the current values for the update time (actual value, average, min. and max.) as well as the data size. In addition, there is an error counter for the individual connections during the acquisition.

See ↗ *Connection table*, page 26.

<Manage address books>
This button opens the address book management of *ibaPDA*. The table shows a list of all of the address books currently present in the system with IP address of the PLC from which the address book was created, as well as the date of creation, size and modules that were configured for the respective CPU. Use the <Delete selected address books> button to delete selected address books.

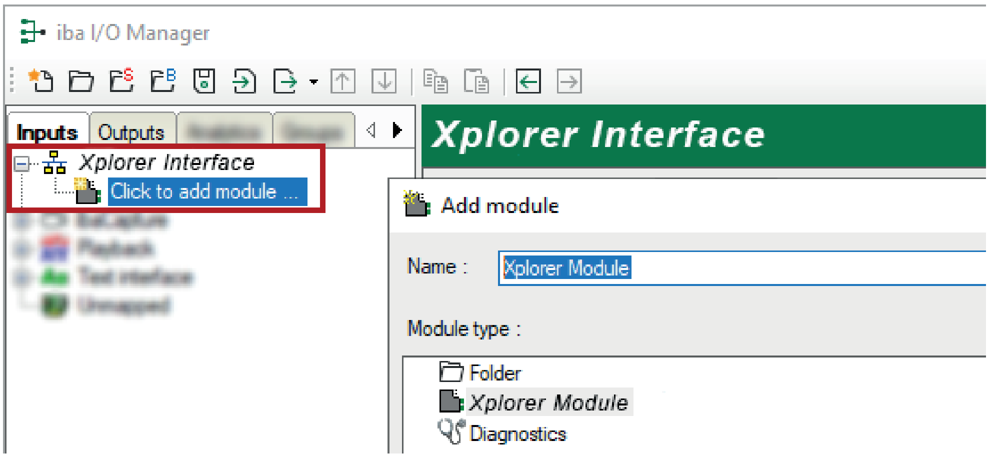
<Open log file>
If connections to controllers have been established, all connection specific actions are recorded in a text file. Using this button, you can open and check this file. In the file system on the hard disk, you find the log files of this interface in the path …`\ProgramData\iba\ibaPDA\Log`. The file name of the current log file is `InterfaceLog.txt`; the name of the archived log files is `InterfaceLog_yyyy_mm_dd_hh_mm_ss.txt`.

<Reset statistics>
Click this button to reset the calculated times and error counters in the table to 0.

3.3.2 Adding a module

To add a new module to the Xplorer interface, proceed as follows.

1. Click on the blue link *Click to add module* located under each data interface in the *Inputs* or *Outputs* tab.
2. Select the desired module type in the dialog box and assign a name via the input field if required.
3. Confirm the selection with <OK>.



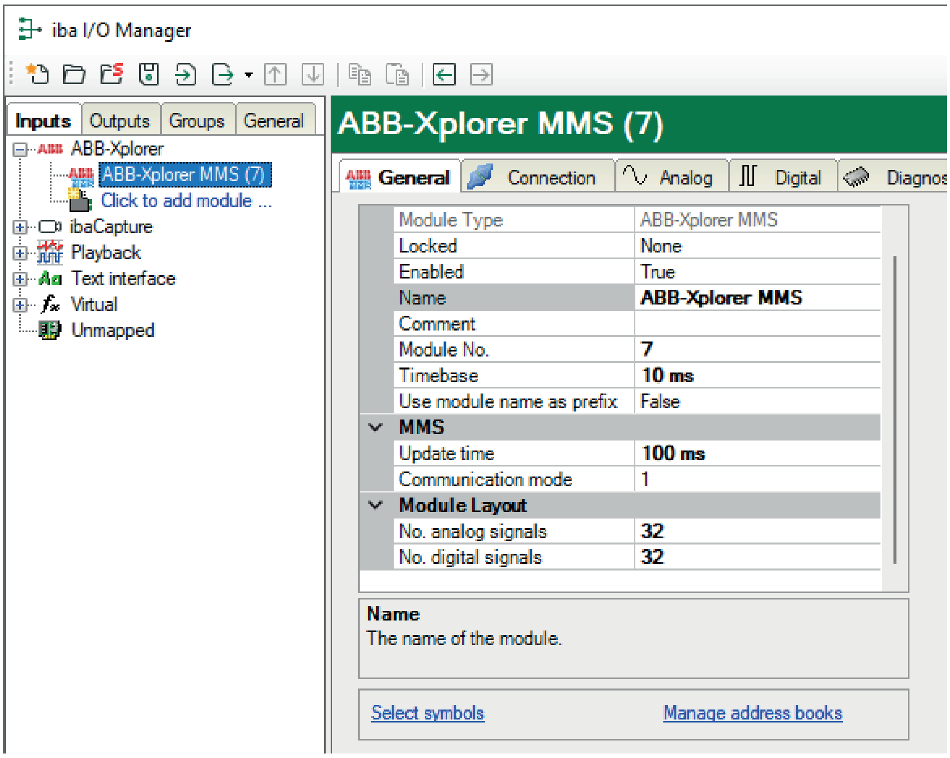
Name : Xplorer Module

Module type : Xplorer Module

3.3.3 General module settings

To configure a module, select it in the tree structure.

All modules have the following setting options.



Module Type	ABB-Xplorer MMS
Locked	None
Enabled	True
Name	ABB-Xplorer MMS
Comment	
Module No.	7
Timebase	10 ms
Use module name as prefix	False
MMS	
Update time	100 ms
Communication mode	1
Module Layout	
No. analog signals	32
No. digital signals	32

Name
The name of the module.

Basic settings

Module Type (information only)
Indicates the type of the current module.

Locked
You can lock a module to avoid unintentional or unauthorized changing of the module settings.

Enabled
Enable the module to record signals.

Name
You can enter a name for the module here.

Comment
You can enter a comment or description of the module here. This will be displayed as a tooltip in the signal tree.

Module No.
This internal reference number of the module determines the order of the modules in the signal tree of *ibaPDA* client and *ibaAnalyzer*.

Timebase
All signals of the module are sampled on this timebase.

Use module name as prefix

This option puts the module name in front of the signal names.

MMS

Update time

Specifies the reference update time in which the data is requested from the PLC. During measurement, the real current update time may be higher than the specified value if the PLC needs more time to transmit the data. You can check in the connection table how fast the data is actually updated.

Communication mode

Displays the communication mode used: Mode *1* for older systems; Mode *2* skips the checksum for newer systems.

Module Layout

No. analog signals/No. digital signals

Define the number of configurable analog and digital signals in the signal tables. The default value is 32 for each. The maximum value is 1000. The signal tables are adjusted accordingly.

"Select symbols" link

Click on this link after the connection has been successfully established to configure the signals to be measured.

"Manage address books” link

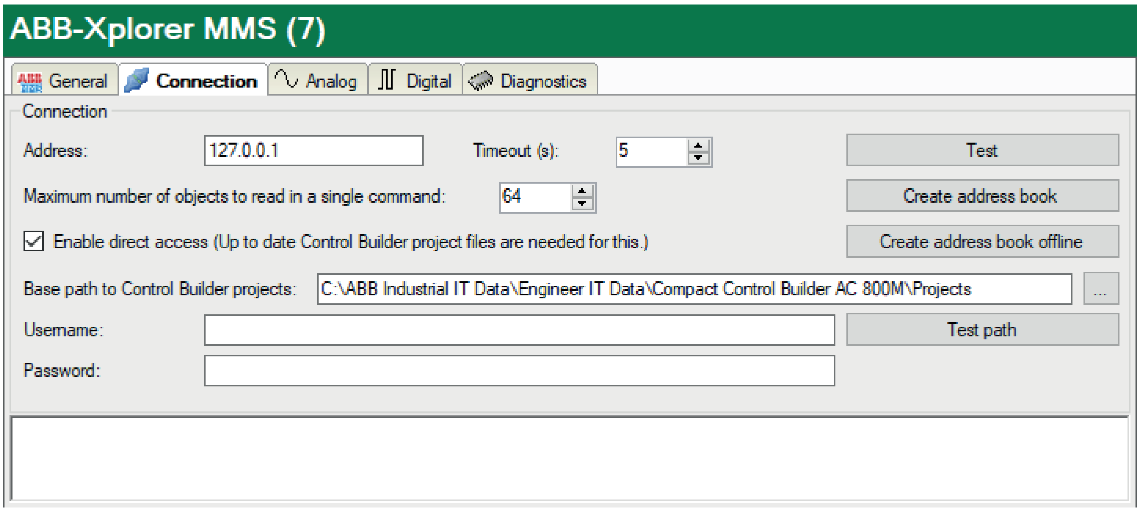
If you click on the *Manage address books* link, the *Address books* node opens in the general settings of *ibaPDA*.

Other documentation

Further information on the configuration of address books can be found in the *ibaPDA* documentation.

3.3.4 Connection settings

Configure the connection of the module to the controller in the *Connection* tab.



Address: 127.0.0.1

Timeout (s): 5

Maximum number of objects to read in a single command: 64

☑ Enable direct access (Up to date Control Builder project files are needed for this.)

Base path to Control Builder projects: C:\ABB Industrial IT Data\Engineer IT Data\Compact Control Builder AC 800M\Projects

Username:

Password:

Address
IP address of the controller

Timeout
Specify a value for the timeout in seconds for establishing the connection and for read access. If the time set here is exceeded, *ibaPDA* declares the controller as not accessible or not responsive.

Maximum number of objects to be read in a single command
ibaPDA tries to group the requested elements in as few requests as possible. If the capacity utilization of the PLC-CPU is too high, this leads to smaller packets that can be processed without a timeout due to the limitation of the parameters.

<Test>
ibaPDA tests the connection to the CPU and displays available diagnostic data.

<Create address book>
This button establishes a connection to the PLC. *ibaPDA* reads the address book and saves it on the *ibaPDA* server. In the process, *ibaPDA* overwrites all previous address books relating to the selected IP address.

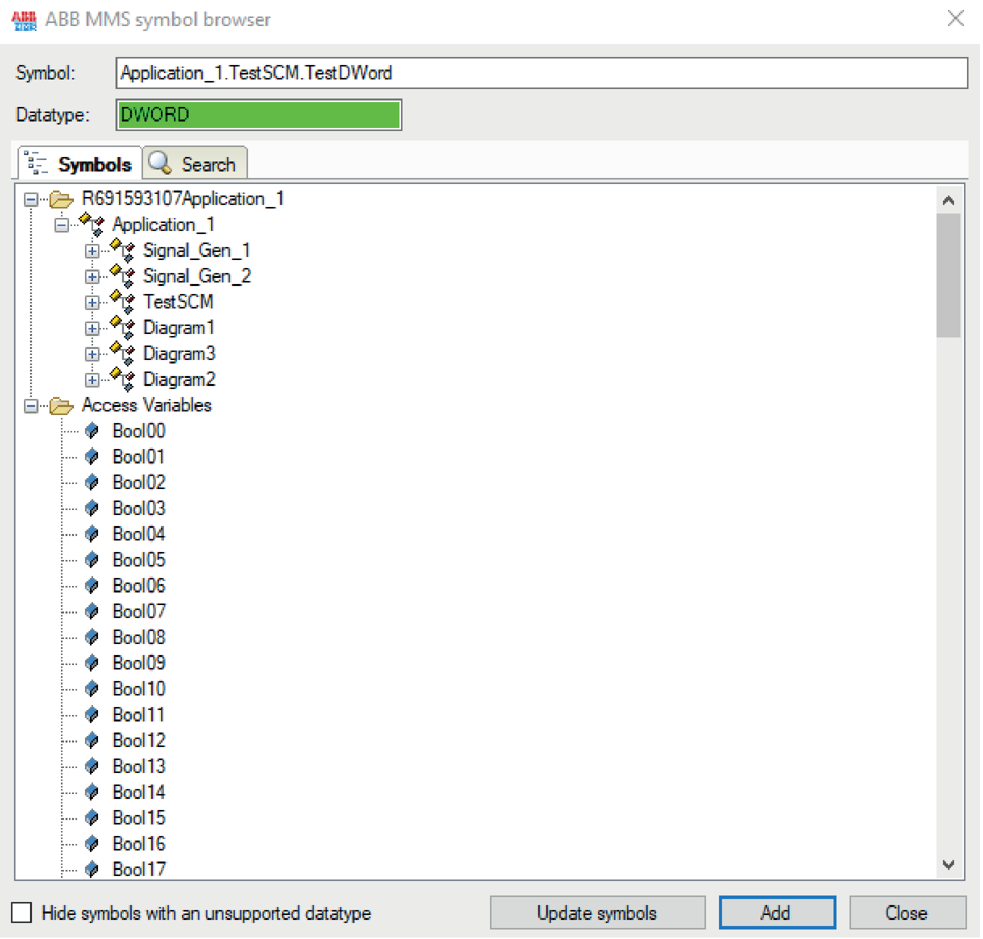
<Create address book offline>
This button generates the address book from the project.

Enable direct access (Up to date Control Builder project files are needed for this.)
In order to use the ABB Compact Control Builder project files to access internal variables of the PLC, you have to configure additional parameters.

On the PC where ABB Compact Control Builder is installed, this path is usually:
`C:\ABB Industrial IT Data\Engineer IT Data\Compact Control Builder AC 800M\Projects`

You can use this project folder either directly when *ibaPDA* is running on the same computer or you can share the folder over the network. In this dialog, you can also enter the user name and password for access sharing. If you copy the folders to another location, you have to update these copies in *ibaPDA* as soon as you make changes in the ABB Compact Control Builder and in the *ibaPDA* address book. Otherwise, access to the variables is denied and a message is displayed that you have to update the address book.

The selected folder must be the base path that contains all projects. *ibaPDA* then selects the corresponding project folder automatically based on the information from the PLC.



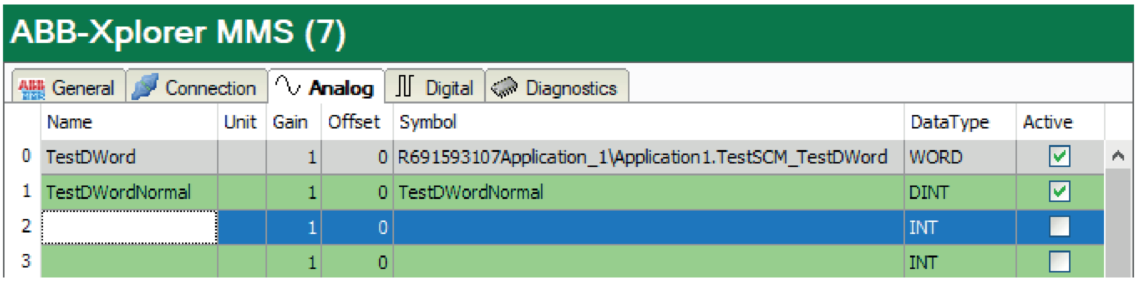
Symbol: Application_1.TestSCM.TestDWord

Datatype: DWORD

```
R691593107Application_1
  Application_1
    Signal_Gen_1
    Signal_Gen_2
    TestSCM
    Diagram1
    Diagram3
    Diagram2
Access Variables
  Bool00
  Bool01
  Bool02
  Bool03
  Bool04
  Bool05
  Bool06
  Bool07
  Bool08
  Bool09
  Bool10
  Bool11
  Bool12
  Bool13
  Bool14
  Bool15
  Bool16
  Bool17
```

☐ Hide symbols with an unsupported datatype

After creating the new address book, an additional branch appears in the symbol browser containing the variables for the imported project, including the named structure elements. The symbol created from this contains the complete variable path, in contrast to the direct name of the access variable.



	Name	Unit	Gain	Offset	Symbol	DataType	Active
0	TestDWord		1	0	R691593107Application_1\Application1.TestSCM_TestDWord	WORD	☑
1	TestDWordNormal		1	0	TestDWordNormal	DINT	☑
2			1	0		INT	☐
3			1	0		INT	☐

3.3.5 Signal configuration

In the *Analog* or *Digital* tab, you configure the signals to be measured. In the *General* tab under *Module Layout* you define the length of the signal tables or the number of signals per table.

Note

Observe the maximum number of signals permitted by your license.

Note

Take into consideration that the number of signals, which are read by a CPU, influences the minimum achievable read cycle. The more signals recorded, the slower the reachable reading cycle.

Selection of measuring signals

You have two options to select the signals to be measured:

- In the module's *General* tab, click on the *Select symbols* link.

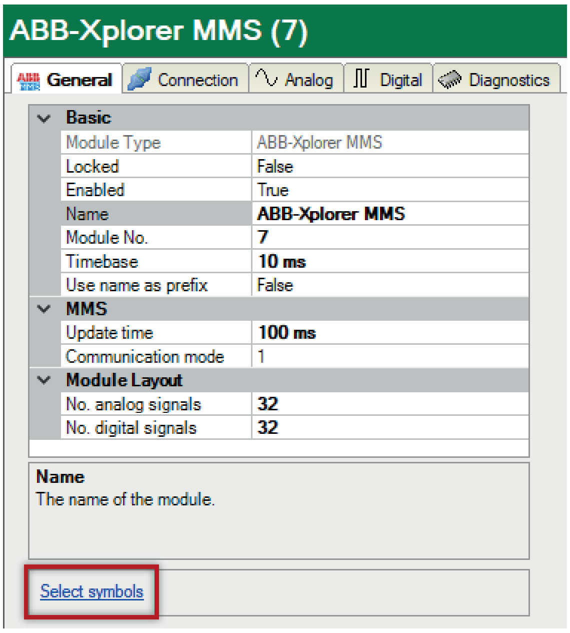


ABB-Xplorer MMS (7)

Basic	
Module Type	ABB-Xplorer MMS
Locked	False
Enabled	True
Name	**ABB-Xplorer MMS**
Module No.	**7**
Timebase	**10 ms**
Use name as prefix	False
MMS	
Update time	**100 ms**
Communication mode	1
Module Layout	
No. analog signals	**32**
No. digital signals	**32**

Name
The name of the module.

The symbol browser opens.

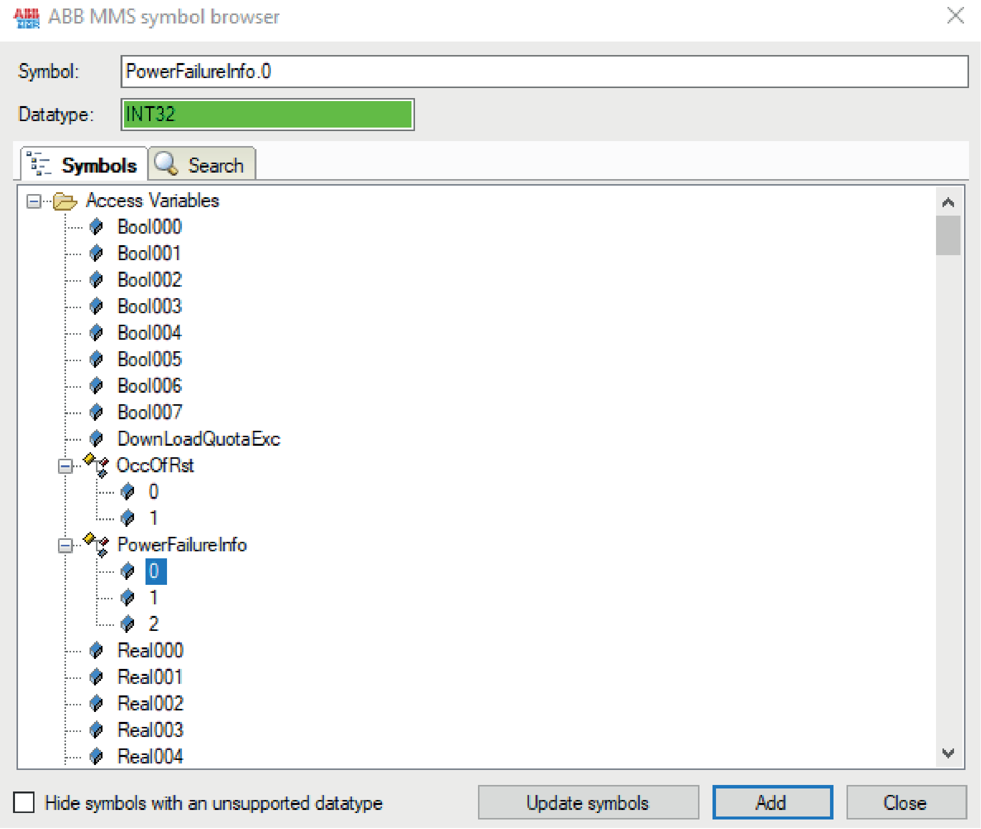
In the symbol browser, you can select all symbols of the address book. *ibaPDA* enters the selected signals automatically in the appropriate table *Analog* or *Digital*. You can add several signals successively.

- On the *Analog* or *Digital* tab, click in a cell of the *Symbol* column.

The button <...> appears. Click this button to open the symbol browser.

In the symbol browser, you can only select symbols with the data type matching the table. *ibaPDA* enters the symbol in the previously selected cell. After every selection, the symbol browser closes.

ABB MMS symbol browser



Symbol: PowerFailureInfo.0

Datatype: INT32

For the variables to be visible in the symbol browser, you must have configured and tested a connection to the target PLC. You can only use variables that you have defined in the ABB Compact Control Builder as "ABB MMS access variables", see ↗ *Access variables in the ABB Compact Control Builder*, page 10

Meaning of the colors:

Green	The operand is valid and can be included in the module setting with OK.
Yellow	The operand has a data type that does not match the selected column, e.g., if you have selected a Boolean variable.

On the *Search* tab, you can search variables by name. The search result tree works in the same way as the complete variable tree.

You can hide all unsupported datatypes by activating the *Hide symbols with an unsupported datatype* checkbox.

3.3.6 Output module

The output module is not an autonomous module, but rather an extension of the *ABB-Xplorer MMS* module. With the output module, you can write data from *ibaPDA* to a controller.

Note

Depending on the configured protection-level access of the ABB CPU, writing values to the CPU may be not possible.

Enable the ABB-Xplorer outputs in the interface settings, see ➚ *General interface settings*, page 13.

Note

A warning indicates that the ABB system does not contain any access control and can therefore be written to all variables. This can lead to malfunctions of the program running on the PLC. The user may only select variables that are intended for this purpose by *ibaPDA*.

You can configure the module via the *Outputs* tab. You do not have to add it separately. The module is available as soon as you add an *ABB-Xplorer MMS* module in the *Inputs* register.

The settings correspond to those in the *Inputs* tab, apart from the module-specific settings, and can also be configured there. The connection settings also correspond to those in the *Inputs* tab.

For more information on the module settings, see ➚ *General module settings*, page 15.

Module-specific settings

Calculation timebase
Timebase (in ms) used for the calculation of the output values.

Technically, the calculation timebase is identical to the timebase of the input module. This means a change in the calculation timebase also changes the module timebase of the input side and vice versa!

The calculation timebase is not the same as the output timebase, with which the values are output!

Minimum output timebase
Timebase with which the outputs can be updated as quickly as possible.

The value is acquired automatically by the system based on the current I/O configuration and is only displayed here. The output timebase results from the smallest common multiple of all module timebases or is at least 50 ms.

Send mode
Determines when new data is written to the controller:

- Cyclic: Data is written cyclically at the set update time.
- On change: Data is written each time the signal data is changed.
- On trigger: Data is written with every rising edge of the trigger signal.

All signals of a module are always written, regardless of the write mode.

Trigger signal

This field only appears when the "on trigger" send mode is selected. Select here a digital signal. A rising edge on this digital signal writes the signal values taken at the time of the rising edge.

Number of analog/digital output signals

Define the number of configurable analog and digital signals in the signal tables. The default value is 32 for each. The maximum value is 1000. The signal tables are adjusted accordingly.

Signal configuration

You can configure the signals to be output in each case via the expression editor. Open the expression editor via the <fx> button in each signal row.

Other documentation

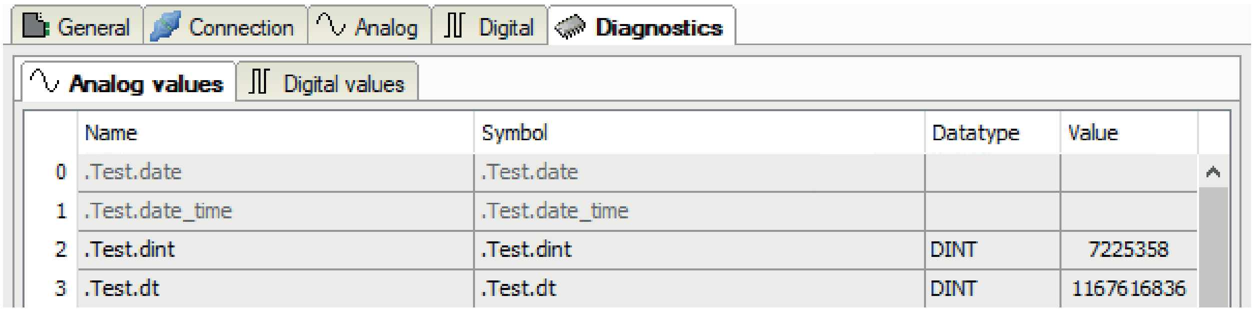
Further information about the expression editor can be found in the *ibaPDA* manual, part 4.

Note

ibaPDA reads and writes all signals for an Xplorer module via a common connection. Therefore, the total number of configured signals influences the update time.

3.3.7 Module diagnostics

After applying the configuration the actual values of the analog and digital signals are displayed in the *Diagnostics* tab of the relevant module.



	Name	Symbol	Datatype	Value
0	.Test.date	.Test.date		
1	.Test.date_time	.Test.date_time		
2	.Test.dint	.Test.dint	DINT	7225358
3	.Test.dt	.Test.dt	DINT	1167616836

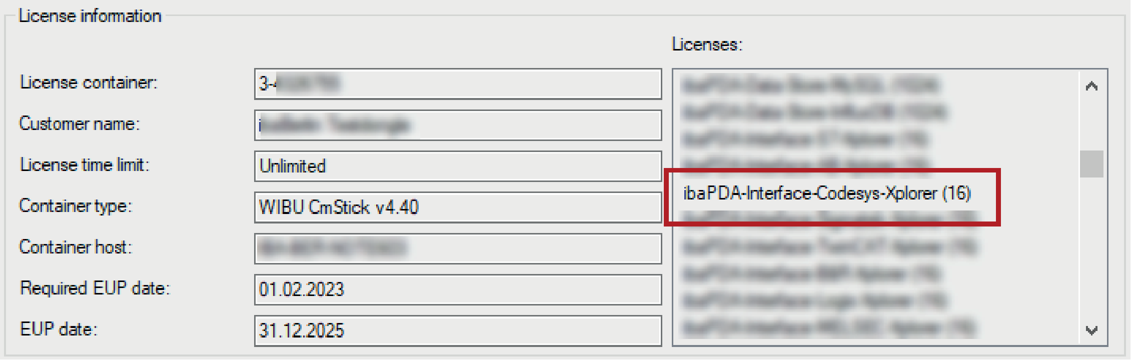
Inactive signals are grayed out.

4 Diagnostics

4.1 License

If the interface is not displayed in the signal tree, you can either check in *ibaPDA* in the I/O Manager under *General – Settings* or in the *ibaPDA* service status application whether your license for the interface *ibaPDA-Interface-ABB-Xplorer* has been properly recognized. The number of licensed connections is shown in brackets.

The figure below shows the license for the *Codesys Xplorer* interface as an example.



License information

License container:	3-[redacted]
Customer name:	[redacted]
License time limit:	Unlimited
Container type:	WIBU CmStick v4.40
Container host:	[redacted]
Required EUP date:	01.02.2023
EUP date:	31.12.2025

Licenses:

[redacted]
ibaPDA-Interface-Codesys-Xplorer (16)
[redacted]

4.2 Visibility of the interface

If the interface is not visible despite a valid license, it may be hidden.

Check the settings in the *General* tab in the *Interfaces* node.

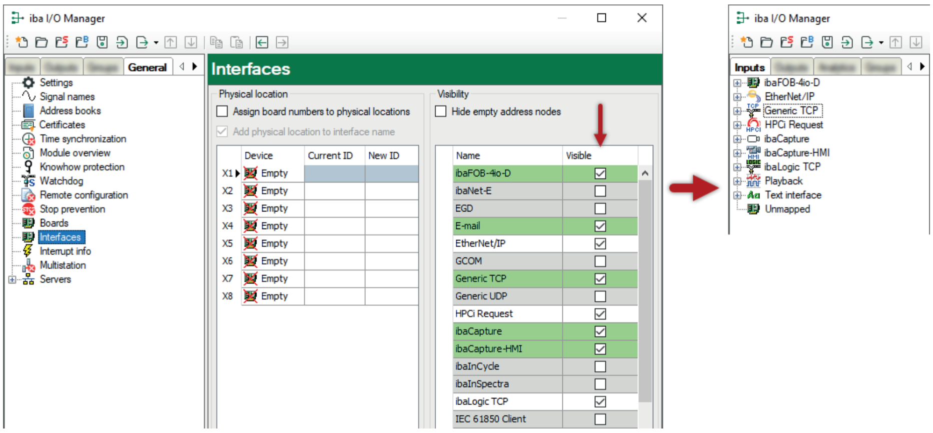
Visibility

The table *Visibility* lists all the interfaces that are available either through licenses or installed cards. These interfaces can also be viewed in the interface tree.

You can hide or display the interfaces not required in the interface tree by using the checkbox in the *Visible* column.

Interfaces with configured modules are highlighted in green and cannot be hidden.

Selected interfaces are visible, the others are hidden:



Physical location

☐ Assign board numbers to physical locations

☑ Add physical location to interface name

	Device	Current ID	New ID
X1	Empty		
X2	Empty		
X3	Empty		
X4	Empty		
X5	Empty		
X6	Empty		
X7	Empty		
X8	Empty		

Visibility

☐ Hide empty address nodes

Name	Visible
ibaFOB-4io-D	☑
ibaNet-E	☐
EGD	☐
E-mail	☑
EtherNet/IP	☑
GCOM	☐
Generic TCP	☑
Generic UDP	☐
HPCi Request	☑
ibaCapture	☑
ibaCapture-HMI	☑
ibaInCycle	☐
ibaInSpectra	☐
ibaLogic TCP	☑
IEC 61850 Client	☐

4.3 Log files

If connections to target platforms or clients have been established, all connection-specific actions are logged in a text file. You can open this (current) file and, e.g., scan it for indications of possible connection problems.

You can open the log file via the button <Open log file>. The button is available in the I/O Manager:

- for many interfaces in the respective interface overview
- for integrated servers (e.g. OPC UA server) in the *Diagnostics* tab.

In the file system on the hard drive, you can find the log files of the *ibaPDA* server (…`\ProgramData\iba\ibaPDA\Log`). The file names of the log files include the name or abbreviation of the interface type.

Files named `interface.txt` are always the current log files. Files named `Interface_yyyy_mm_dd_hh_mm_ss.txt` are archived log files.

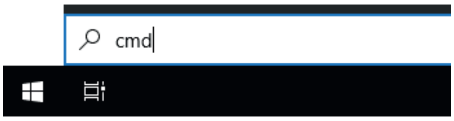
Examples:

- `ethernetipLog.txt` (log of EtherNet/IP connections)
- `AbEthLog.txt` (log of Allen-Bradley Ethernet connections)
- `OpcUAServerLog.txt` (log of OPC UA server connections)

4.4 Connection diagnostics with PING

PING is a system command with which you can check if a certain communication partner can be reached in an IP network.

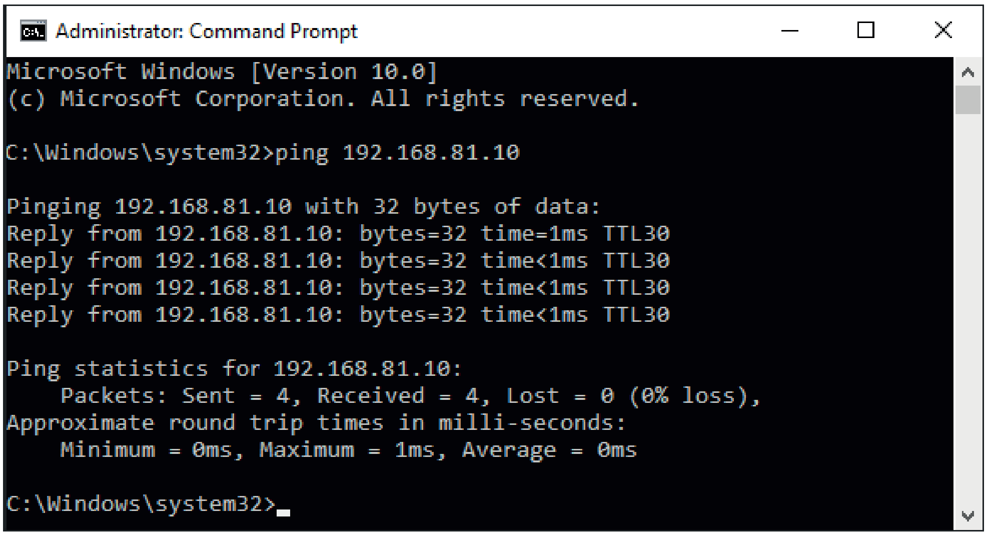
1. Open a Windows command prompt.



```
cmd
```

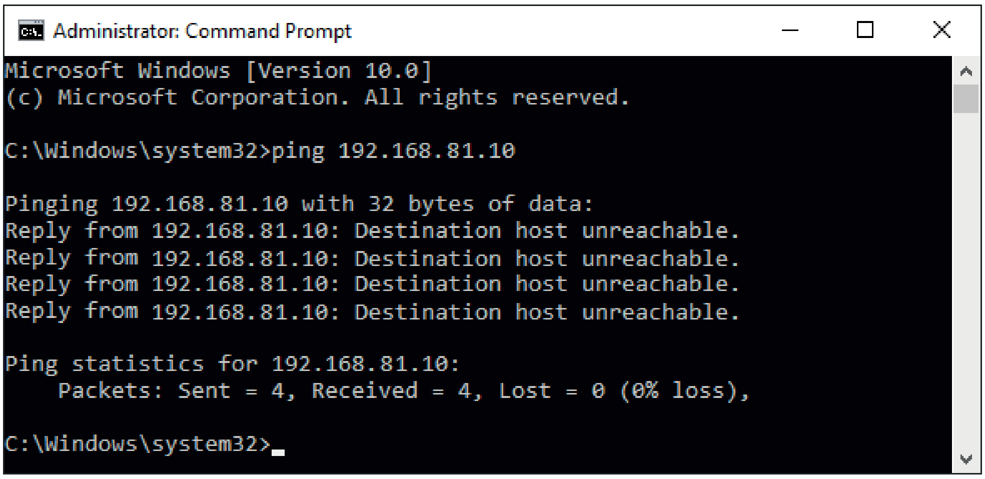
2. Enter the command "ping" followed by the IP address of the communication partner and press <ENTER>.

→ With an existing connection you receive several replies.



```
Microsoft Windows [Version 10.0]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>ping 192.168.81.10

Pinging 192.168.81.10 with 32 bytes of data:
Reply from 192.168.81.10: bytes=32 time=1ms TTL30
Reply from 192.168.81.10: bytes=32 time<1ms TTL30
Reply from 192.168.81.10: bytes=32 time<1ms TTL30
Reply from 192.168.81.10: bytes=32 time<1ms TTL30

Ping statistics for 192.168.81.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\Windows\system32>_
```

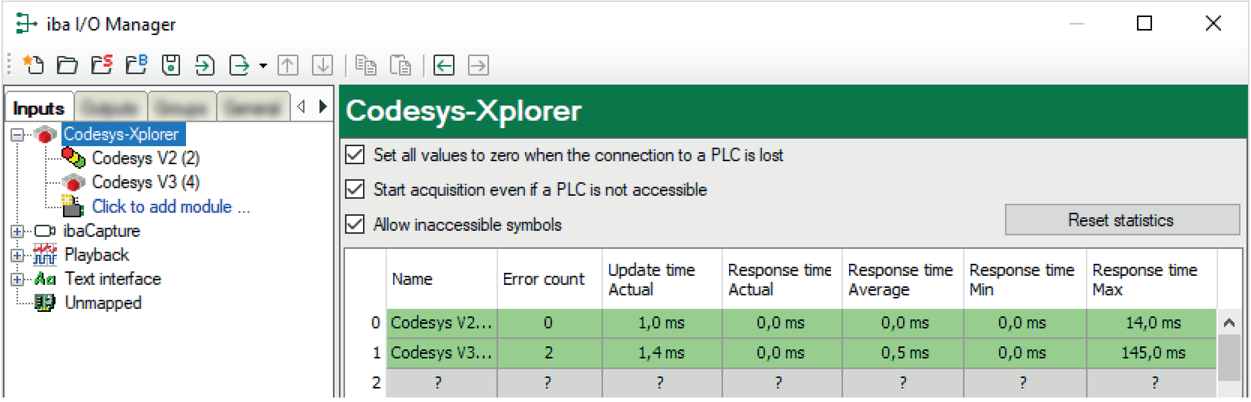
→ With no existing connection you receive error messages.



```
Microsoft Windows [Version 10.0]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>ping 192.168.81.10

Pinging 192.168.81.10 with 32 bytes of data:
Reply from 192.168.81.10: Destination host unreachable.
Reply from 192.168.81.10: Destination host unreachable.
Reply from 192.168.81.10: Destination host unreachable.
Reply from 192.168.81.10: Destination host unreachable.

Ping statistics for 192.168.81.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

C:\Windows\system32>_
```

4.5 Connection table

For every Ethernet-based interface, there is a table available in the I/O Manager which shows the status of each connection. Each line represents one connection. The following figure shows, as an example, the connection table of the Codesys-Xplorer interface:



Codesys-Xplorer

☑ 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	Update time Actual	Response time Actual	Response time Average	Response time Min	Response time Max
0	Codesys V2...	0	1,0 ms	0,0 ms	0,0 ms	0,0 ms	14,0 ms
1	Codesys V3...	2	1,4 ms	0,0 ms	0,5 ms	0,0 ms	145,0 ms
2	?	?	?	?	?	?	?

The connected target systems (controllers) are identified by their name or IP address in the first (left) column.

Depending on the interface type the table shows error counters, read counters and/or data sizes, as well as the cycle times, refresh times and/or update times of the different connections during the data acquisition.

Click the <Reset statistics> button to reset the error counters and the calculation of the response times.

Additional information is provided by the background color of the table rows:

Color	Meaning
Green	The connection is OK and the data are read.
Yellow	The connection is OK, however the data update is slower than the configured update time.
Red	The connection has failed.
Gray	No connection configured.

4.6 Diagnostic modules

Diagnostic modules are available for most Ethernet based interfaces and Xplorer interfaces. Using a diagnostic module, information from the diagnostic displays (e.g. diagnostic tabs and connection tables of an interface) can be acquired as signals.

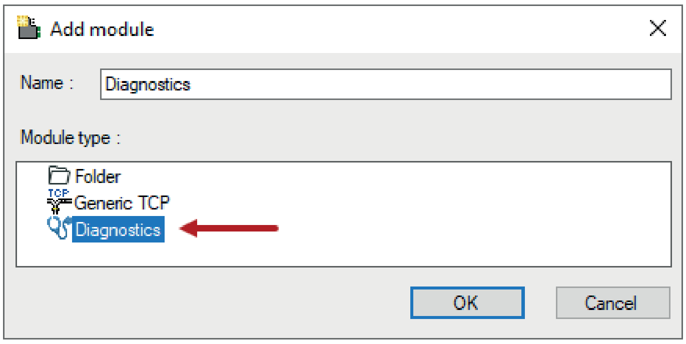
A diagnostic module is always assigned to a data acquisition module of the same interface and supplies its connection information. By using a diagnostic module, you can record and analyze the diagnostic information continuously in the *ibaPDA* system.

Diagnostic modules do not consume any license connections because they do not establish their own connection but refer to another module.

Example for the use of diagnostic modules:

- A notification can be generated, whenever the error counter of a communication connection exceeds a certain value or the connection gets lost.
- In case of a disturbance, the current response times in the telegram traffic may be documented in an incident report.
- The connection status can be visualized in *ibaQPanel*.
- You can forward diagnostic information via the SNMP server integrated in *ibaPDA* or via OPC DA/UA server to superordinate monitoring systems like network management tools.

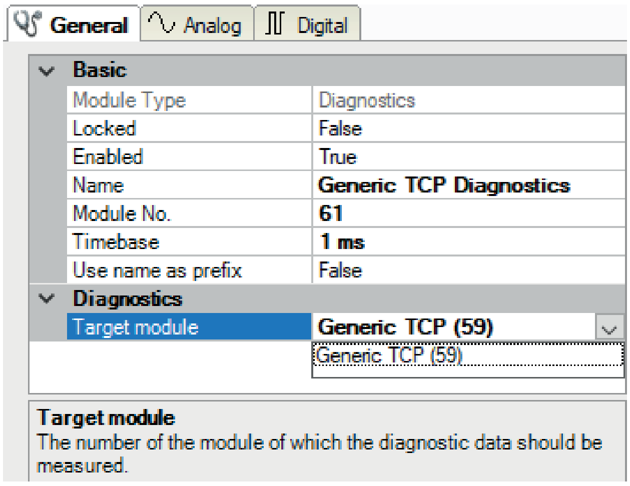
In case the diagnostic module is available for an interface, a "Diagnostics" module type is shown in the *Add module* dialog (example: Generic TCP).



Name : Diagnostics

Module settings diagnostic module

For a diagnostic module, you can make the following settings (example: Generic TCP):



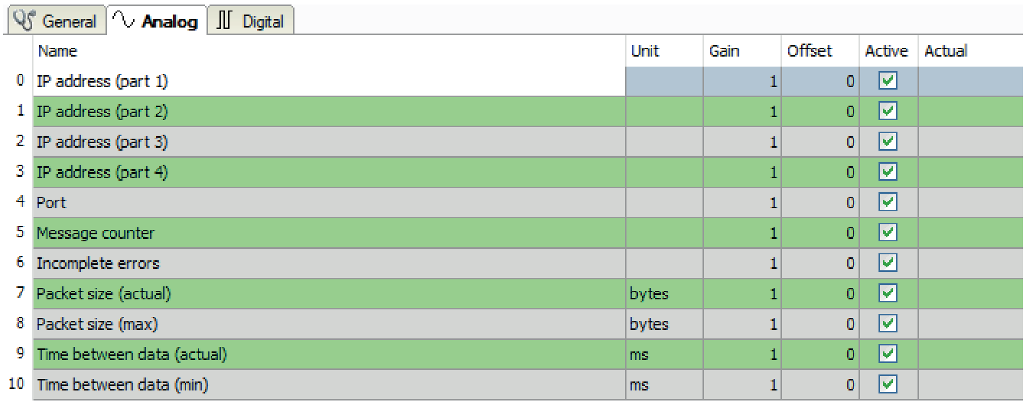
Basic	
Module Type	Diagnostics
Locked	False
Enabled	True
Name	**Generic TCP Diagnostics**
Module No.	**61**
Timebase	**1 ms**
Use name as prefix	False
Diagnostics	
Target module	**Generic TCP (59)**
	Generic TCP (59)

Target module
The number of the module of which the diagnostic data should be measured.

The basic settings of a diagnostic module equal those of other modules.

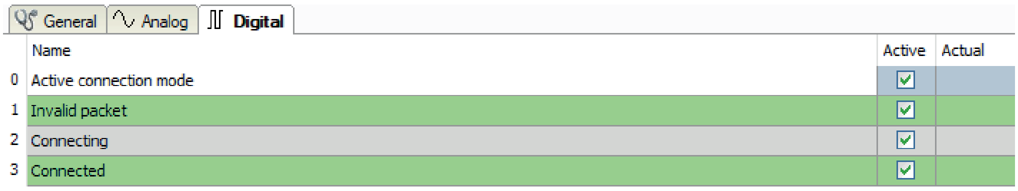
There is only one setting which is specific for the diagnostic module: the target module.

By selecting the target module, you assign the diagnostic module to the module on which you want to acquire information about the connection. You can select the supported modules of this interface in the drop-down list of the setting. You can assign exactly one data acquisition module to each diagnostic module. When having selected a module, the available diagnostic signals are immediately added to the *Analog* and *Digital* tabs. It depends on the type of interface, which signals exactly are added. The following example lists the analog values of a diagnostic module for a Generic TCP module.



	Name	Unit	Gain	Offset	Active	Actual
0	IP address (part 1)		1	0	☑	
1	IP address (part 2)		1	0	☑	
2	IP address (part 3)		1	0	☑	
3	IP address (part 4)		1	0	☑	
4	Port		1	0	☑	
5	Message counter		1	0	☑	
6	Incomplete errors		1	0	☑	
7	Packet size (actual)	bytes	1	0	☑	
8	Packet size (max)	bytes	1	0	☑	
9	Time between data (actual)	ms	1	0	☑	
10	Time between data (min)	ms	1	0	☑	

For example, the IP (v4) address of a Generic TCP module (see fig. above) will always be split into 4 parts derived from the dot-decimal notation, for better reading. Also other values are being determined, as there are port number, counters for telegrams and errors, data sizes and telegram cycle times. The following example lists the digital values of a diagnostic module for a Generic TCP module.



	Name	Active	Actual
0	Active connection mode	☑	
1	Invalid packet	☑	
2	Connecting	☑	
3	Connected	☑	

Diagnostic signals

Depending on the interface type, the following signals are available:

Signal name	Description
Active	Only relevant for redundant connections. Active means that the connection is used to measure data, i.e. for redundant standby connections the value is 0. For normal/non-redundant connections, the value is always 1.
Buffer file size (actual/avg/max)	Size of the file for buffering statements
Buffer memory size (actual/avg/max)	Size of the memory used by buffered statements
Buffered statements	Number of unprocessed statements in the buffer
Buffered statements lost	Number of buffered but unprocessed and lost statements
Connected	Connection is established
Connected (in)	A valid data connection for the reception (in) is available
Connected (out)	A valid data connection for sending (out) is available
Connecting	Connection being established
Connection attempts (in)	Number of attempts to establish the receive connection (in)
Connection attempts (out)	Number of attempts to establish the send connection (out)
Connection ID O->T	ID of the connection for output data (from the target system to *ibaPDA*). Corresponds to the assembly instance number
Connection ID T->O	ID of the connection for input data (from *ibaPDA* to target system). Corresponds to the assembly instance number
Connection phase (in)	Status of the ibaNet-E data connection for reception (in)
Connection phase (out)	Status of the ibaNet-E data connection for sending (out)
Connections established (in)	Number of currently valid data connections for reception (in)
Connections established (out)	Number of currently valid data connections for sending (out)
Data length	Length of the data message in bytes
Data length O->T	Size of the output message in byte
Data length T->O	Size of the input message in byte
Destination IP address (part 1-4) O->T	4 octets of the IP address of the target system Output data (from target system to *ibaPDA*)
Destination IP address (part 1-4) T->O	4 octets of the IP address of the target system Input data (from *ibaPDA* to target system)
Disconnects (in)	Number of currently interrupted data connections for reception (in)
Disconnects (out)	Number of currently interrupted data connections for sending (out)
Error counter	Communication error counter
Exchange ID	ID of the data exchange
Incomplete errors	Number of incomplete messages

Signal name	Description
Incorrect message type	Number of received messages with wrong message type
Input data length	Length of data messages with input signals in bytes (*ibaPDA* receives)
Invalid data points	Number of received data points with missing configuration
Invalid packet	Invalid data packet detected
IP address (part 1-4)	4 octets of the IP address of the target system
Keepalive counter	Number of Keepalive messages received by the OPC UA Server
Lost images	Number of lost images (in) that were not received even after a retransmission
Lost Profiles	Number of incomplete/incorrect profiles
Message counter	Number of messages received
Messages per cycle	Number of messages in the cycle of the update time
Messages received since configuration	Number of received data telegrams (in) since start of acquisition
Messages received since connection start	Number of received data telegrams (in) since the start of the last connection setup. Reset with each connection loss.
Messages sent since configuration	Number of sent data telegrams (out) since start of acquisition
Messages sent since connection start	Number of sent data telegrams (out) since the start of the last connection setup. Reset with each connection loss.
Multicast join error	Number of multicast login errors
Number of request commands	Counter for request messages from *ibaPDA* to the PLC/CPU
Output data length	Length of the data messages with output signals in bytes (*ibaPDA* sends)
Packet size (actual)	Size of the currently received message
Packet size (max)	Size of the largest received message
Ping time (actual)	Response time for a ping telegram
Port	Port number for communication
Producer ID (part 1-4)	Producer ID as 4-byte unsigned integer
Profile Count	Number of completely recorded profiles
Read counter	Number of read accesses/data requests
Receive counter	Number of messages received
Response time (actual/average/max/min)	Response time is the time between measured value request from *ibaPDA* and response from the PLC or reception of the data. Actual: current value Average/max/min: static values of the update time since the last start of the acquisition or reset of the counters.
Retransmission requests	Number of data messages requested again if lost or delayed

Signal name	Description
Rows (last)	Number of resulting rows by the last SQL query (within the configured range of result rows)
Rows (maximum)	Maximum number of resulting rows by any SQL query since the last start of acquisition (possible maximum equals the configured number of result rows)
Send counter	Number of send messages
Sequence errors	Number of sequence errors
Source IP address (part 1-4) O->T	4 octets of the IP address of the target system Output data (from target system to *ibaPDA*)
Source IP address (part 1-4) T->O	4 octets of the IP address of the target system Input data (from *ibaPDA* to target system)
Statements processed	Number of executed statements since last start of acquisition
Synchronization	Device is synchronized for isochronous acquisition
Time between data (actual/max/min)	Time between two correctly received messages Actual: between the last two messages Max/min: statistical values since start of acquisition or reset of counters
Time offset (actual)	Measured time difference of synchronicity between *ibaPDA* and the ibaNet-E device
Topics Defined	Number of defined topics
Topics Updated	Number of updated topics
Unknown sensor	Number of unknown sensors
Update time (actual/average/configured/max/min)	Specifies the update time in which the data is to be retrieved from the PLC, the CPU or from the server (configured). Default is equal to the parameter "Timebase". During the measurement the real actual update time (actual) can be higher than the set value, if the PLC needs more time to transfer the data. How fast the data is really updated, you can check in the connection table. The minimum achievable update time is influenced by the number of signals. The more signals are acquired, the greater the update time becomes. Average/max/min: static values of the update time since the last start of the acquisition or reset of the counters.
Write counter	Number of successful write accesses
Write lost counter	Number of failed write accesses

5 Support and contact

Support

Phone:	+49 911 97282-14
Email:	support@iba-ag.com

Note

If you need support for software products, please state the number of the license container. For hardware products, please have the serial number of the device ready.

Contact

Headquarters

iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

Phone:	+49 911 97282-0
Email:	iba@iba-ag.com

Mailing address

iba AG
Postbox 1828
D-90708 Fuerth, Germany

Delivery address

iba AG
Gebhardtstrasse 10
90762 Fuerth, Germany

Regional and Worldwide

For contact data of your regional iba office or representative
please refer to our web site:

www.iba-ag.com