



# ibaPDA-Request-DTBox

Request Data Interface for DTBox Systems

Manual  
Issue 1.0

Measurement Systems for Industry and Energy  
[www.iba-ag.com](http://www.iba-ag.com)

---

## Manufacturer

iba AG  
Koenigswarterstr. 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](mailto:iba@iba-ag.com)  
Web [www.iba-ag.com](http://www.iba-ag.com)

Unless explicitly stated to the contrary, it is not permitted to pass on or copy this document, nor to make use of its contents or disclose its contents. Infringements are liable for compensation.

© iba AG 2020, All rights reserved.

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](http://www.iba-ag.com).

Version	Date	Revision - Chapter / Page	Author	Version SW
1.0	04/2020	First issue	St	7.0.0

Windows® is a brand and registered trademark of Microsoft Corporation. Other product and company names mentioned in this manual can be labels or registered trademarks of the corresponding owners.

## Content

<b>1</b>	<b>About this Manual .....</b>	<b>4</b>
1.1	Target group and previous knowledge .....	4
1.2	Notations .....	5
1.3	Used symbols.....	6
<b>2</b>	<b>System requirements .....</b>	<b>7</b>
<b>3</b>	<b>Introduction .....</b>	<b>8</b>
3.1	General information .....	8
3.2	Failover Management.....	9
3.3	System integration via data path UDP .....	10
3.4	System integration via data path Reflective Memory.....	10
<b>4</b>	<b>Configuration in ibaPDA .....</b>	<b>11</b>
4.1	Setting up the connection.....	11
4.2	DTBox Request module .....	13
4.2.1	Add module .....	13
4.2.2	General settings.....	14
4.3	Address books .....	16
4.4	Signal configuration .....	19
4.5	UDP data path configuration .....	20
4.6	Data modules.....	20
<b>5</b>	<b>Diagnosis .....</b>	<b>23</b>
5.1	License .....	23
5.2	Connection diagnostics.....	23
5.3	Log files.....	24
5.4	Connection diagnostics with PING.....	25
<b>6</b>	<b>Appendix .....</b>	<b>26</b>
6.1	Architecture.....	26
<b>7</b>	<b>Support and contact.....</b>	<b>28</b>

# 1 About this Manual

This document describes the function and application of the software interface *ibaPDA-Request-DTBox*.

The product *ibaPDA-Request-DTBox* is an extension of *ibaPDA* for the optional access to variables when recording data from Dualis Technologies DTBox controllers. This manual only shows the extensions and differences. Refer to the manual from *ibaPDA* for all other functions and operating options. DTBox Controllers are used by Nidec ASI, France, too.

## 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 as a professional if he/she is capable of assessing the work assigned to him/her and recognizing possible 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 of *ibaPDA-Request-DTBox* the following basic knowledge is required and/or useful

- Windows operating system
- Basic knowledge of *ibaPDA*
- Basic knowledge of network technology
- 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 <i>Logic diagram</i>
Calling the menu command	<i>Step 1 – Step 2 – Step 3 – Step x</i> Example: Select the menu <i>Logic diagram - Add - New function block</i> .
Keys	<Key name> Example: <Alt>; <F1>
Press the keys simultaneously	<Key name> + <Key name> Example: <Alt> + <Ctrl>
Buttons	<Key name> Example: <OK>; <Cancel>
File names, paths	"Filename", "Path" Example: "Test.doc"

## 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

- *ibaPDA* v7.0.0 or higher
- License for *ibaPDA-Request-DTBox*
- If Reflective Memory is used as a data path:
  - Additional license *ibaPDA-Interface-Reflective-Memory*
  - Fiber optic board of the Abaco's 5565 Reflective Memory family (such as the PCI-e-5565PIORC) in the *ibaPDA* computer
- DTBox controller
- Ethernet connection to the controller
- DT v4.2.0 or higher (request blocks already installed)

### Licenses

Order no.	Product name	Description
31.001380	ibaPDA-Request-DTBox-128	Extension license for an ibaPDA system to be able to use the request functionality with Dualis DTBox, up to 128 signals
31.001381	ibaPDA-Request-DTBox-1024	Extension license for an ibaPDA system to be able to use the request functionality with Dualis DTBox, up to 1024 signals
31.001382	ibaPDA-Request-DTBox-unlimited	Extension license for an ibaPDA system to be able to use the request functionality with Dualis DTBox, unlimited number of signals
31.001385	upgrade-ibaPDA-Request-DTBox-128 to 1024	Increase number of signals
31.001386	upgrade-ibaPDA-Request-DTBox-1024 to unlimited	Increase number of signals

Table 1: Available licenses

### Hardware

Order no.	Product name	Description
19.114003	RTNET-PCIE-5565PIORC	PC plug-in board for Reflective Memory

Table 2: Hardware

The Reflective Memory board as well as other modules for the Reflective Memory communication are manufactured and sold by Abaco Systems.

## 3 Introduction

### 3.1 General information

The interface *ibaPDA-Request-DTBox* is suitable for the measuring data acquisition with a free variable selection from a DTBox via Ethernet (UDP) or Reflective Memory. In *ibaPDA*, the variables to be measured are selected with a browser. A request block is included in DTBoxes v4.2.0 or higher.

A DTBox can contain up to 100 resources which contain global variables and/or a collection of programs which can in turn contain local variables. Each resource has 4 instances of the request agent running where each instance corresponds to a different time class (to subsample the DTBox's cycle time). The value of this time class can be configured in *ibaPDA*. See also chapter [➤ Architecture, page 26](#)

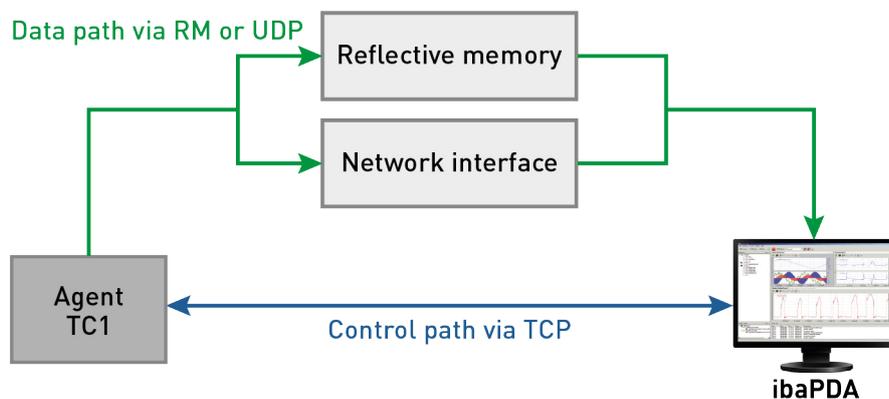


Fig. 1: Communication ibaPDA - DTBox controller

There is a control path and a data path between *ibaPDA* and an agent instance. The control path is a TCP/IP connection through which *ibaPDA* sends request messages containing a list of the desired variables and also periodically requests a watchdog/status message from the agent.

Once the variables are requested the actual data is sent through the data path, which can be either UDP or Reflective Memory. The data path needs to be configured in the DTBox Request agent and cannot be selected in *ibaPDA*.

Switching between Reflective Memory and UDP is configured in file *ibaSettings.JSON* (by Dualis Technologies or Nidec ASI), see DTBox documentation.

## 3.2 Failover Management

A failover solution is feasible with one *ibaPDA* server connected to two DTBoxes.

The DTBox system embeds the ISaGRAF failover mechanism. The *ibaPDA-DTBox* driver must manage this feature in order to have *ibaPDA* working when this feature is enabled.

Failover is a backup operational mode in which the functions of an industrial system are assumed by secondary industrial system components when the primary component becomes unavailable through either failure or scheduled down time. Used to make systems more fault-tolerant, failover is typically an integral part of mission-critical systems that must be constantly available.

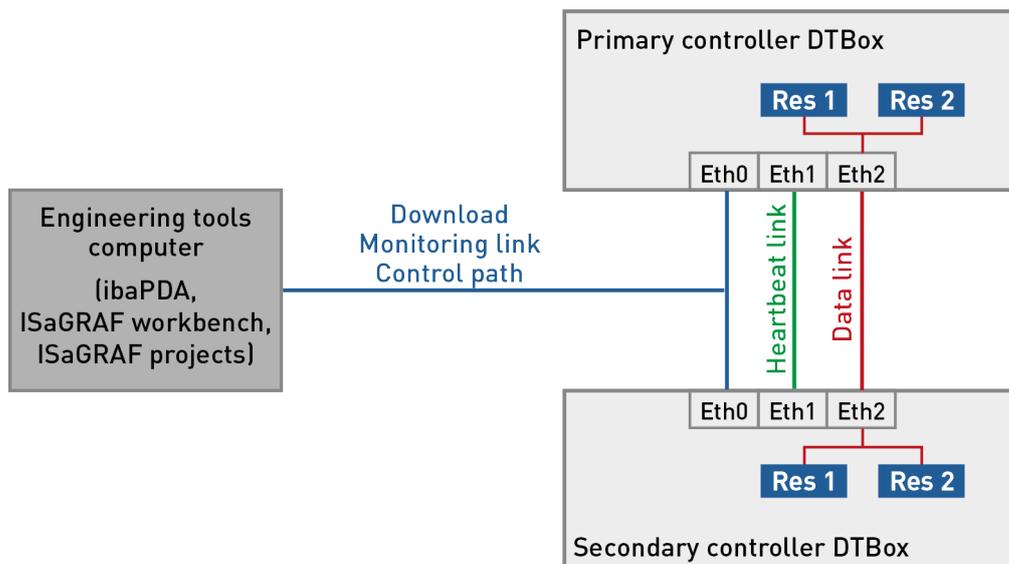


Fig. 2: Failover mechanism

### Failover mechanism

- In case of failure on the active controller, the standby controller becomes the active one and controls the process.
- Each DTBox has a different IP address (IP address of each system is available in the device definition file)

The *ibaPDA* server is informed via the Failover status in the watchdog response message whether the failover mechanism is enabled in the connected DTBoxes. Each DTBox itself also reports whether it is active or in standby.

In case of failover architecture the configuration is loaded inside both DTBoxes, but only the active DTBox sends data. When the DTBox becomes inactive, the FailoverStatus changes to inactive state and stops sending data. The new active DTBox starts to send immediately data and updates his FailoverStatus to active state.

### 3.3 System integration via data path UDP

The measurement data is transmitted via UDP to *ibaPDA*. You need an Ethernet connection via standard network cards.

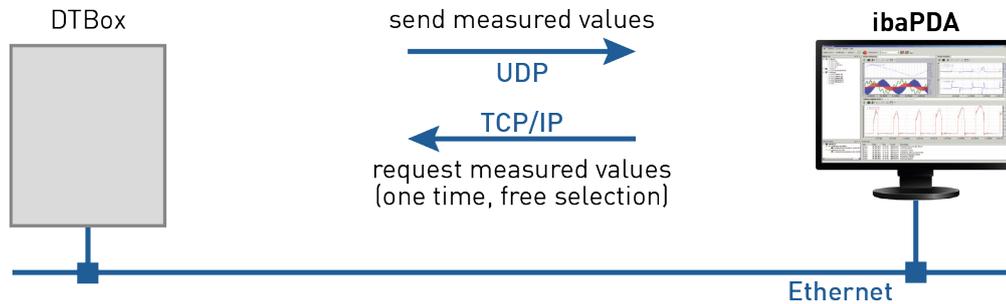


Fig. 3: System integration via data path UDP

*ibaPDA* tells the agent to which UDP port it should send data. The IP address of *ibaPDA* can be taken from the control TCP connection.

### 3.4 System integration via data path Reflective Memory

The measurement data is transmitted via Reflective Memory either directly to *ibaPDA* or via a Reflective Memory hub.

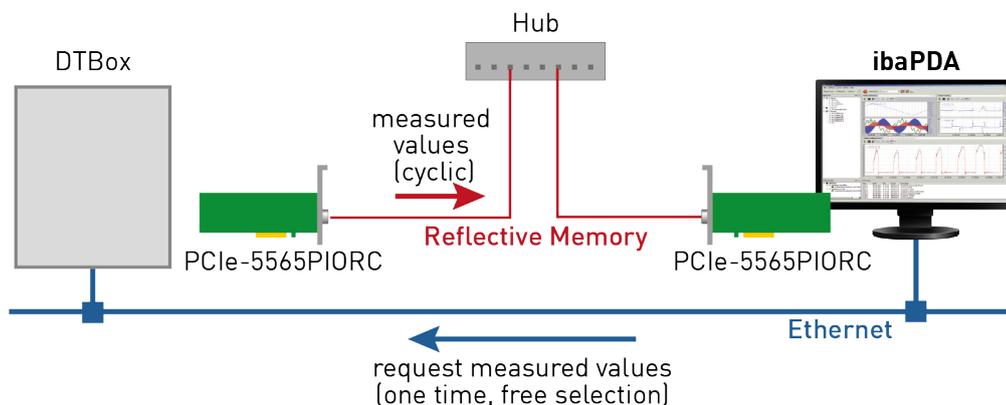


Fig. 4: Connection with data path via Reflective Memory

You need the following connections:

- Ethernet connection between *ibaPDA* and the DTBox
- Fiber optic link via Reflective Memory board in the *ibaPDA* computer and in the DTBox controller (e.g. PCIe-5565PIORC, sales: Abaco Systems)

There is a Reflective Memory board in the DTBox that is shared by all agents. Each agent should get its own memory region on the Reflective Memory. In the *ibaPDA* PC there is also a Reflective Memory board. The agent tells *ibaPDA* the memory region it has access to via the watchdog message.

## 4 Configuration in ibaPDA

The configuration of the DTBox interface is done in the I/O Manager of *ibaPDA*. First set up the connection from *ibaPDA* to the DTBox controller via Ethernet and configure the DTBox Request interface.

Once the connection is set up, add a DTBox Request module. See chapter [↗ DTBox Request module](#), page 13.

The configuration of the signals and the selection in the symbol browser is described in chapter [↗ Address books](#), page 16.

### 4.1 Setting up the connection

When a DTBox Request interface license is available in *ibaPDA* two new interface nodes are displayed in the I/O Manager: the *DTBox Request* interface and the *DTBox Request UDP* interface.

The *DTBox Request* interface corresponds to the control path, where the actual variables are selected that will be requested from the DTBox Request agent(s). The interface has the following functions and configuration options in the *Configuration* tab:

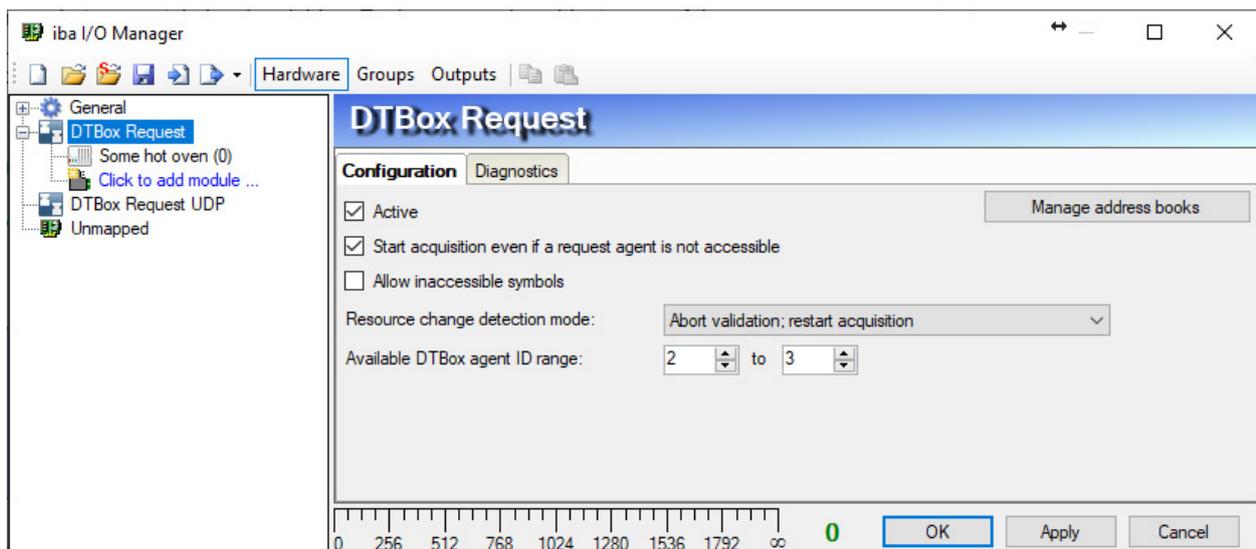


Fig. 5: Configuration tab

#### Active

Indicates whether the request interface is active. The interface is automatically activated when adding a first *DTBox Request* module.

#### Start acquisition even if a request agent is not accessible

When this option is enabled and *ibaPDA* is unable to connect to a request agent during the start of the acquisition the validation process will not abort but be able to continue. During the acquisition, *ibaPDA* will periodically try to connect to the unavailable request agents. Once the connection to one or more request agents can be established the acquisition is automatically restarted.

**Allow inaccessible symbols**

When this option is enabled and *ibaPDA* receives an error code for one or more variables from the DTBox request agent it will automatically disable those variables and restart the acquisition.

**Resource change detection mode**

Each DTBox resource has a checksum. Through means of the watchdog messages, a DTBox request agent periodically transmits the current checksum of the DTBox resource it is running on. Apart from that, the checksum at the time the DTBox solution was compiled, exported and imported in *ibaPDA* is also stored in the address book. At the start of the acquisition *ibaPDA* will check whether the current checksum value (as received from the watchdog messages) corresponds to the value stored in the address book. During the acquisition, *ibaPDA* will keep on checking for a change in checksum in consequent watchdog messages. The *Resource change detection mode* specifies what *ibaPDA* will do when a change is detected:

- *Log warning* (*ibaPDA* reports a warning and continues)
- Abort validation; restart acquisition
- Reload address book during validation; restart acquisition (*ibaPDA* tries to re-import the address book provided that the original ISaGRAF files are still available)

When one of the last two options is selected *ibaPDA* will resend the request messages to the DTBox Request agent to check if the change in checksum has any repercussions on the requested variables. If an error occurs the entire acquisition is restarted.

**Available DTBox agent ID range**

Each resource contains 4 DTBox request agents with IDs 0 to 3. When multiple *ibaPDA* systems need to access the same DTBox resource normally both systems would try to configure agents starting with ID 0. Since this would result in a conflict it is possible to define separate DTBox agent ID ranges for both systems.

For example: system A uses DTBox agents 0 to 1 and system B uses DTBox agents 2 to 3. Both systems will now each have two agents at their disposal without coming into conflict with one another. Note that this settings applies to each resource in the DTBox.

**<Manage address books>**

Click this button to create and manage address books for convenient symbol selection. See chapter [➤ Address books](#), page 16.

## 4.2 DTBox Request module

### 4.2.1 Add module

Add a *DTBox Request* module in the I/O Manager by clicking below the *DTBox Request* interface. Enter a unique name and click <OK>.

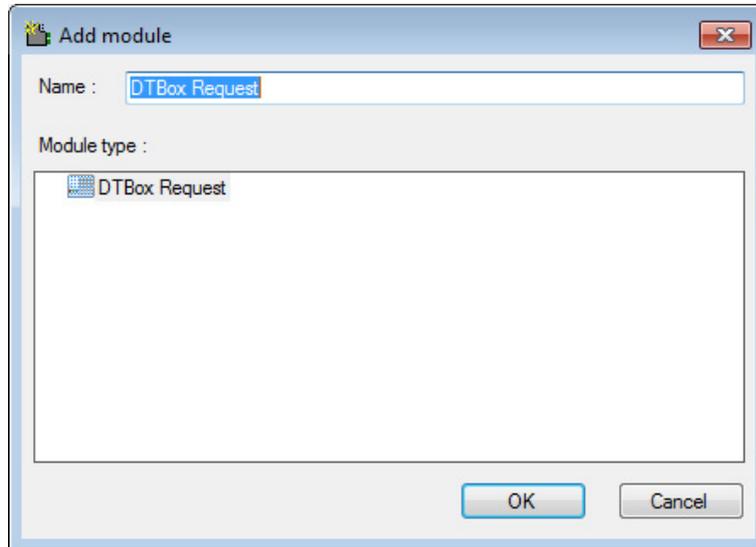


Fig. 6: Add module dialog

## 4.2.2 General settings

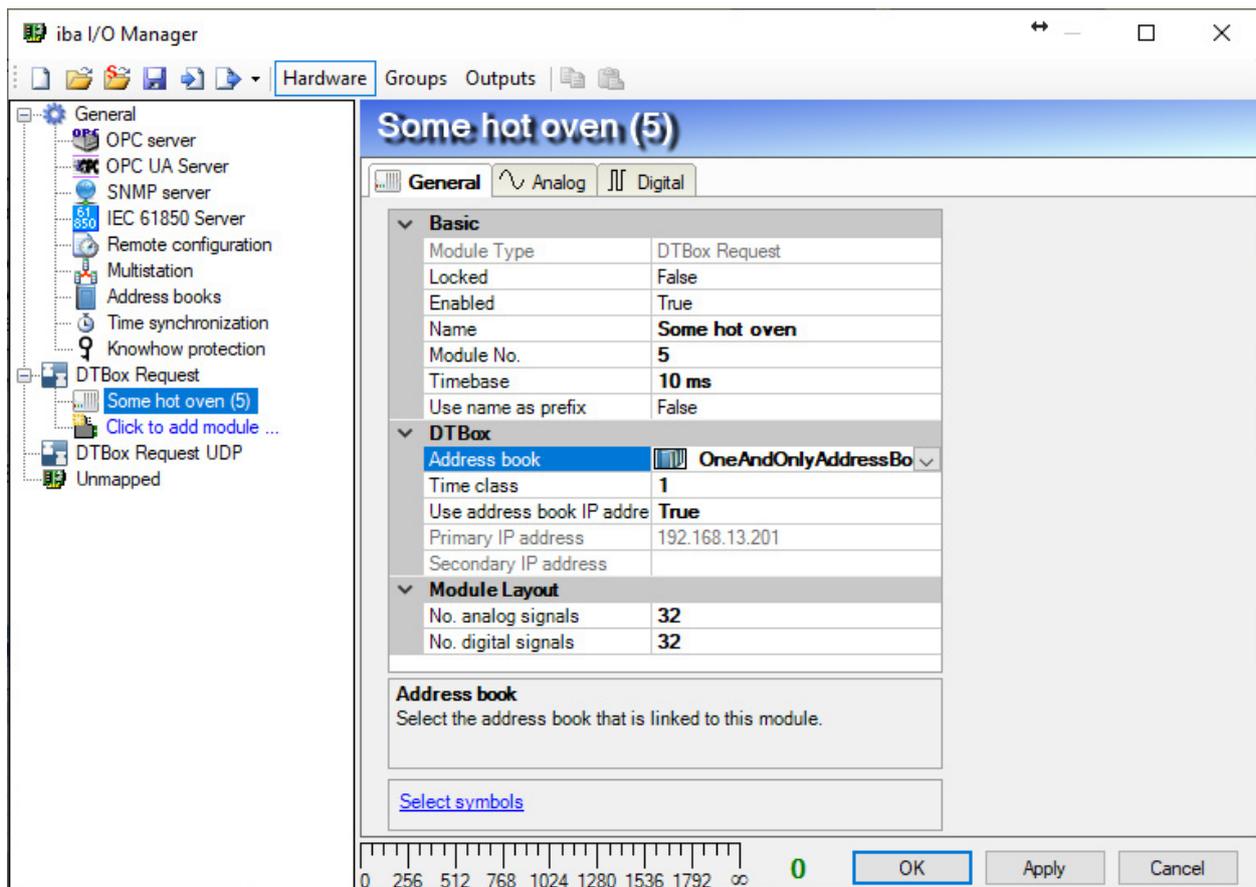


Fig. 7: General tab

### Basic settings

#### Module Type (information only)

Indicates the type of the current module.

#### Locked

A module can be locked to avoid unintentional or unauthorized changing of the module settings.

#### Enabled

Disabled modules are excluded from signal acquisition.

#### Name

The plain text name should be entered here as the module designation.

#### Module No.

Internal reference number of the module. This number determines the order of the modules in the signal tree of *ibaPDA* client and *ibaAnalyzer*.

#### Time base

All signals of the module will be sampled on this time base.

**Use name as prefix**

Puts the module name in front of the signal names.

**DTBox****Address book**

Indicates which address book (corresponding to an ISaGRAF solution) will be used to select variables. In order to create an address book, click the drop-down menu and select *Create address book...* or go to the *General* node, select the *Address books* node and select the *DTBox* tab (see also chapter ↗ *Address books*, page 16).

**Time class**

Defines how fast data is sent from a DTBox Request agent to *ibaPDA*. This is expressed in a number of DTBox cycle times.

**Use address book IP addresses**

- True: The DTBox IP addresses will be extracted from the address book.
- False: Custom IP addresses can be supplied (in the next two fields).

**Primary IP address**

The IP address of the DTBox (or primary DTBox in case of a failover scenario). Only when *Use address book IP addresses* is set to *False* the IP address can be edited.

**Secondary IP address**

The IP address of the secondary DTBox in case of a failover scenario. Only when *Use address book IP addresses* is set to *False* the IP address can be edited. When only one DTBox device is used (in case of no failover scenario) this can be left blank.

---

**Note.**

A DTBox Request module can contain variables from different resources of the same DTBox device. In ISaGRAF solutions with more than 1 device, only the first device can be used.

---

**Module Layout****No. analog/digital signals**

Determination of the number of analog/digital signals for this module (max. 1000)

## 4.3 Address books

The address books for DTBox controllers are created and managed across modules. One address book can be used in more than one module. There are several ways to open the address books dialog:

- Mark the *Address books* node under the *General* node in the *I/O Manager* and select the *DTBox* tab or
- Mark the *DTBox Request* interface and click the button <Manage address books>.

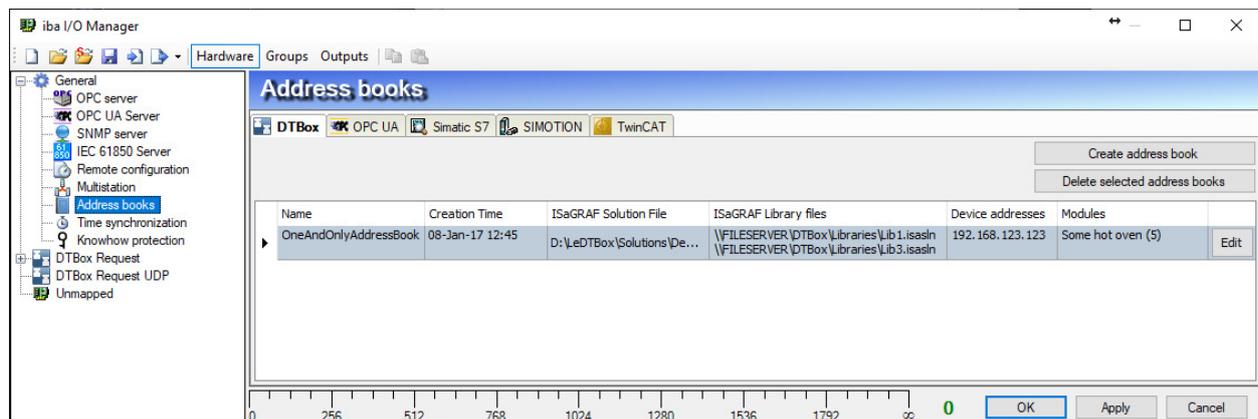


Fig. 8: Address books for DTBox

If there are already address books available you will see an overview of the address books in this view. Otherwise you have to create an address book using the button <Create address book>, see below. Existing address books can be deleted by clicking the <Delete selected address books> button.

An address book is referenced in *ibaPDA* through its unique *Name*. The following additional properties are shown in the address book grid:

### Creation Time

The *ibaPDA* client time at which the address book was generated in *ibaPDA*.

### ISaGRAF Solution file

The original path of the ISaGRAF solution. When clicking *Update symbols* in the symbol browser (see below) and the original path is still available, *ibaPDA* will update the address book.

### ISaGRAF Library files

The original paths of the ISaGRAF solutions that are included as libraries.

### Device addresses

A list of the IP addresses of the DTBox devices contained in the solution

### Modules

A list of the DTBox Request modules currently using the address book

### Edit

The properties of an existing address book can be edited by clicking this button. After you've finished editing the properties the address book will be recreated.

## Creating an address book

When clicking the <Create address book> button a dialog appears prompting you to enter the parameters to import the ISaGRAF solution.

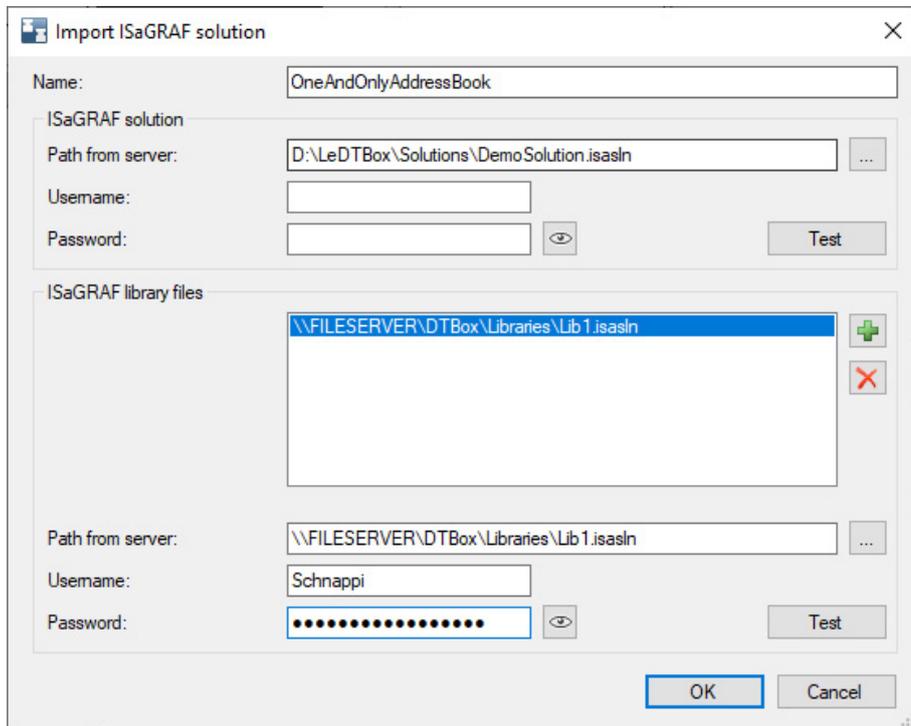


Fig. 9: Address book creation

### Name

By default the Name of the address book is the filename (without the *isasln* extension) of the ISaGRAF solution. Alternatively you can choose to provide a different name.

### ISaGRAF solution

The path entered is relative to the *ibaPDA* server so files on a different *ibaPDA* client machine might not be directly accessible. When the ISaGRAF solution is accessed using a UNC path the *Username* and *Password* to access the file should be provided.

### ISaGRAF library files

In case the ISaGRAF solution references other solutions or libraries there is a possibility to import these libraries as well. For each library you have to enter the *path* to the library file, possibly accompanied with *Username* and *Password*.

Note that *ibaPDA* will also need to access the project and other files referenced in the ISaGRAF solution file so make sure the entire (sub)folder structure is kept in its original form.

### Selecting symbols

When an address book is created return to the DTBox Request module and select the created address book in the *General* tab. Once a valid address book has been selected, the *Select symbols* link at the bottom of the module properties will become available.

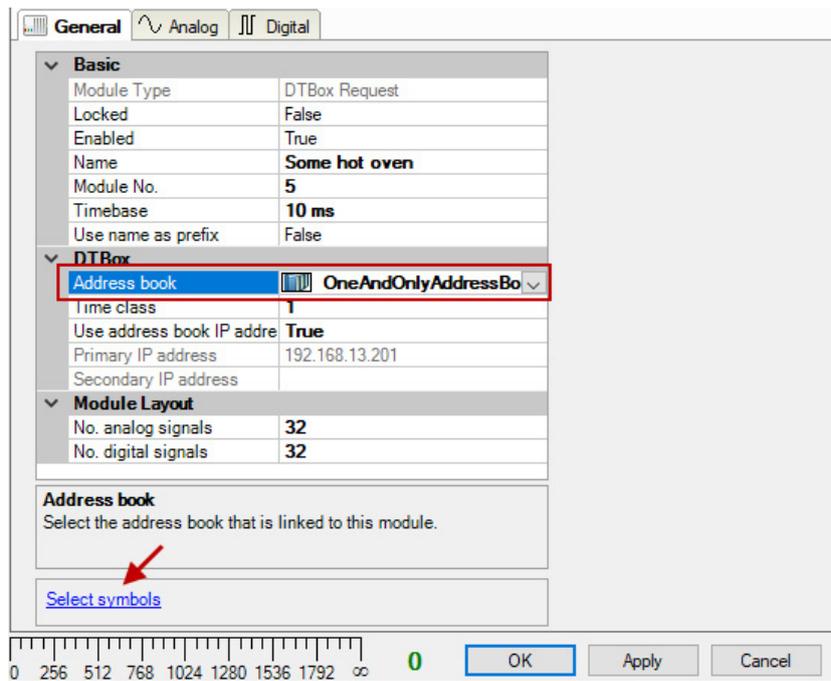


Fig. 10: Call the symbol browser

When clicking the *Select symbols* link the symbol browser will appear containing a tree structure with all available symbols in the address book.

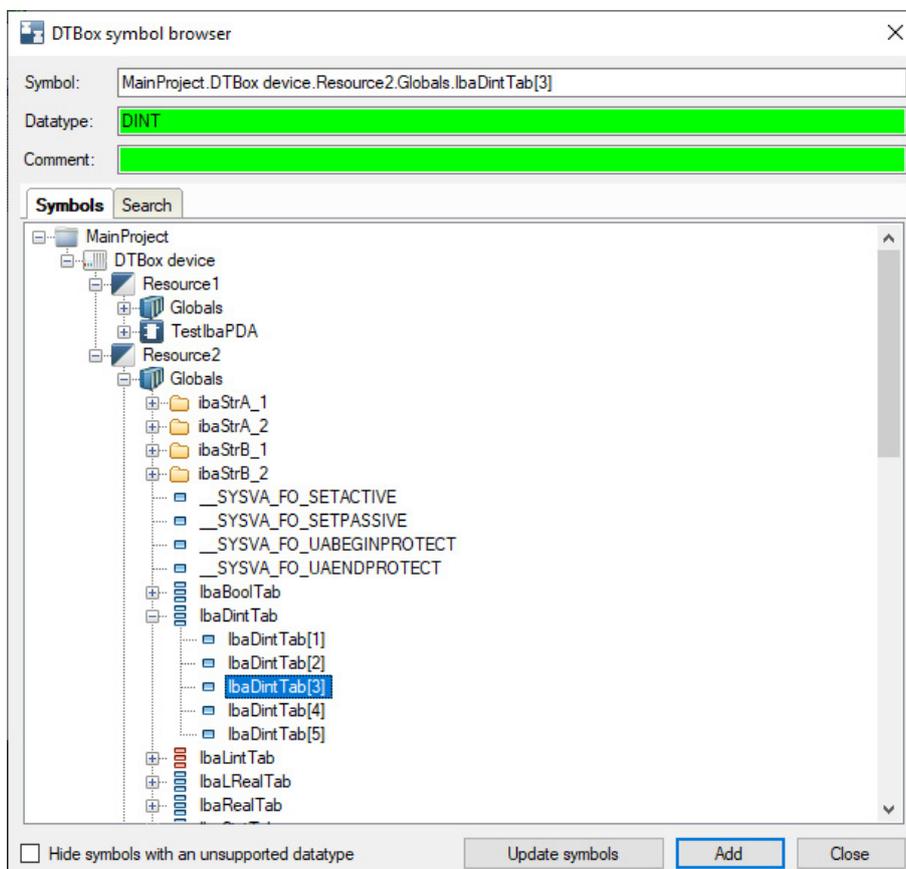


Fig. 11: DTBox symbol browser

When clicking the <Update symbols> button *ibaPDA* looks for the original ISaGRAF solution from which the address book was first generated. If the file is still available *ibaPDA* will update the symbols in the address book.

Some datatypes (e.g. time or date datatypes) are not supported by *ibaPDA* and will be marked in red in the symbol browser. You can choose not to display symbols with unsupported datatypes altogether by checking *Hide symbols with an unsupported datatype* at the bottom left of the symbol browser.

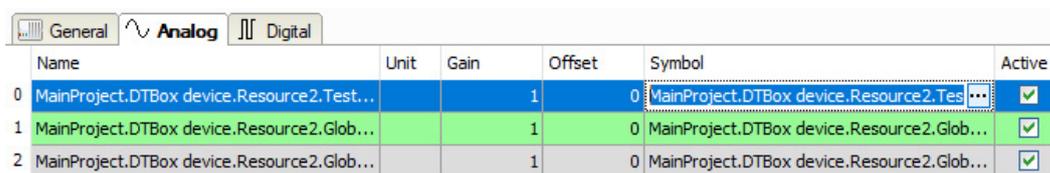
You can add symbols to the DTBox Request module by double clicking on a symbol or by selecting it and clicking the <Add> button. The symbol browser will automatically select the *Analog* or *Digital* signals tab depending on the datatype of the symbol. You can add multiple symbols by using the SHIFT and CTRL keys and clicking the <Add> button.

If you're looking for a specific symbol you can use the search function by selecting the *Search* tab in the symbol browser. You can enter a search pattern there (wildcards are allowed) and all matching symbols will be displayed.

Complete the signal selection with <Close>.

## 4.4 Signal configuration

You can find a list of all of the symbols recorded with the DTBox Request module in the *Analog* and *Digital* tabs of the DTBox Request module.



	Name	Unit	Gain	Offset	Symbol	Active
0	MainProject.DTBox device.Resource2.Test...		1	0	MainProject.DTBox device.Resource2.Tes...	<input checked="" type="checkbox"/>
1	MainProject.DTBox device.Resource2.Glob...		1	0	MainProject.DTBox device.Resource2.Glob...	<input checked="" type="checkbox"/>
2	MainProject.DTBox device.Resource2.Glob...		1	0	MainProject.DTBox device.Resource2.Glob...	<input checked="" type="checkbox"/>

Fig. 12: Analog tab with analog signals

The columns and their meaning:

### Name

When selecting signals via the symbol browser, the signal name is automatically imported.

### Unit

You can enter a physical unit.

### Gain/offset

The settings are useful to scale normalized values to physical values.

### Symbol

Contains the full path to the symbol associated with the selected signal. The symbol can be easily edited by clicking the <...> button. The symbol browser will be opened and – if the symbol is still available in the current address book – it will be selected in the tree structure.

The *Digital* tab displays the digital signals in a similar manner.

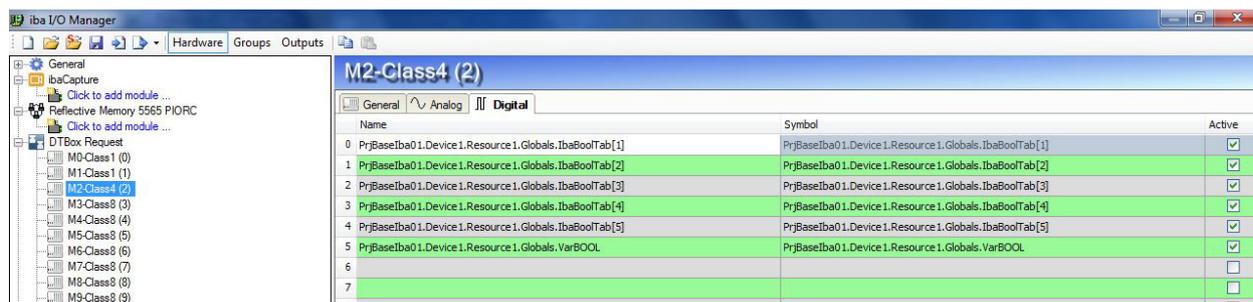


Fig. 13: Digital tab with digital signals

## 4.5 UDP data path configuration

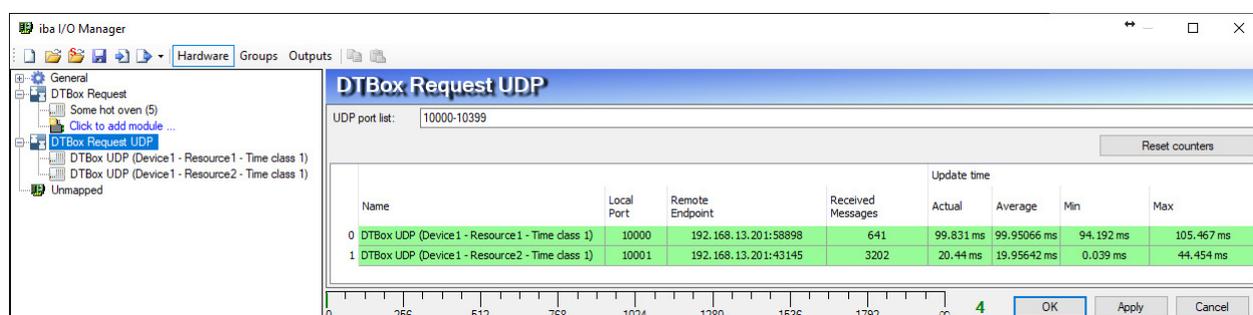


Fig. 14: UDP port configuration

When one or more modules are using UDP as a data path, the local port numbers at the *ibaPDA* server side should be configured first. These port numbers are communicated to the DTBox Request agents so they know where to send the data to.

The list of UDP ports can be entered in the *UDP port list* of the *DTBox Request UDP* interface node. A list of ports consists of a list of individual port numbers and port ranges separated by a comma. For example: 1000-1002, 1007, 2000-2002 refers to ports 1000, 1001, 1002, 1007, 2000, 2001 and 2002.

Note that when the data path is Reflective Memory no additional configuration needs to be done in *ibaPDA*; the offsets at which the data is written on the Reflective Memory board are configured in the DTBox Request agents.

## 4.6 Data modules

Once all DTBox Request modules have been properly configured apply the I/O configuration. *ibaPDA* will send the list of requested symbols the DTBox Request agents and in turn will be informed by each request agent which data path is being used - UDP or Reflective Memory. Based on that information *ibaPDA* will create data modules for the *DTBox Request UDP* interface or a Reflective Memory board. These data modules are not associated with module numbers and are used for diagnostic purposes. Each data module corresponds to one DTBox Request agent.

When selecting the *DTBox Request UDP* interface node a list of all UDP connections is displayed. The following columns are available:

**Name**

Name of the corresponding *DTBox Request UDP* data module

**Local port**

The local port number at the *ibaPDA* server side where UDP messages are being received.

**Remote endpoint**

The endpoint at the DTBox side that is sending the UDP data messages.

**Received messages**

The number of received data messages.

**Update time**

The current, average, minimum and maximum interval between two data messages.

When clicking the <Reset counters> button, all timing values and message counters are reset.

ID	Symbol	Comment 1	Address	DataType	Actual
0 [5:0]	T999_DTBOX4_FULL_DTB6822_IbaPDA.Dev...		0x0	DINT	191
1 [5:1]	T999_DTBOX4_FULL_DTB6822_IbaPDA.Dev...		0x4	DINT	291

Fig. 15: Requested signals

When selecting a UDP data module a list of the request signals mapped to this data module are displayed along with their actual value and address in the internal *ibaPDA* buffer. This can be useful for support cases.

For Reflective Memory data modules a similar layout is shown.

ID	Symbol	Comment 1	Address	DataType	Actual
0 [2:13]	PrjBaseIba01.Device1.Resource1.Globals.IbaRealTab[100]		0x1F40	LREAL	10058
1 [2:14]	PrjBaseIba01.Device1.Resource1.Globals.IbaRealTab[101]		0x1F48	LREAL	10158
2 [2:15]	PrjBaseIba01.Device1.Resource1.Globals.IbaRealTab[102]		0x1F50	LREAL	10258
3 [2:16]	PrjBaseIba01.Device1.Resource1.Globals.IbaRealTab[103]		0x1F58	LREAL	10358
4 [2:17]	PrjBaseIba01.Device1.Resource1.Globals.IbaRealTab[104]		0x1F60	LREAL	10458
5 [2:10]	PrjBaseIba01.Device1.Resource1.TestIbaPDA.local_Dint		0x1F68	DINT	139337
6 [2:18]	PrjBaseIba01.Device1.Resource1.Globals.IbaRealTab[6]		0x1F6C	REAL	658
7 [2:19]	PrjBaseIba01.Device1.Resource1.Globals.IbaRealTab[7]		0x1F70	REAL	758
8 [2:20]	PrjBaseIba01.Device1.Resource1.Globals.IbaRealTab[8]		0x1F74	REAL	858
9 [2:21]	PrjBaseIba01.Device1.Resource1.Globals.IbaRealTab[9]		0x1F78	REAL	858
10 [2:8]	PrjBaseIba01.Device1.Resource1.TestIbaPDA.local_Int		0x1F7C	DINT	0
11 [2:12]	PrjBaseIba01.Device1.Resource1.TestIbaPDA.local_Real		0x1F80	REAL	139337
12 [2:7]	PrjBaseIba01.Device1.Resource1.Globals.varUDINT		0x1F84	UDINT	139337
13 [2:5]	PrjBaseIba01.Device1.Resource1.Globals.varREAL		0x1F88	REAL	139337
14 [2:3]	PrjBaseIba01.Device1.Resource1.Globals.varDINT		0x1F8C	DINT	139337
15 [2:9]	PrjBaseIba01.Device1.Resource1.TestIbaPDA.CNT100		0x1F90	DINT	58
16 [2:4]	PrjBaseIba01.Device1.Resource1.Globals.varINT		0x1F94	INT	8265
17 [2:11]	PrjBaseIba01.Device1.Resource1.TestIbaPDA.local_Int		0x1F96	INT	8265
18 [2:0]	PrjBaseIba01.Device1.Resource1.Globals.varUINT		0x1F98	UINT	8265
19 [2:1]	PrjBaseIba01.Device1.Resource1.Globals.varUINT_Max		0x1F9A	UINT	0
20 [2:2]	PrjBaseIba01.Device1.Resource1.Globals.varUINT_Min		0x1F9C	UINT	0
21 [2:6]	PrjBaseIba01.Device1.Resource1.Globals.varSINT		0x1F9E	SINT	73

Fig. 16: Requested analog signals via Reflective Memory

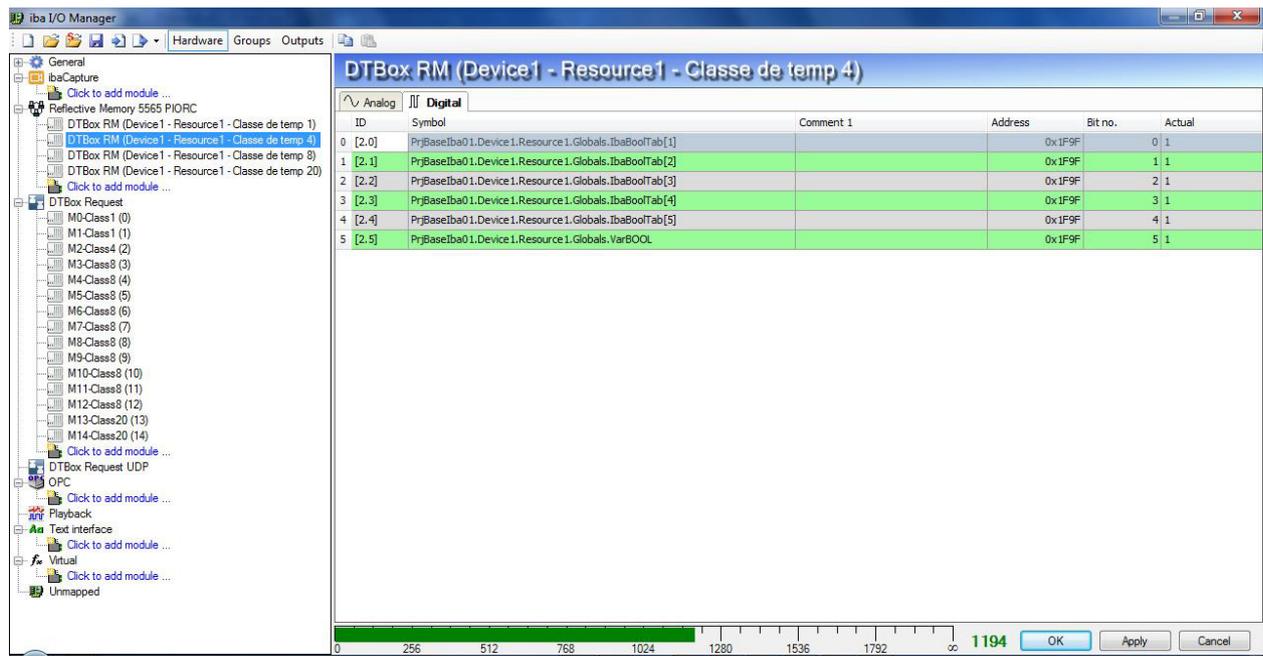


Fig. 17: Requested digital signals via Reflective Memory

# 5 Diagnosis

## 5.1 License

If the "DTBox Request" interface is not displayed in the signal tree, you can either check in *ibaPDA* under *General - Settings - License info* in the I/O Manager or in the *ibaPDA* service status application, whether your license "Request DTBox" has been properly recognised. The number of licensed connections is indicated in brackets.

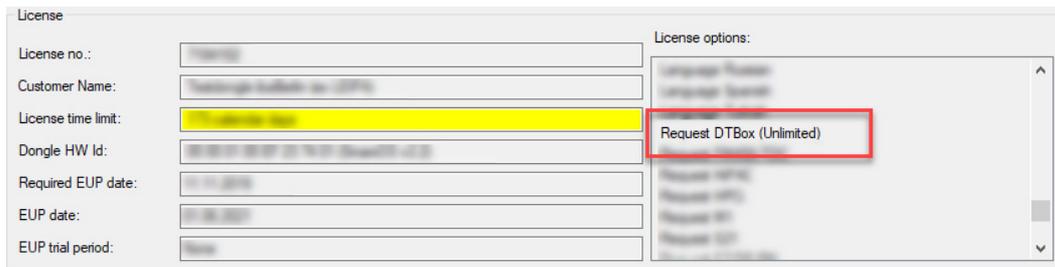


Fig. 18: License display in the ibaPDA I/O Manager

## 5.2 Connection diagnostics

Remote Endpoint		Received Watchdog messages	Invalid Messages	Failover Count	Watchdog interval			
					Actual	Average	Min	Max
0+1	202.202.202.202:10000	17	0	0	10000.12 ms	10000 ms	9999.75 ms	100000.3 ms
	202.202.202.203:10000	17	0	0	10000.79 ms	10000 ms	9999.7 ms	100000.3 ms

Fig. 19: Diagnostics tab

In the *Diagnostics* tab of the DTBox Request interface you will get an overview of all (TCP/IP) control connections with the DTBox request agents. For each control connection an entry in the grid is shown with the following properties:

### Remote endpoint

IP address and port number of the DTBox Request agent to which *ibaPDA* is connected (or trying to connect).

### Received watchdog messages

Total amount of received watchdog messages.

### Invalid messages

Displays how many messages have been received from the DTBox Request agent that contained invalid data. This value should always be 0 under normal circumstances.

### Failover count

Indicates how many times an agent has switched from being standby to active during the acquisition. This value is only displayed in case of a failover scenario.

When failover support is enabled in the DTBox two control connections are always maintained: one to the active controller and one to the standby controller. In the diagnostic grid these connections are grouped. The entry for the standby controller in the diagnostic grid is shown with an orange background.

### Watchdog interval

These columns show the actual, average, minimum and maximum time between two received watchdog messages.

### <Open log file>

A log file for DTBox related messages can be opened.

### <Reset counters>

Diagnostic counters can be reset

The background color in the grid indicates the connection status:

- Green: the connection is active
- Red: the connection is inactive (e.g. in case of a connection problem).

## 5.3 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.

The log file can be opened 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 will find the log files in the program path of the *ibaPDA* server (...\\Programs\\iba\\ibaPDA\\Server\\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)

## 5.4 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.

Open a Windows command prompt.



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

With an existing connection you receive several replies.

```
Administrator: C:\Windows\system32\cmd.exe
C:\Users>ping 192.168.21.120
Pinging 192.168.21.120 with 32 bytes of data:
Reply from 192.168.21.120: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.21.120:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\Users>
```

Fig. 20: PING successful

With no existing connection you receive error messages.

```
Administrator: C:\Windows\system32\cmd.exe
C:\Users>ping 192.168.21.121
Pinging 192.168.21.121 with 32 bytes of data:
Reply from 192.168.21.104: Destination host unreachable.
Ping statistics for 192.168.21.121:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
C:\Users>
```

Fig. 21: PING unsuccessful

## 6 Appendix

### 6.1 Architecture

The DTBox runs ISaGRAF. The ISaGRAF project consists of multiple resources. In each resource one or more iba request agents can be added. These agents can each have a different time class. The time class is configured by *ibaPDA* and is expressed as a number of cycles. When the time class is 1 it means that the agent will send the data every cycle. When the time class is 2 then the agent will send the data only every second cycle. The time class factor is entirely configurable by the user.

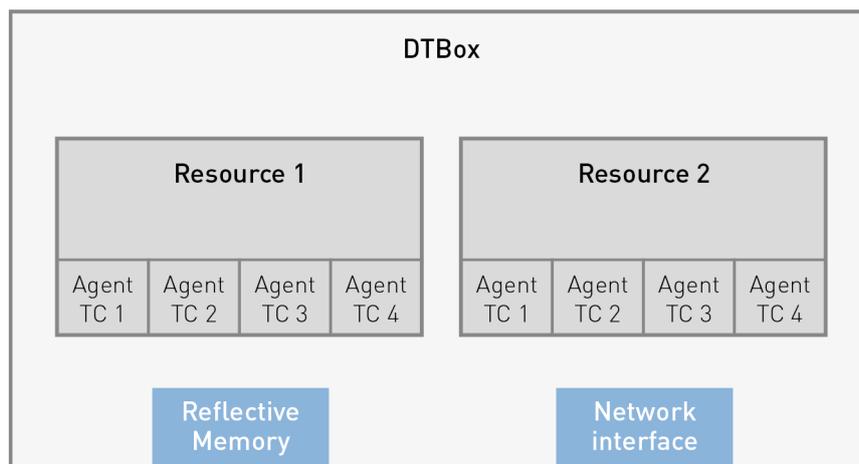


Fig. 22: DTBox architecture

Each agent is independent of all other agents (both within the same resource and within other resources). *ibaPDA* connects to each agent separately via TCP. This connection is called the control path.

For each resource there are 4 agents in order to manage different time classes. Each agent is listening on its own TCP port. The maximum number of resources is 100, that means that we need to reserve 400 TCP ports inside DTBox.

**The calculation rule for determining the port numbers is the following:**

$$\text{TCP Port} = (\text{Base port} + (\text{ResNum} - 1) * 4 + \text{Agent Number})$$

Base port is fixed : 10000

Agent Number is : 0..3

Resource Number is : 1..100

**Example:**

If *ResNum* is 2 and *Agent Number* is 2: TCP Port = 10000 + (2-1) \* 4 + 2 = 10006

Resource number	Agent number	TCP port
1	0	10000
1	1	10001
1	2	10002
1	3	10003
2	0	10004
2	1	10005
2	2	10006
2	3	10007
3	0	10008
3	1	10009
3	2	10010
3	3	10011
100	0	10396
100	1	10397
100	2	10398
100	3	10399

The control path is used to exchange watchdog and request messages.

## 7 Support and contact

### Support

Phone: +49 911 97282-14  
Fax: +49 911 97282-33  
Email: support@iba-ag.com

---

#### Note



If you require support, indicate the serial number (iba-S/N) of the product or the license number.

---

### 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](http://www.iba-ag.com)**.