ibaPDA-Interface-OPC-UA Client

Data Interface OPC UA Client

Manual

Issue 3.0

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
Fax	+49 911 97282-33
Support	+49 911 97282-14
Engineering	+49 911 97282-13
E-mail	iba@iba-ag.com
Web	www.iba-ag.com

Version	Date	Revision	Author	Version SW
3.0	04-2023	Client output module, certificate store	RM	7.3.1

Content

1 About this Manual

This document describes the function and application of the software interface

ibaPDA-Interface-OPC-UA Client

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 addresses qualified professionals, who are familiar with handling electrical and electronic modules as well as communication and measurement technology. A person is regarded to 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 people, who are concerned with the configuration, test, commissioning or maintenance of control systems using OPC UA communication. For the handling of *ibaPDA-Interface-OPC-UA Client* the following basic knowledge is required and/or useful:

- Windows operating system
- Basic knowledge of *ibaPDA*
- Knowledge needed for configuring an OPC UA server

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 for the use of the OPC UA Client data interface:

- *ibaPDA* V7.3.1 or more recent
- License for *ibaPDA-Interface-OPC-UA Client*
- Network connection to one or more OPC UA servers

Note

It is recommended carrying out the OPC UA communication for data acquisition on a separate network segment to exclude an influence of the OPC UA messages by the Ethernet data traffic between *ibaPDA* and other nodes in the network (file servers, data file requirements etc.).

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

License information

Order No.	Product name	Designation
31.001.111	ibaPDA-Interface-OPC-UA Client	Extension license for an ibaPDA system adding the data interface: + OPC UA client
31.101.111	one-step-up-Interface-OPC-UA Client	Extension license for further 16 OPC UA connections; max.15

Table 1: Available OPC UA Client licenses

Note

With the *ibaPDA-Interface-OPC-UA-Client* license, the *OPC UA* interface is available in the I/O Manager. If you have an additional license *ibaPDA-OPC-UA-Server+*, you can also create and use OPC UA server modules.

3 OPC UA interface

3.1 General information

The OPC UA Client data interface is suitable for the recording of measured data from several OPC UA servers over standard network boards of the PC.

ibaPDA is not cyclically polling for new measurement data. Instead, *ibaPDA* will be notified whenever one of the values to be measured has changed.

ibaPDA can only read and not write the variables provided by the OPC UA server.

Up to 16 connections can be configured with an OPC UA Client interface on each license. A total of a maximum of 256 connections can be implemented by the additional purchase of up to 15 further one-step-up OPC UA Client licenses.

You can import or generate the certificates required for the communication between OPC UA Client (*ibaPDA*) and an OPC UA server in *ibaPDA*.

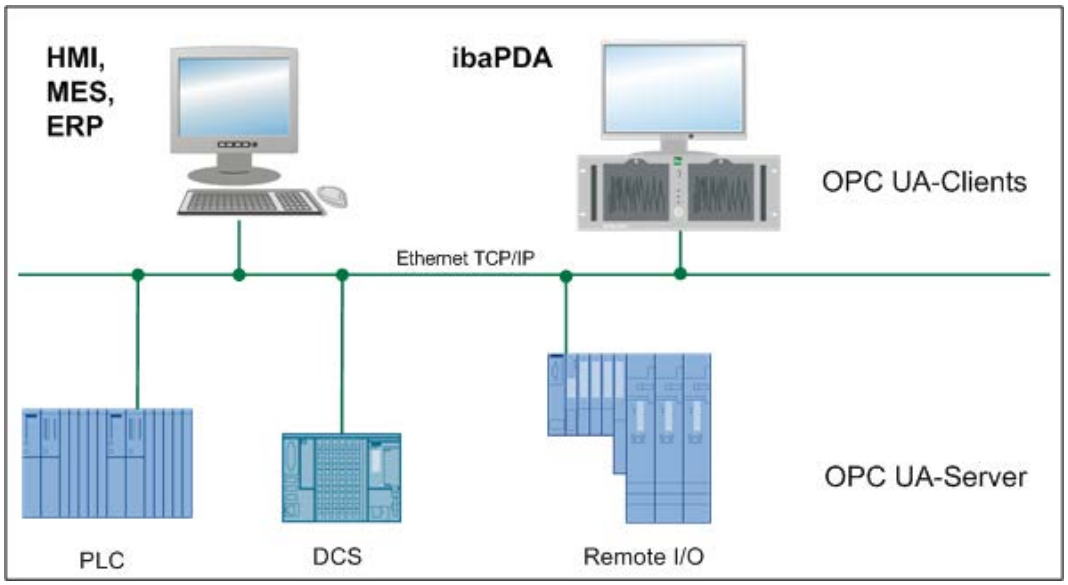
The signals to be measured can be conveniently selected using their symbolic names with support from the *OPC UA* Symbol Browser. This allows access to all measurable symbols, which are defined in the OPC UA server.

Only the acquisition of current values is supported.

The signal tree in the I/O Manager shows only an interface node named *OPC UA*, because OPC UA-Server modules can be created under this interface as well, beside the OPC UA-Client modules. Usage and configuration of OPC UA-Server modules are described in the manual of the product *ibaPDA-OPC-UA-Server+*.

3.2 System topologies

The following drawing gives an overview of a possible configuration.

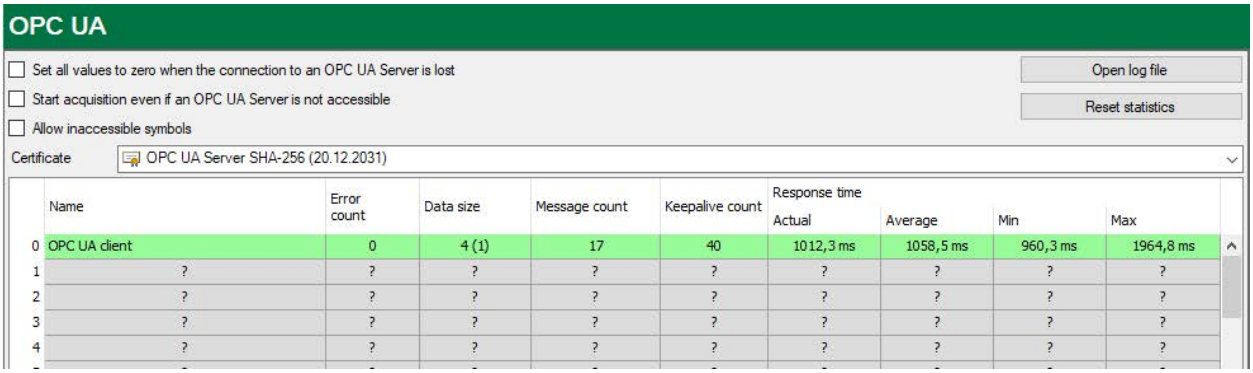


3.3 Configuration and engineering ibaPDA

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

3.3.1 Interface - Connections

The interface itself has the following functions and configuration options:



Set all values to zero when the connection to an OPC UA server is lost
If enabled, all measured values of the respective OPC UA server are set to zero as soon as the connection is lost. If this option is disabled, *ibaPDA* will keep the last valid measured value in memory at the time the connection was lost.

Start acquisition even if an OPC UA server is not accessible
If this option is enabled, the acquisition will start even if the OPC UA server is not accessible. In case of an error, a warning is indicated in the validation dialog. If the system has been started without a connection to a configured OPC UA server, *ibaPDA* will periodically try to connect to that server. The measurement values remain at zero, as long as the OPC UA server is not connected.

Allow inaccessible symbols
Enable this option if you wish to start acquisition even if symbols are not accessible. The inaccessible symbols are issued as warnings in the validation dialog box.
This can only occur, in case the symbol whose address is requested by *ibaPDA* from the OPC UA server is no longer available on the server. Then, the OPC UA server will prompt an error.

If you enable this option, *ibaPDA* will ignore this error message and starts the acquisition nonetheless.

Measurement will not start when inaccessible symbols are present if you do not enable this option.

Certificate
Select from the drop-down list the certificate that is to be used by *ibaPDA* as OPC UA client for the communication.

Only certificates that are available in the central certificate store and have been classified as trustworthy are displayed in this list.

If you have not yet created or imported a certificate, select the entry *Create new certificate* or *Manage certificates* from the list.

You will then be redirected to the central certificate store, which you can also find in the interface tree under *General - Certificates*.

The handling of the certificates is described in chapter ↗ *Certificates*, page 10

Connection table
The table shows the response time values (actual, average, minimum and maximum) and error counters for the individual connections during data measurement. To reset the calculated times and error counters to zero, simply click on the <Reset counters> button.

Please see also chapter ↗ *Connection table*, page 32

The data size column shows how much data is read per read.

<Open log file>
If connections to OPC UA servers have been established, all connection-specific actions are logged in a text file. Using this button, you can open and check this file. In the file system on the hard disk, you will find the log files in the program path of the *ibaPDA* server (...\Programs\iba\ibaPDA\Server\Log\). The file name of the current log file is `OpcUAClientLog.txt`, the name of the archived log files is `OpcUAClientLog_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 Certificates

Communication with the OPC UA server is secured by means of certificates. The certificates are managed in the central certificate store in *ibaPDA*. This store also contains other certificates, e.g., for interfaces and data stores, in addition to the certificates for OPC UA communication.

3.3.2.1 Certificates

For secure and encrypted TLS/SSL communication between a client and a server, so-called certificates are used because they enable secure authentication.

Certicates used by iba programs can be administered in a central certificate store.

Before a client can connect to a server, an application certificate must first be configured. Certificates can be provided from both the server and client side. Communication can only take place if each partner trusts the partner certificate.

Certificates can either be exchanged spontaneously when a connection is established or registered as trusted in advance. If a previously unknown certificate is offered when a connection is established, the user must manually accept or reject the certificate. Accepted certificates are automatically entered into the table in the *certificate store* and marked as trusted. If the certificate is rejected, then no communication will take place.

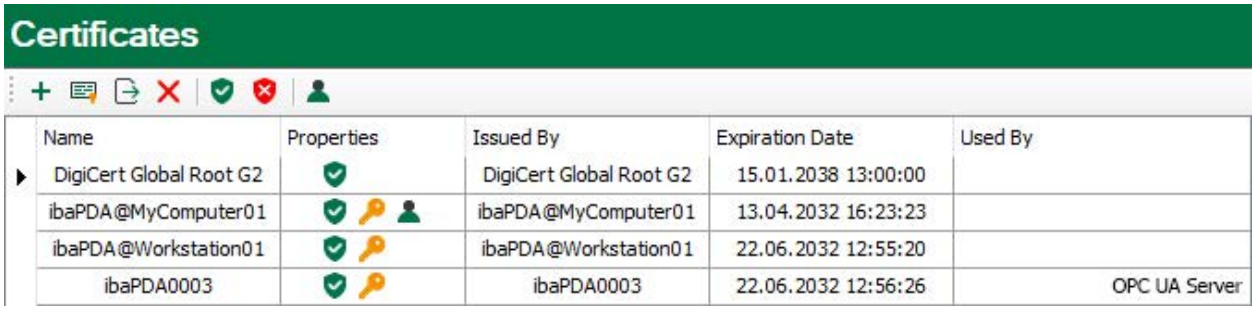
You can also register certificates and then mark them as "not trusted". Communication with partners with such certificates is then always denied. Once certificates have been registered, i.e., entered in the table in the certificate store, the user will no longer be notified or prompted when communication is established – regardless of whether the certificates are marked as "trusted" or "not trusted".

Note

Some interfaces in *ibaPDA*, such as the e-mail output, use Windows certificates. Other features, such as OPC UA server or MQTT data stores use certificates from the central certificate store of *ibaPDA*.

3.3.2.2 Central certificate store

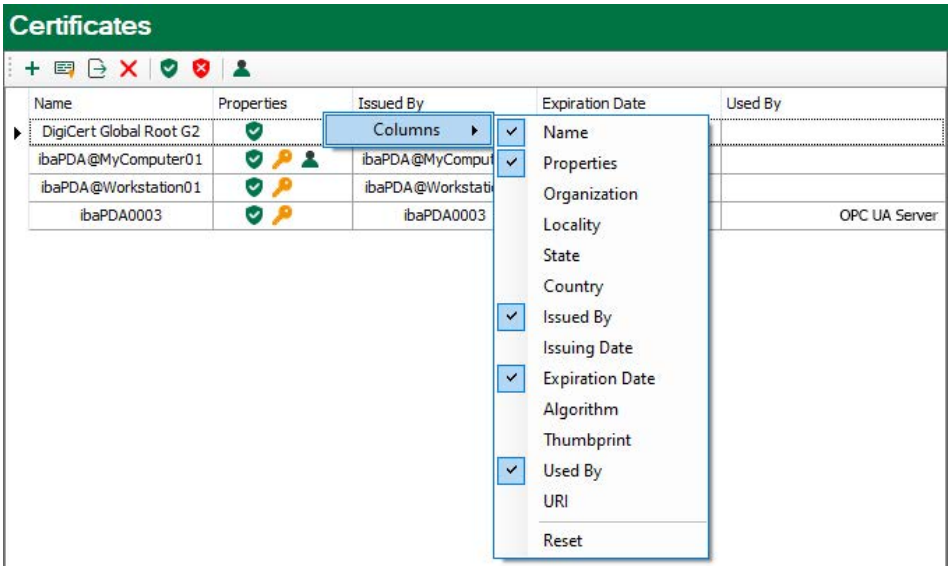
All registered certificates are listed in a table. The following figure shows the certificate store in the I/O Manager of *ibaPDA* as an example.



Each row refers to one certificate.

The columns *Name*, *Properties*, *Expiration Date* and *Used By* are displayed by default.

If needed, you may add or remove other columns via the context menu.



The *Name* column holds the name of a certificate. Different cerficates may have the same name, thus it is not unique. Only the finger print of a certificate is unique.

The symbols in the *Properties* column have the following meaning:

Symbol	Meaning
	The certificate is trusted as long as it has not expired.
	This certificate is not trusted.
	A private key for this certificate is available.
	This certificate can also be used for user authentication.
	This certificate is invalid. If the certificate is invalid because it expired, the expiration date is highlighted in red color.

Table 2: Symbols for certificate properties

The *Used By* column shows by which application/function the certificate is used. In the example shown in the figure above, the certificate *ibaPDA003@D* is used by the OPC UA server in *ibaPDA*. This means that this certificate was selected during configuration of the OPC UA server.

Note

One characteristic of *ibaPDA* is that certificates from different areas are managed in the certificate store: from the I/O Manager and from the data storage configuration.

For that matter the *Used By* column field has a link function. Via a double-click on a filled field you jump to the respective dialog.

If the entry belongs to the other manager, the link does not work. In the example above, a double-click on the “OPC UA-Server” entry would open the configuration dialog of the OPC UA server, provided it’s done in the I/O Manager. If you had opened the certificate store in the data storage configuration, the link to the OPC UA server does not work.

Vice versa, jumping to an “MQTT data store” would only work if the certificate store was opened in the data storage configuration.

The column *Expiration date* shows the date which marks the end of the validity of the certificate. Beyond this date the certificate cannot be used anymore. You have to renew the certificate or replace it by another, yet valid certificate. A red highlighted date indicates an expired certificate.

3.3.2.3 Manage certificates

The central certificate store is used to manage the certificates. Here you can add, create and delete certificates.

In the certificate store toolbar you will find a number of buttons with the following functions:

Button	Function
	This button opens a dialog that allows you to load an existing certificate file. Various file formats are supported (.der, .cer, .crt, .cert, .pem, .pfx, .p12). If you have a certificate with an unknown file extension, expand the file filter to "*.*" and try to open the file anyway. This works in most cases.
	This button opens a dialog that lets you create a new certificate file.
	This button lets you export a certificate to a file to register it for Windows or another application, e.g. on an OPC UA client. Multiple file formats are supported here as well.
	Use this button to remove the selected certificate from the table.
	Use this button to designate the selected certificate as "trusted".
	Use this button to designate the selected certificate as "not trusted". However, the certificate will still remain in the certificate store table. However, certificates that are not trusted are not available in the selection list for use in the corresponding configuration dialog.
	With this button you can define whether a certificate can also be used for user authentication for OPC UA.

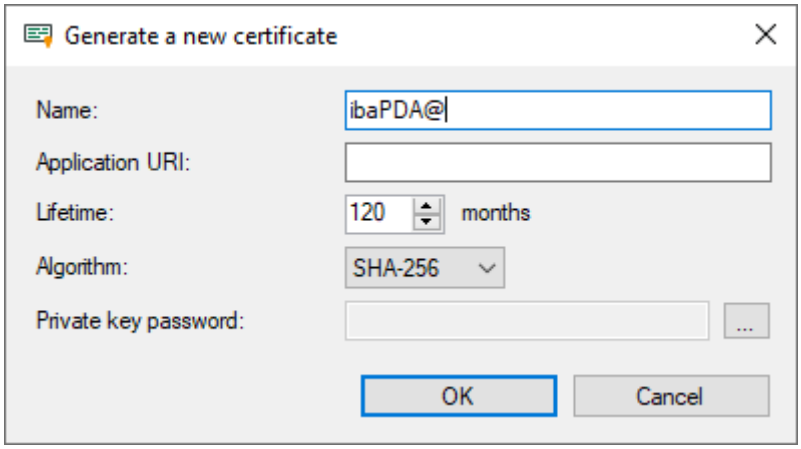
Table 3: Buttons in the toolbar for certificate management

The commands always refer to the certificate selected in the table, which is indicated by an arrow on the left at the start of the line.

3.3.2.3.1 Generate a new certificate

If no certificates are available to load, it is necessary to generate one.

1. Click the button and the following dialog box will open:



2. Enter a name of your choice for the certificate.
3. If required, enter an Application URI.
   The URI (Uniform Resource Identifier) is a global unique identifier for the application. If you do not fill in this field, a standard URI will be generated, provided that the OPC UA client verifies an Application URI. This standard URI consists of the machine name and the name of the application:
   `urn:machinename:applicationName.`
4. Define the desired validity period (lifetime) of the certificate.
5. Select the desired hash algorithm for the encryption.
   You have the choice between the algorithms SHA-256, SHA-384 and SHA-512.
   Make sure that the other communication partners support the selected algorithm too.
6. Define a password for the private key. If no password has been entered, the <OK> button remains inactive. To assign the password, click the <...> button and enter the password twice and confirm with <OK>. There are no special requirements for the password. Keep the password in a safe place so that the self-generated certificate can be exported and used for Windows or other applications.
7. Close the dialog with <OK>.

The new certificate is now entered into the list held by the certificate store and immediately assigned the properties "trusted" + private key.

You can now also export the certificate and register it with the communication partner, e.g., an OPC UA client. Afterwards, the client can then connect to *ibaPDA* (OPC UA-Server).

3.3.2.3.2 Add certificate

1. In the certificate store toolbar, click the button + .
   A dialog will open that lets you navigate to the desired certificate file and open it.
   Different file formats are supported (.der, .cer, .crt, .cert, .pem, .pfx, .p12).
   If you have a certificate with an unknown file extension, expand the file filter to "*.*" and try to open the file anyway. This works in most cases.
2. When the certificate is loaded, it appears in the certificate store list.
3. If you have not already done so, trust the certificate.

Certificates can sometimes be added without manual import.

Thus, during the first connection attempt by an OPC UA client to the OPC UA server (*ibaPDA*), the application certificate of the OPC UA client is automatically added to the certificate list and initially rejected.

Once you have selected the OPC UA client certificate in the list and confirmed it as trusted with the button the OPC UA client can subsequently connect automatically.

Use the button to reject a certificate at any time or to classify it as not trusted.

3.3.2.3.3 Export certificates

All certificates in the certificate store can be exported individually as a file and subsequently used for Windows or other applications. An exported certificate can also be re-imported into.

To export a certificate, first select the desired certificate in the table and then click the button in the toolbar for the certificate store.

If you wish to export a certificate without a private key, a dialog that lets you save the file opens immediately.

If the certificate to be exported has a private key, there are some options.

First, you will be asked if the existing private key should be exported as well. If you answer "no", the certificate will be saved immediately, just like a certificate without a key.

If you answer "yes", then you must enter the correct password afterwards. The correct password is the password used when importing or generating the certificate. If the password is correct, the certificate can be saved as a PFX-file. This file is password protected and contains the certificate and the private key.

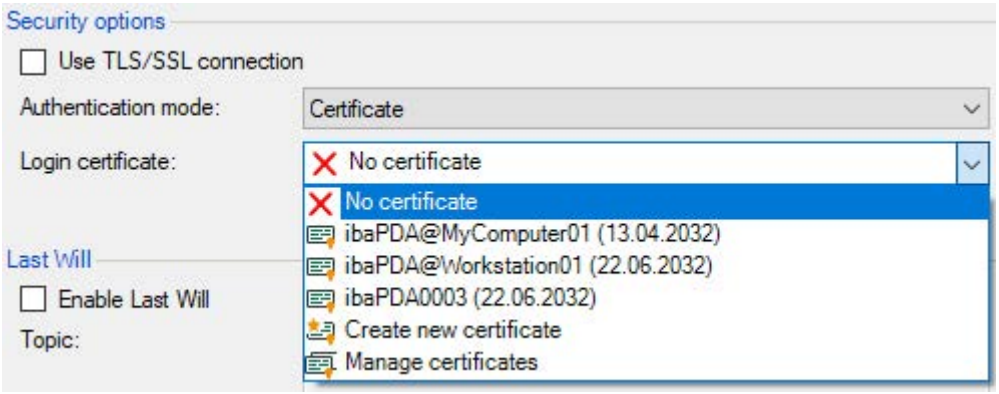
If the password is incorrect, the certificate will not be exported.

Under certain circumstances, a certificate with a private key may be stored, however the key is not password protected. In this case, the certificate can only be exported without a private key. You will then be notified accordingly.

3.3.2.4 Use certificates

At the points where certificates are applied, you will find a drop-down list offering the available certificates for selection.

There are the following options:



- No certificate: No certificate is used. As a rule, this leads to an invalid configuration.
- Available certificates: All certificates are displayed that are contained in the central certificate store, are valid, and are suitable for use at this point.
- Create a new certificate: The dialog for creation of a certificate opens. If the operation is successful, the new certificate is also selected immediately. If not, "No certificate" is selected.
- Manage certificates: Calling up the central certificate store.

Note

The selected certificate is saved in the registry file of the computer. In case of a new configuration, the same certificate will be selected unless another certificate is actively selected.

Other documentation

For more information on the use and functions of the certificates, please refer to the descriptions and manuals of the relevant applications, such as *ibaPDA*, div. *ibaPDA* interfaces, *ibaHD-Server*, *ibaDatCoodinator* etc.

3.3.2.5 Save and protect certificates

The certificates are stored in the `settings.xml` file, which is located in the program directory the respective application, in the `…\Certificates` subfolder. This file is automatically encrypted.

There are a number of measures whereby certificates with private keys can be used to protect your identity or that of your organization. Specifically, these are measures that make their simple export and reuse in Windows or other applications more difficult.

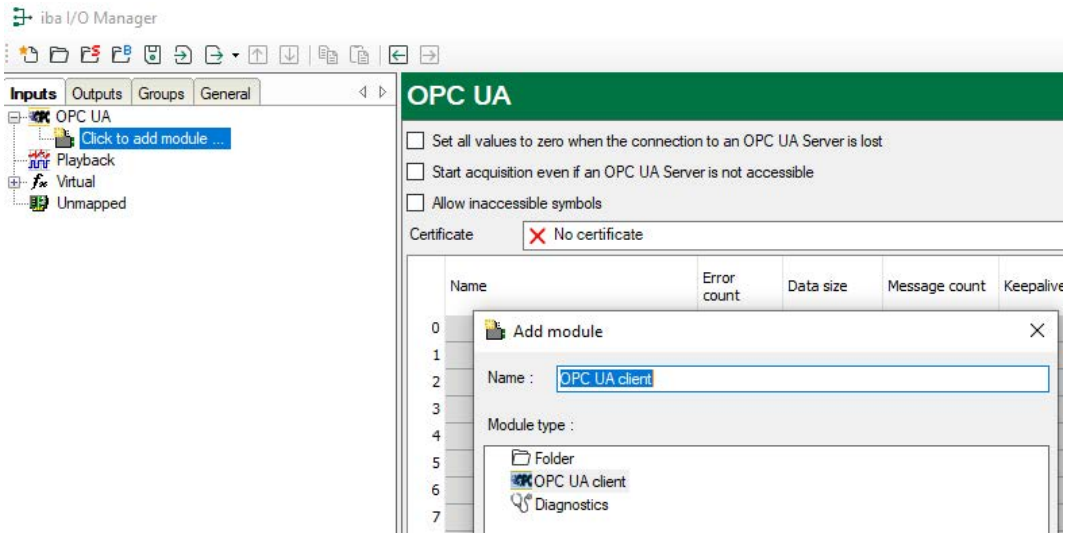
- Certificates are always stored in encrypted form.
- For certificates with a private key, the input of a password is required...
  - when a new certificate is generated
  - when a certificate with a private key is exported
  - when a certificate with a private key is imported
- Certificates with a private key can only be exported if there is also a password for the key. If there is no password or the password is unknown, the certificate can no longer be exported. Therefore, keep the passwords in a safe place.
- The password for a private key cannot be changed.
- It is not necessary to enter a password to use a certificate. The `settings.xml` file can be copied from one installation to another to transfer the certificates there. Password entry is not required for this either.

Should the private key fall into the wrong hands, many types of misuse are possible. Therefore, make sure that the passwords are kept safe.

3.3.3 Adding a module

Procedure

1. Click on the blue command *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>.



Module type

- OPC UA Client

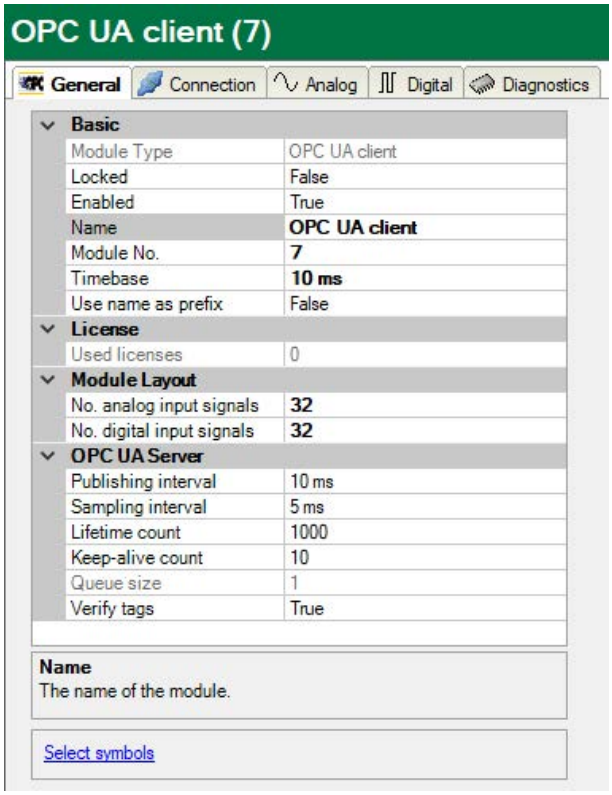
Note

If a license *ibaPDA-OPC UA-Server+* is available, then modules of the type OPC UA-Server can also be added. The following description refers to OPC UA client modules only. For information about OPC UA server modules, see *ibaPDA-OPC-UA-Server+* manual.

3.3.4 General module settings

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

All modules have the following setting options.



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.

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

Module layout

Number of analog and digital signals
Here, you can increase or decrease the number of signals in the module. By default, 32 signals are preset. You may enter any value between 0 and 1000. The signal tables will be adjusted accordingly.

OPC UA Server

Publishing interval
Minimum period of time the OPC UA server will wait before notifying *ibaPDA* about a change in values for one of the requested variables.

Default setting: 10 ms

Sampling interval
Time base for sampling the underlying data source in the OPC UA server. Basically, the sampling interval can equal the publishing interval.

However, we recommend setting the sampling interval no more than half the size of the publishing interval. This way, small delays within the OPC UA server can be compensated.

Default setting: 5 ms

Lifetime count
The amount of publishing intervals the connection between client and server can be interrupted before the server decides to delete the subscription.

Default setting: 1000

Keep-alive count
The amount of publishing intervals without new notifications before the server sends an empty message to let the client know the server is still alive.

Default setting: 10

Queue size
Maximum number of values stored for monitored items during one publishing interval.

Verify tags
When enabled *ibaPDA* will verify that the signals' properties (e.g. data type, write access) correspond to the current OPC UA tags state in the OPC UA server. It is recommended to enable this setting.

Link "Select symbols"

With this link, you open the Symbol Browser for selecting the variables to be measured. First, you have to establish the connection to the OPC UA server. Only then, the signals can be configured.

3.3.5 Module - Connections

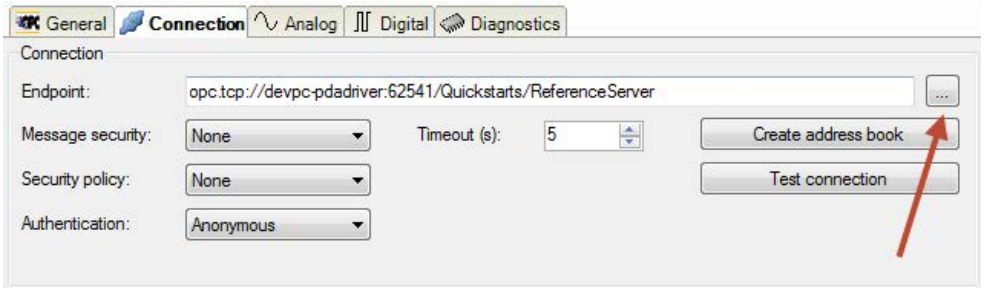
On the *Connection* tab of the module, you configure the connection to the OPC UA server. The communication connection between OPC UA Client and server is always being established to a so-called Endpoint on the OPC UA server side. Several endpoints can be defined within one OPC UA server.

An endpoint is defined by the following parameters:

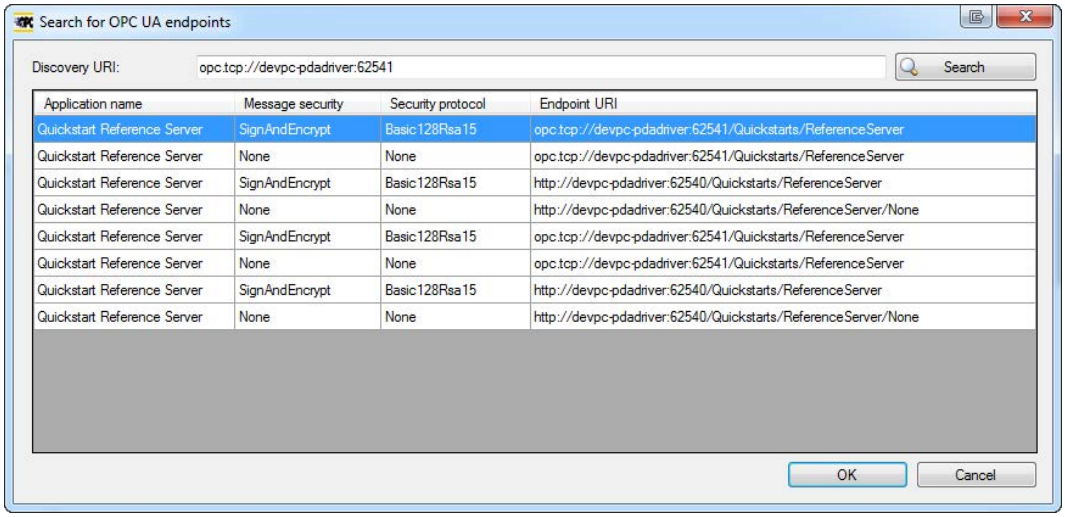
- URI of the endpoint
- Message Security
- Security Protocol
- User authentication

ibaPDA offers a convenient search function for available endpoints.

1. Click on the ... button to open the dialog for the endpoint search.



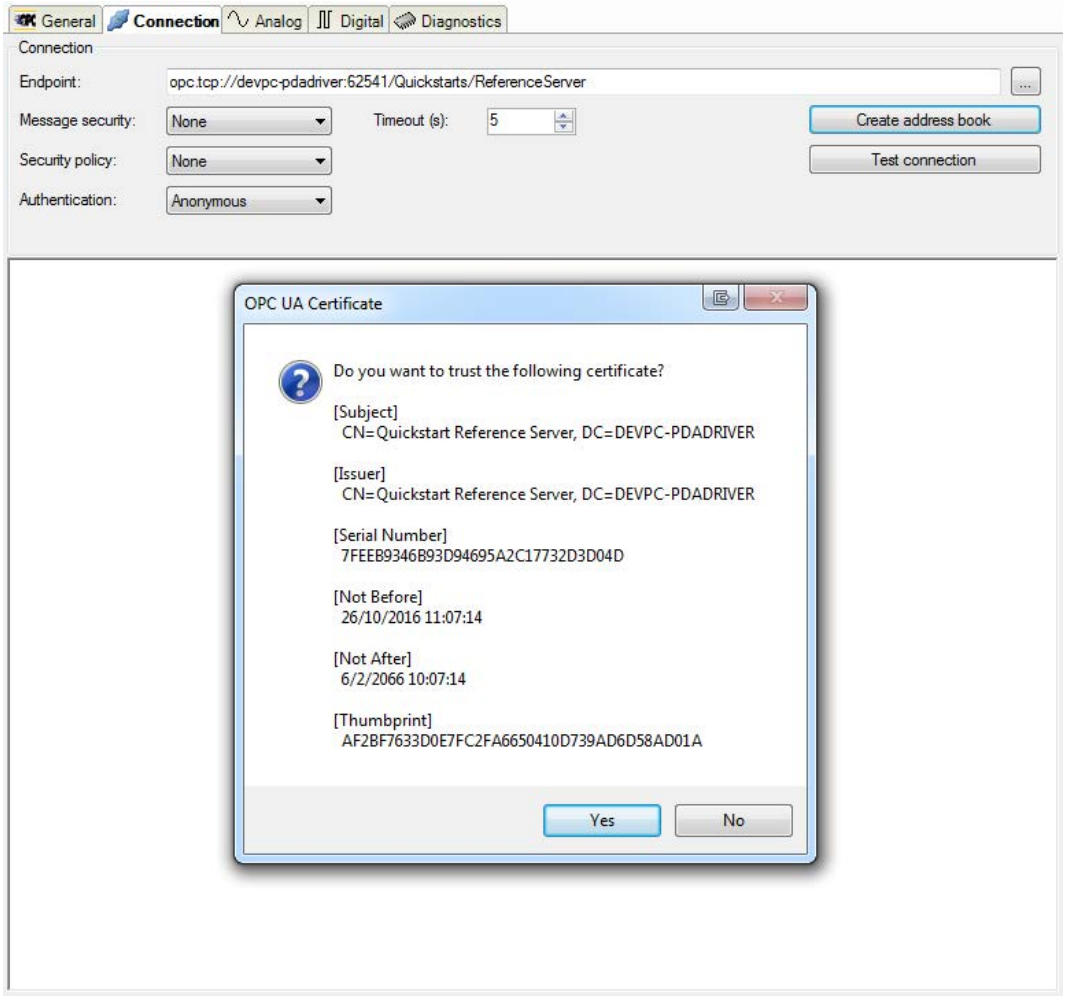
2. Enter in the “Discovery URI” field the so-called "Discovery” address of the OPC UA server and click on <Search>.
   Now, a list of the endpoints available on the OPC UA server will be displayed in the dialog.



3. Select the desired endpoint and close the dialog with <OK> or double-click on the endpoint.
   Now, the complete endpoint URI will be copied to the “Endpoint” field on the *Connection* tab.

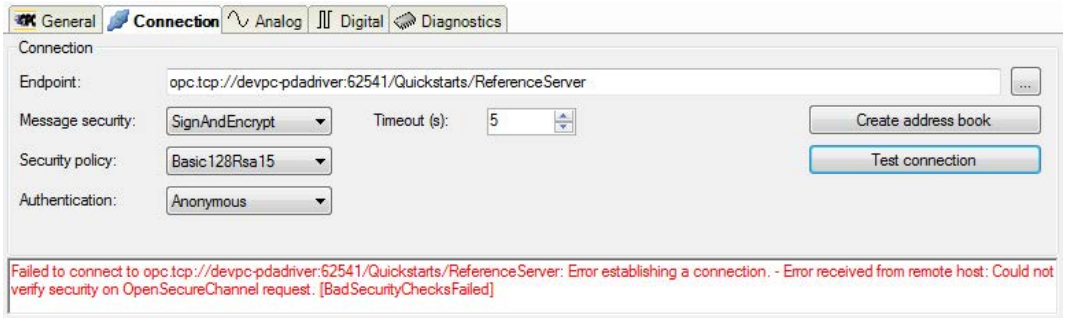
4. When selecting the endpoint, also the message security and security protocol are adopted. Please also synchronize the authentication with the configuration of the OPC UA server.
   ibaPDA supports the authentications “Anonymous” and “User/Password”.

5. Test the connection with a click on <Test connection>.
   When *ibaPDA* connects to the OPC UA server for the first time, the certificate is not “known” in the *ibaPDA* list of certificates. This is why you are asked, if you want to trust the certificate or not.
   Click on <Yes> or <No>.



In case you reject the certificate, no connection can be established. If you trust the certificate, the device will try to establish a connection.
When establishing a connection, the OPC UA server also checks the *ibaPDA* OPC UA Client certificate.

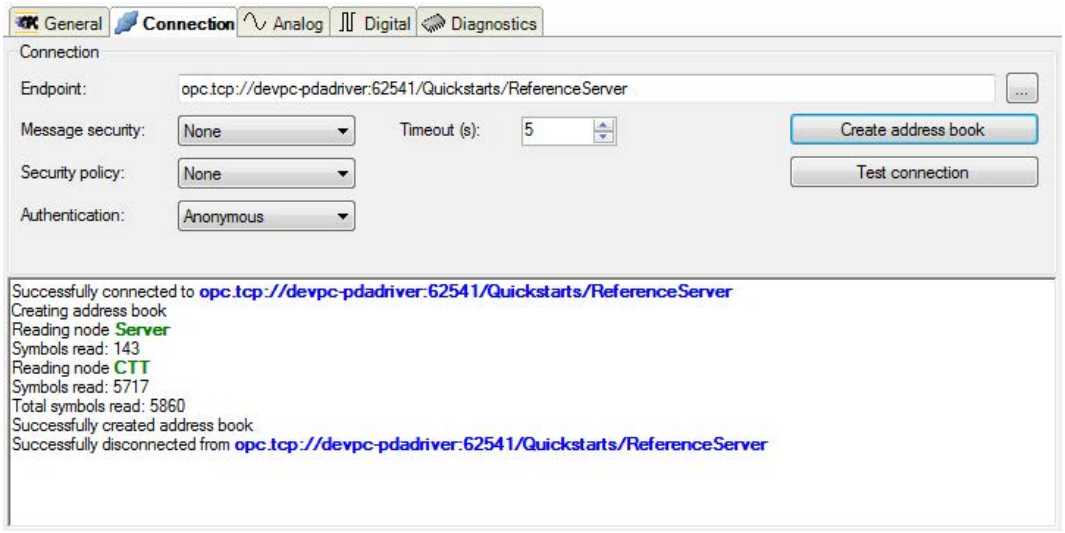
The error message shown in the next figure will be displayed, unless the *ibaPDA* OPC UA Client certificate is identified yet by the OPC UA server as trusted. Otherwise no security guidelines will be applied.



Occasionally, the *ibaPDA* OPC UA Client certificate is categorized automatically by the OPC UA server in the list of the rejected certificates. In that case, you have to mark the OPC UA Client certificate on the OPC UA server as trustworthy.

Then, there should not occur any problems, when establishing a connection.

If *ibaPDA* can establish a valid connection to the OPC UA server, you can see some characteristic values in the display area of the dialog, as shown in the following figure.



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

3.3.6 Signal configuration

The variables to be measured are configured in the *Analog* and *Digital* tabs.

The length of the signal tables, i.e. the number of signals per table, is specified in the general module settings, module layout (see ↗ *General module settings*, page 18).

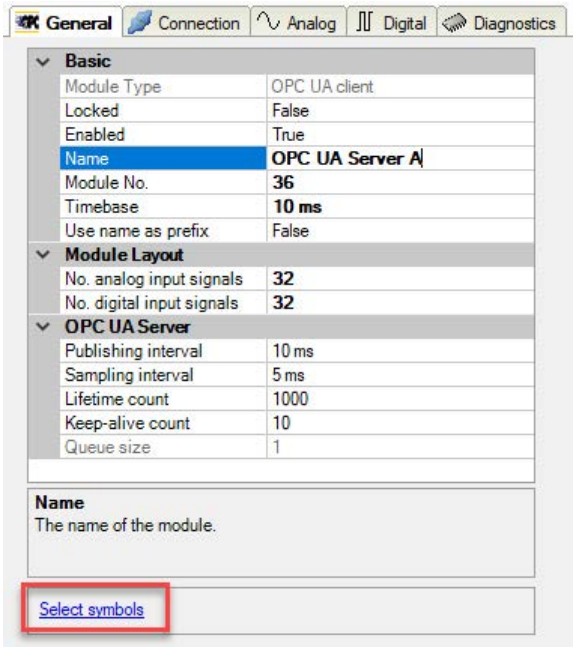
Note

Observe the maximum number of signals permitted by your license.

Selecting the signals to be measured by means of the Symbol Browser

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

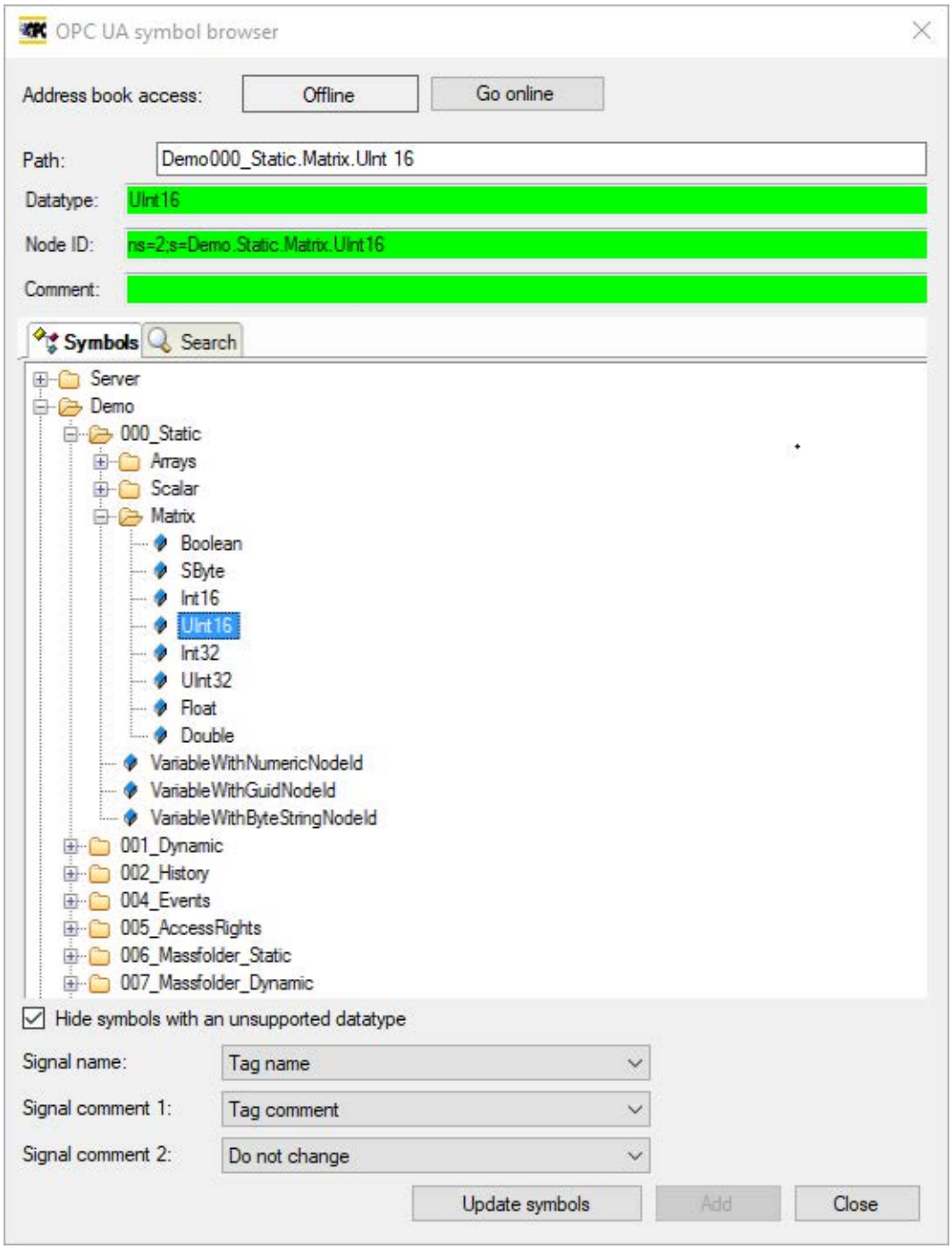
1. Clicking on the *Select symbols* hyperlink in the module's *General tab*.



A click on the link opens the OPC UA Symbol Browser.

2. Clicking on a field in the *Symbol* column of the *Analog* or *Digital* tab.

The icon is shown. A click on the icon opens the OPC UA Symbol Browser.



The Symbol Browser shows all the symbols that were loaded from the OPC UA server. You can select single or multiple symbols in the tree.

Click the <Add> button to add them to the corresponding analog or digital signal table. If you selected a single symbol then the next symbol will be selected after you clicked the <Add> button. This allows you to click <Add> multiple times in order to add consecutive symbols. You can also double click a symbol to add it to the signal table. Use the <Update symbols> button to read the symbols again from the OPC UA server.

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

Note

You can hide all unsupported datatypes by checking the "Hide symbols with an unsupported datatype" checkbox.

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

Signal Table

Name
In the Name column, enter a clear text name for each signal.

When selecting the signals in the Symbol Browser, the symbol names are applied automatically, provided that there is no entry in the *Name* column.

Up to two lines of comment may be entered for each signal in the column *Name.*

You can access the comments by clicking on the small button  in the Name field of the corresponding signal.

Tip

A useful feature when completing the name fields is the automatic fill function. If you enter a signal name and double-click on the column header as long as the cursor is still in the name field then all empty fields below will be filled automatically with that name. If the name is ending with a number you will get names with an incrementing number per line. You may use this function in any row of the table. Fields which already contain names will not be overwritten.

Unit
Assignment of an engineering unit (such as °C, Ampere, Volt, N etc.) for the signal.

Gain and Offset
The values for gain and offset describe the inclination and position of a linear characteristic curve for scaling.

Gain and offset can be entered directly in the corresponding fields or by means of the two-point-scaling dialog with two pairs of applicable values.

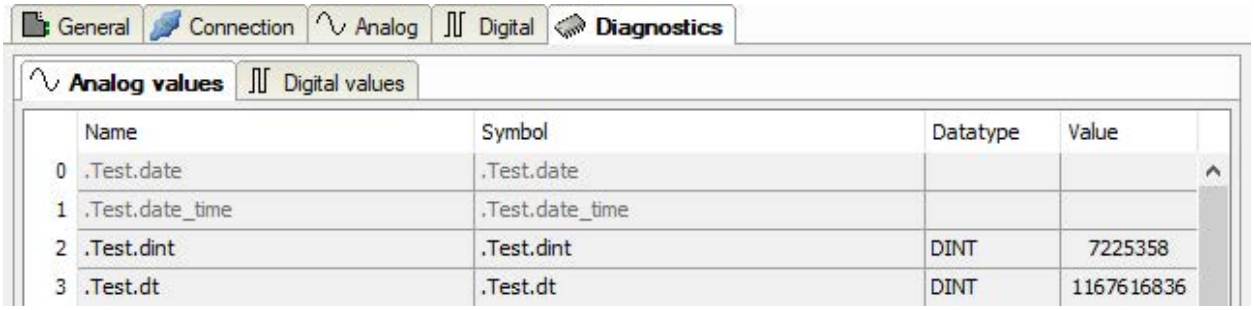
You can open the two-point-scaling dialog with a click in the fields gain or offset and then on the little button  .

Symbol
Here, you can enter the symbol name or open the Symbol Browser.

Activating the channels
You can enable and disable every channel for acquisition just with a mouse-click.

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.



Inactive signals are grayed out.

3.3.8 Output modules

The interface OPC UA Client provides the capability to send data from the *ibaPDA* system to a PLC or OPC UA Server respectively. Therefore, the following output modules are available:

- OPC UA Client
  This is not a discrete module but rather the output extension of an OPC UA Client module

3.3.8.1 OPC UA Client module

Modules, which have been added on the input side are available in the output interface tree too and can be used for output signals.

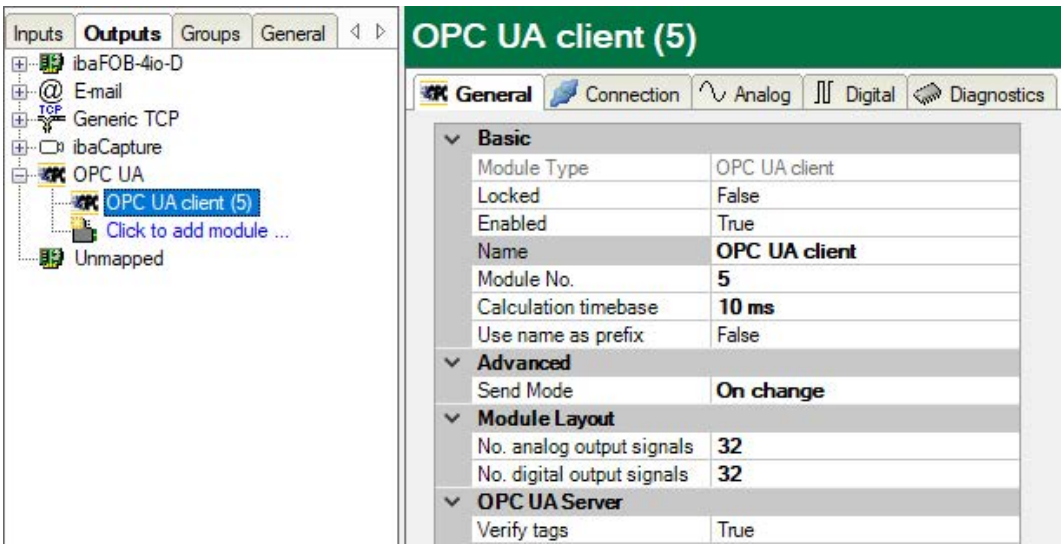
If you want to configure an output module, mark the module in the tree structure of the *Outputs* tab.

If you want to use output signals, make sure that appropriate tags have been configured on the corresponding OPC UA server.

As for all output modules, the time base of the output cycle is at least 50 ms.

General module settings

The general module settings are partly the same as on the input side.



On the output module, no separate settings can be made for the OPC UA server.

Calculation timebase

The calculation timebase determines the refresh interval of the values and corresponds to the timebase of the module. The transmission of values to the server occurs in "minimum output time" (see I/O-Manager - node *General* - tab *Timing*).

The number of analog and digital output signals is set to 32 by default. You can change the number if needed (max. 1000).

With a click on the blue hyperlink, you can open the OPC UA symbol browser. Provided, you have created the address book of the connected OPC UA server before you can now see all

available tags. Double-click on a tag or select a tag and click on <Add> in order to add the tag in the next free row of the appropriate signal table (*Analog* or *Digital*). The tag will be entered in the *Node ID* column.

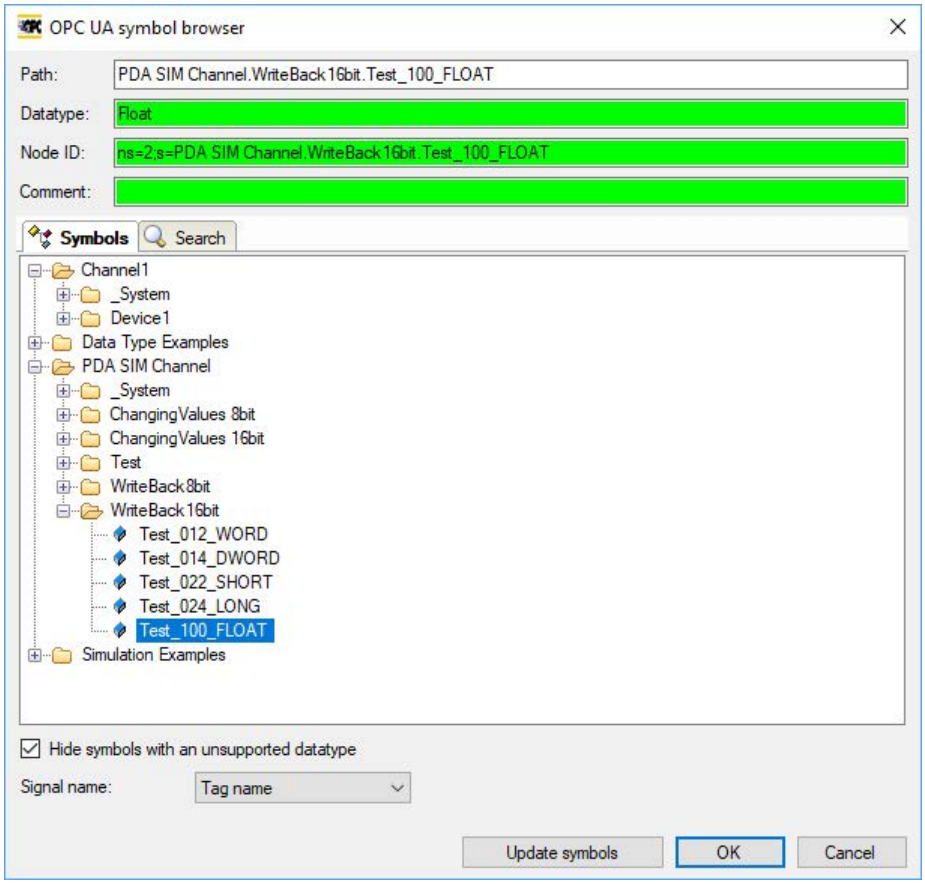
Connection

The connection settings are the same as on the input side (OPC UA Client module, I/O-Manager, section *Hardware*).

Analog and digital signals

On the tabs *Analog* and *Digital*, in the column *Value* you can either select signals from the signal tree and publish them as OPC UA output signals or create your own (virtual) signals by means of the expression builder.

The assignment of the signals and the OPC UA tags of the OPC UA server can be done in the column *Node ID*. With a click on the button <...> in a table cell, you can open the OPC UA symbol browser and select the desired tag.



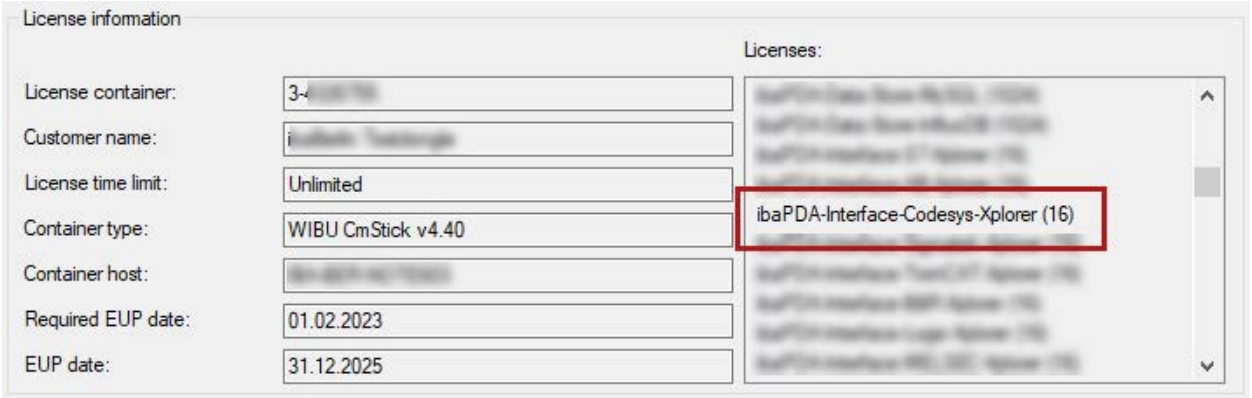
If you open the symbol browser out of the signal tables instead of clicking on the hyperlink on the *General* tab, then the tags with matching data types are presented.

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 this interface 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.



4.2 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.

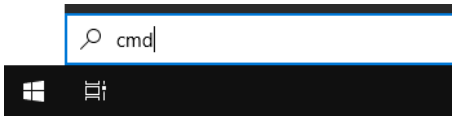
Examples:

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

4.3 Connection diagnostics with PING

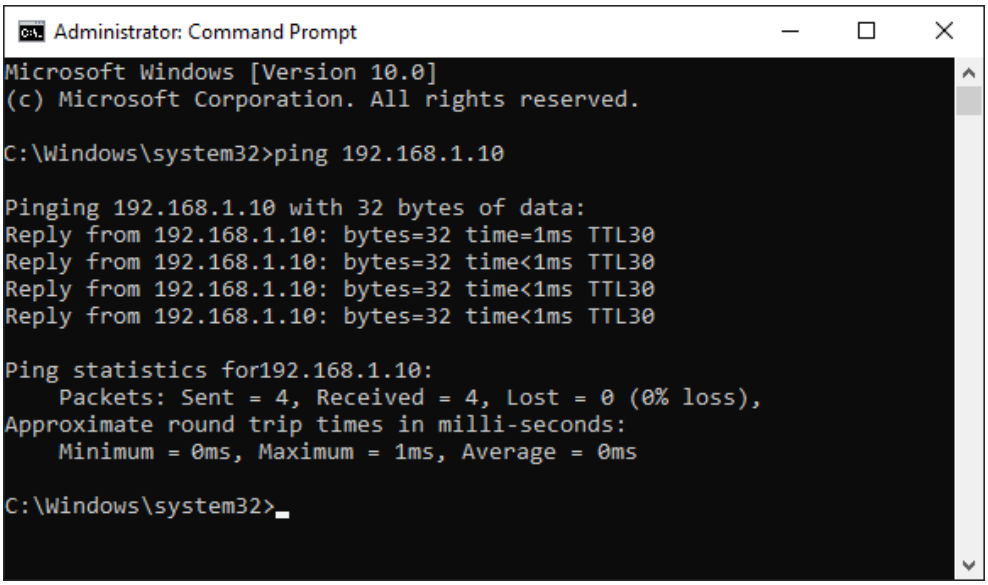
PING is a system command with which you can check if a certain communication partner can be reached in an IP network.

1. Open a Windows command prompt.

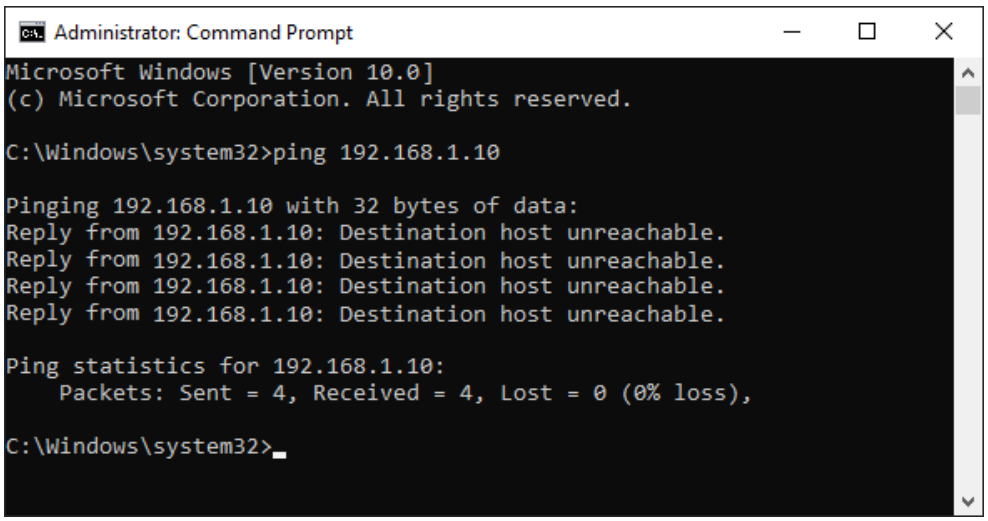


2. Enter the command "ping" followed by the IP address of the communication partner and press <ENTER>.

→ With an existing connection you receive several replies.

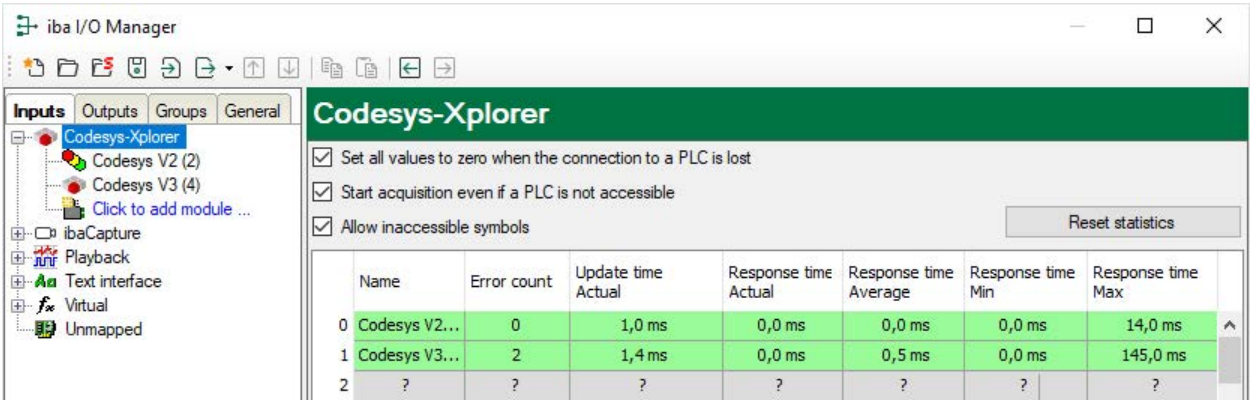


→ With no existing connection you receive error messages.



4.4 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:



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.

4.5 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.

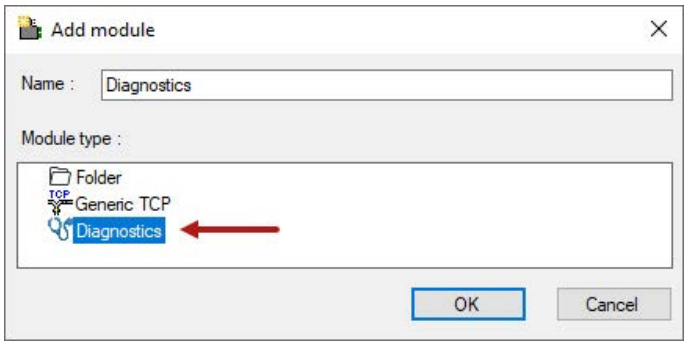
A diagnostic module is always assigned to a data acquisition module of the same interface and supplies its connection information. By using a diagnostic module you can record and analyze the diagnostic information continuously in the *ibaPDA* system.

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

Example for the use of diagnostic modules:

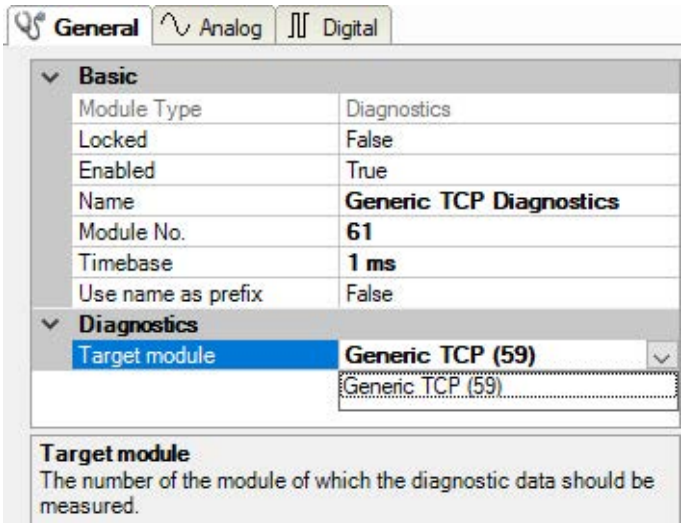
- A notification can be generated, whenever the error counter of a communication connection exceeds a certain value or the connection gets lost.
- In case of a disturbance, the current response times in the telegram traffic may be documented in an incident report.
- The connection status can be visualized 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).



Module settings diagnostic module

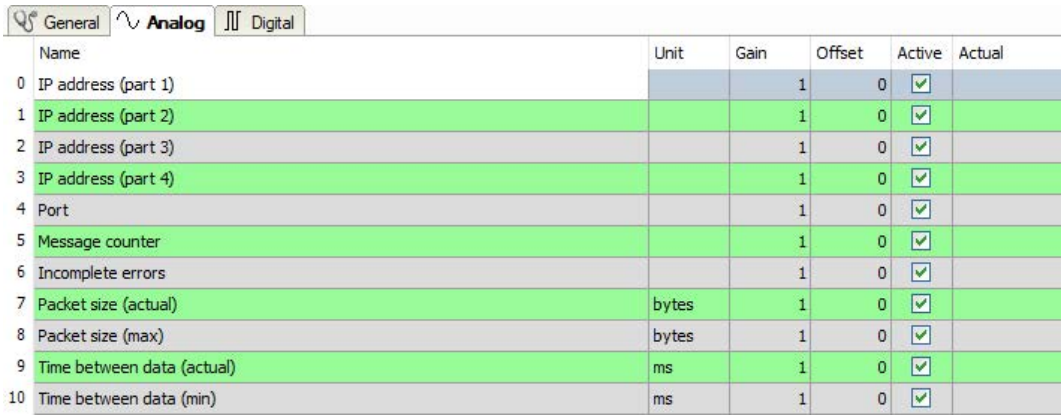
For a diagnostic module, you can make the following settings (example: Generic TCP):



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

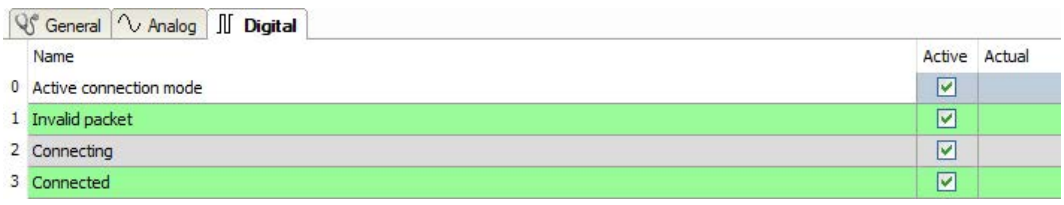
There is only one setting which is specific for the diagnostic module: the target module.

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 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

Fax: +49 911 97282-33

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

Fax: +49 911 97282-33

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.