



ibaPDA

Appendix

Manual Part 7
Issue 8.9

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

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 2024, 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.

Version	Date	Revision	Author	Version SW
8.9	12-2024	Uninstalling ibaPDA	rm	8.9.0

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.

Contents

1	About this documentation	4
1.1	Target group and previous knowledge	4
1.2	Notations	4
1.3	Used symbols.....	5
1.4	Documentation structure	6
2	ASCII watchdog telegram structure.....	7
3	Binary watchdog telegram.....	8
4	Setting up the ibaPDA client as an ActiveX control.....	9
4.1	Methods and properties.....	11
4.2	ibaPDA ActiveX control in SIMATIC WinCC	17
4.3	ibaPDA ActiveX control in SIMATIC WinCC flexible	21
4.4	Sample VB Script for printing out of an ActiveX Control.....	24
5	ibaPDA-S7-Xplorer Proxy for PC/CP interface.....	25
6	The I/O configuration file	26
6.1	Saving an I/O configuration	26
6.2	Loading an existing I/O configuration	26
6.3	Loading a backup configuration file.....	28
6.4	Exporting an I/O configuration to a text file	28
6.5	Importing a configuration text file	31
7	Advanced filter in tables.....	32
7.1	Simple filter function	32
7.2	Advanced filter functions.....	34
7.3	Search function.....	38
8	Uninstalling ibaPDA.....	39
9	Support and contact.....	41

1 About this documentation

This documentation describes the function and application of the software *ibaPDA*.

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.

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>
Filenames, paths	<i>Filename, Path</i> Example: <i>Test.docx</i>

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.

1.4 Documentation structure

This documentation fully describes the functionality of the *ibaPDA* system. It is designed both as a tutorial as well as a reference document. The sections and chapters essentially follow the procedure for configuring the system.

In addition to this documentation, you can examine the version history in the main menu, *Help – Version history* (file [versions.htm](#)) for the latest information about the installed version of the program. This file not only lists the bugs that have been eliminated, but also refers to extensions of the system in note form.

In addition, special "NewFeatures..." documentation comes with any software update that includes significant new features, which provides a more detailed description of the new features.

The state of the software to which the respective part of this documentation refers is listed in the revision table on page 2.

The *ibaPDA* system documentation (PDF and printed version) is divided into seven separate parts. Each part has its own section and page numbering beginning at 1, and is updated independently.

Part 1	Introduction and installation	General notes, license policy and add-ons Installation and program start User interface, system architecture, client server User management, printing
Part 2	I/O Manager	Basic information about the I/O Manager, general settings Groups and vector signals, text signals, outputs, configuration files
Part 3	Data interfaces and modules	Interfaces for the measurement data acquisition Standard interfaces, ibaFOB, Ethernet-based interfaces and more. For interfaces for which there are separate manuals, these are referred to.
Part 4	Expression builder	All functions for calculating virtual signals
Part 5	Data storage	Types of data recording, recording profiles, signal selection
Part 6	Data visualization	All display modes for live data, their operation and settings
Part 7	Appendix	Various additions, error listings, etc.

2 ASCII watchdog telegram structure

Beginning										End (= 10)	
Timestamp ibaPDA server PC	PC name	Iba product name	Data store 1			Data store 2			Data store n		
dd/mm/yy hh:mm:ss.ms			File name	Status	MB free memory for 1 st data store	File name	Status	MB free memory for 2 nd data store	File name
		PDA6	Complete path	Idle Waiting for Trigger Recording Posttrigger	Free memory space on harddisk, where files of data store 1 are stored.	Complete path	Idle Waiting for Trigger Recording Posttrigger	Free memory space on harddisk, where files of data store 1 are stored.	Complete path
Sample telegram with information about two data stores											
25/10/05 13:45:10.280	pda-pc	PDA6	C:\DATA\ABC00.dat	Recording	1314	C:\DATA\DEF01.dat	Recording	1314		10	

3 Binary watchdog telegram

```
#ifndef PDA_WATCHDOG_H
#define PDA_WATCHDOG_H

#define N_MAX_DATASTORES      16

typedef struct _PDA_DATASTORE_INFO
{
    unsigned short status;          // (0) Status of the datastore:
                                    //      =0: Inaktiv:
                                    //      =1: Waiting for trigger
                                    //      =2: Recording
                                    //      =3: Post-trigger
    unsigned short directory;       // (2) Current directory:
                                    //      =0: Pda is writing to the base directory
                                    //      =1: Pda is writing to the backup directory
    unsigned int   freeSpace;       // (4) Free space on the harddisk in MB
                                    //      (only available if datastore is active)
    unsigned char  reserved[4];     // (8)
} PDA_DATASTORE_INFO, *PPDA_DATASTORE_INFO;

typedef struct _QDR_DATASTORE_INFO
{
    unsigned short status;          // (0) Status of the QDR datastore:
                                    //      =0: Inaktiv:
                                    //      =1: Not synchronized
                                    //      =2: Synchronized
    unsigned short directory;       // (2) Current directory:
                                    //      =0: Pda is writing to the base directory
                                    //      =1: Pda is writing to the backup directory
    unsigned int   freeSpace;       // (4) Free space on the harddisk in MB
                                    //      (only available if datastore is active)
    unsigned char  reserved[4];     // (8)
} QDR_DATASTORE_INFO, *PQDR_DATASTORE_INFO;

typedef struct _PDA_WATCHDOG
{
    unsigned int   msgCounter;      // (0) Message counter is incremented after each message
    unsigned int   msgVersion;      // (4) Version number (currently = 1)

    unsigned int   isMeasuring;     // (8) =1: Pda is measuring
    unsigned char  allOk           : 1; // (12.0) =1: Everything is OK (measuring und no disabled signals)
    unsigned char  disabledSignals : 1; // (12.1) =1: There are disabled signals
    unsigned char  reserved[3];     // (13)
    unsigned int   captureCamStatus[2]; // (16) Bit per connection
                                    //      to ibaCapture-CAM server
                                    //      bit = 1: connection ok
    unsigned int   captureHmiStatus[2]; // (24) Bit per connection
                                    //      to ibaCapture-HMI server
                                    //      bit = 1: connection ok

    QDR_DATASTORE_INFO qdrDatastore; // (32) Information about the QDR datastore
    PDA_DATASTORE_INFO datastores[N_MAX_DATASTORES]; // (44) Information
                                                    //      about all datastores
} PDA_WATCHDOG, *PPDA_WATCHDOG;
#endif
```

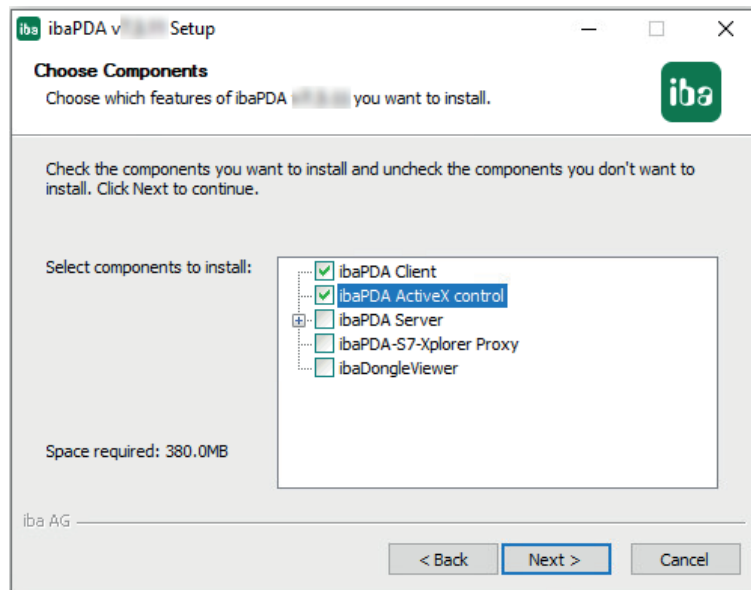

4 Setting up the ibaPDA client as an ActiveX control

Requirements and licenses

As a prerequisite for using the *ibaPDA* client as an ActiveX control, the following software components must be installed on the computer on which the application is to run with ActiveX:

- ibaPDA Client
- ibaPDA ActiveX control

For this purpose, start the *ibaPDA* installation on the corresponding computer and select the above-mentioned components in the selection dialog.



Each ActiveX control establishes a client connection to the configured *ibaPDA* server using the configured user name and password. If there are simultaneously several ActiveX controls in a host process (ActiveX container), which connect to the same *ibaPDA* server using the same user, these ActiveX controls will share a common connection.

An example would be two ActiveX controls located side by side on one image in an HMI application like SIMATIC WinCC, or even on two different images, connecting to the same *ibaPDA* server using the same user. The runtime process for the HMI application is the ActiveX container.

If two ActiveX controls with different users connect to the same *ibaPDA* server, then two connections are created.

An *ibaPDA* client license is required for each connection. If *ibaQPanel* is also used in the views shown in the *ibaPDA* ActiveX control, an *ibaQPanel* license is additionally required.

For workstations on which no images can be called up with the *ibaPDA* ActiveX Control (e.g., due to a corresponding configuration in the HMI application), no license is required.

If you wish to make sure that a computer is always guaranteed to get the license it needs, you can reserve client and QPanel licenses for selected computers using the server access control feature in *ibaPDA* (see *ibaPDA* manual part 1).

The connection is only established when the ActiveX control is called for the first time in the host process. An example of this is when an image with a projected ActiveX control is displayed for the first time in the HMI application, such as SIMATIC WinCC.

From the moment the connection is established, the license is also activated. When switching to an image without ActiveX control, the connection to the *ibaPDA* server is maintained in the background, consequently the license is still in use. The data for the signals visualized in the views continues to be queried in the background via the maintained connection, so that the signal trends can be displayed immediately when the image is displayed again with ActiveX Control.

The *ibaPDA* ActiveX control does not establish a connection to the *ibaPDA* server in design mode (e.g., during configuration in SIMATIC WinCC Graphics Designer). In runtime mode the ActiveX control will automatically try to connect to the configured *ibaPDA* server.

Note

The ActiveX control can only be used for visualization. This means you cannot start/stop the acquisition or configure the I/O manager and data storage. Layout changes and interaction with the signal displays are possible with the corresponding permissions.

The ActiveX Control creates log files and saves them, e.g., in Windows 10 at `LocalAppData%\iba\ibaPdaActiveX`.

This creates a subdirectory with the name of the ActiveX host process (e.g., "PDIrt" for the SIMATIC WinCC Runtime process). All ActiveX control instances within an ActiveX host process use the same log file.

4.1 Methods and properties

The *ibaPDA* ActiveX Control according to its definition provides some object properties which are listed in the table below.

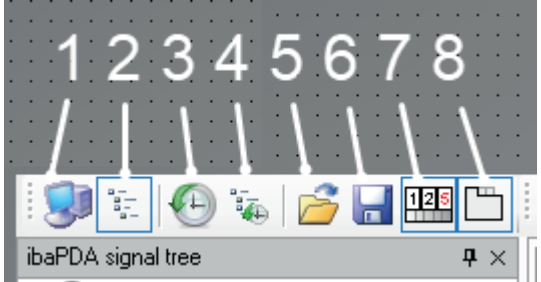
The properties can be manipulated by using scripts which enable you to change the ActiveX Control dynamically. Therefore, some methods are available which are described further below.

Object properties of ibaPDA ActiveX Control

Property	Select, enter	Note
LayoutFile	Path specification [string]	Enter the path and file name of a layout file here. The <i>ibaPDA</i> client will always load this file when the image is called up (even after an image change). If <i>AutoSaveLayout</i> is enabled, layout changes will be saved to this file. Even if you click the Save button in the ActiveX toolbar in runtime mode, the layout is written to this file. Changing this property at runtime will not trigger a reload of the layouts.
ServerAddress	IP address or computer name [string]	Enter the IP address or the computer name of the <i>ibaPDA</i> server to which the <i>ibaPDA</i> client should connect in this ActiveX control.
PortNr	Port number [int32]	Port through which the client connects to the server Default: 9170
UserName	User name [string]	Enter a user name with which the <i>ibaPDA</i> client should log on to the <i>ibaPDA</i> server in this ActiveX Control. The user name must already exist in the user management administration of the <i>ibaPDA</i> server. You must enter a user name here, unless you are using the <i>UseCurrentwindowsUser</i> login (see below).
Password	Password [string]	Enter the password matching the <i>ibaPDA</i> user here.
ShowToolbars	Yes/No [bool]	Lets you choose whether the ActiveX- and/or Layout toolbars are displayed (Yes) or not (No). Changing this property via a script at runtime will immediately take effect.

Property	Select, enter	Note
ShowTabs	Yes/No [bool]	Lets you choose whether the view tabs are displayed (Yes) or not (No). This does the same as the button in the Active-X toolbar. It overwrites the settings in the layout configuration. Changing this property via a script at runtime will immediately take effect.
AutoSaveLayout	Yes/No [bool]	Allows you to specify whether changes to the layout are automatically saved to the <i>CurrentLayout.lay</i> file or not. If you have specified a file for the <i>LayoutFile</i> property (see above), then the changes will be saved to that file.
DisableLayoutChange	Yes/No [bool]	Allows you to specify whether the layout may be changed (Yes or No). If layout changes are disabled here, then no <i>ibaPDA</i> user – even users with the corresponding rights – can change the layout. If layout changes are permitted here, then the user's <i>Change layout</i> right is also taken into account.
DisableInteraction	Yes/No [bool]	Allows you to specify whether the signal displays can be operated interactively (Yes or No). If such interactions are disabled here, then no <i>ibaPDA</i> user – even users with the corresponding rights – can interact with the signal displays. If such interactions are permitted here, then the user's <i>Interact with views</i> right is also taken into account.
DisableMessageboxes	Yes/No [bool]	Allows you to specify whether notification boxes from <i>ibaPDA</i> may be displayed here (Yes or No). These are internal system notifications, e.g., if the I/O configuration has been changed on another client.
HdServerAddress	IP address or computer name [string]	Enter the IP address or the computer name of the <i>ibaHD</i> server to which the <i>ibaPDA</i> client should connect in this ActiveX control.
HdPortNr	Port number [int32]	Port through which the client connects to the server Default: 9180

Property	Select, enter	Note
ShowClientDisconnect	Yes/No [bool]	<p>This lets you define how to proceed if no more free <i>ibaPDA</i> client licenses are available for the ActiveX control.</p> <ul style="list-style-type: none">■ Yes: In the <i>ibaPDA</i> client, a dialog appears in which you can select another client and log off in order to free up the license for the ActiveX control.■ No: An error message is merely displayed notifying you that no more client licenses are available.
HdUserName	User name [string]	<p>Lets you enter the user name with which the <i>ibaPDA</i> client should log on to the <i>ibaHD-Server</i> in this ActiveX control. The user name must already exist in the user management of the <i>ibaHD</i> server. You must enter a user name here, unless you are using the <i>HdUseCurrentwindow-sUser</i> login (see below).</p>
HdPassword	Password [string]	<p>Enter the password matching the <i>ibaHD-Server</i> user here.</p>

Property	Select, enter	Note
ShowActiveXToolbar	Yes/No [bool]	<p>This defines whether the special ActiveX toolbar is displayed (Yes or No).</p> <p>The functions of this toolbar can only be used in runtime mode.</p>  <p>1 Select <i>ibaPDA</i>-Server 2 Show/hide <i>ibaPDA</i> signal tree 3 Select <i>ibaHD</i>-Server 4 Show/hide <i>ibaHD</i>-Server signal tree 5 Open layout, loads a layout from a file 6 Save layout, saves the current layout in the file specified above 7 Show/hide window for digital displays 8 Show/hide view tabs</p> <p>Changing this property via a script at runtime will immediately take effect.</p>
ShowLayoutToolbar	Yes/No [bool]	<p>This defines whether to show (Yes) or hide (No) the Layout toolbar. Changing this property via a script at runtime will immediately take effect.</p>
UseCurrentWindowsUser	Yes/No [bool]	<p>Lets you define whether the current Windows user on the SIMATIC WinCC computer can log on to the <i>ibaPDA</i> server (Yes or No).</p> <p>This permission can be granted in addition to the entering a local <i>ibaPDA</i> user (see <i>UserName</i> above).</p> <p>The prerequisite for this is that an Active Directory is set up and domain users or groups to which the Windows user also belongs are configured in the user management of the <i>ibaPDA</i> server.</p>

Property	Select, enter	Note
HdUseCurrentWindow- sUser	Yes/No [bool]	<p>Lets you define whether the current Windows user on the SIMATIC WinCC computer can log on to the <i>ibaHD</i> server (Yes or No).</p> <p>This permission can be granted in addition to the entering a local <i>ibaHD-Server</i> user (see <i>HdUserName</i> above).</p> <p>The prerequisite for this is that an Active Directory is set up and domain users or groups to which the Windows user also belongs are configured in the user management of the <i>ibaHD</i> server.</p>

Table 1: Object properties of ibaPDA ActiveX control

Methods to control the ibaPDA ActiveX Control

When using scripts you can dynamically control the ActiveX control by using the following methods in the script.

Method name	Parameter	Result	Description
Print	Title [string] ShowPrintSetup [bool]		This prints the currently visible layout. The title will be shown at the top of the page. If <i>ShowPrintSetup</i> is true then the print setup dialog is shown. You can select the printer there and select page size, margins, scaling etc. If <i>ShowPrintSetup</i> is false the last saved settings are used.
ConnectToPdaServer			This will attempt to connect to an <i>ibaPDA</i> server using the current values of the <i>ServerAddress</i> , <i>PortNr</i> , <i>UserName</i> and <i>Password</i> properties (see table before). So in the script you can change the properties and then call <i>ConnectToPdaServer</i> to switch to a different server or log in as a different user.

Method name	Parameter	Result	Description
ConnectToHdServer			This will attempt to connect to an <i>ibaHD-Server</i> using the current values of the HdServerAddress, HdPortNr, HdUserName and HdPassword properties (see table before). So in the script you can change the properties and then call <i>ConnectToHdServer</i> to switch to a different server or log in as a different user.
LoadLayoutFromFile	LayoutFile [string]	bool	This will try to load the layouts file <i>LayoutFile</i> . It basically does the same thing as when you manually click the open layout button on the Active-X toolbar and then select the layout file. The return value is true if the layouts were loaded from the file.
SaveLayoutToFile	LayoutFile [string]	bool	This will try to save the current layouts to the file <i>LayoutFile</i> . The return value is true if the save was successful.

Table 2: Methods to control the ibaPDA ActiveX Control

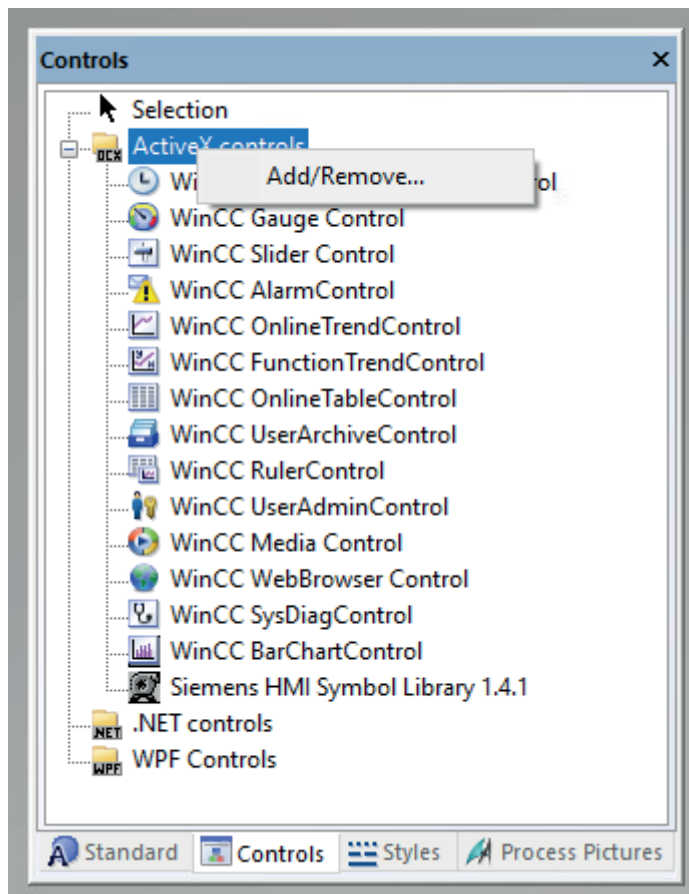
4.2 ibaPDA ActiveX control in SIMATIC WinCC

To set up an ibaPDA *ActiveX Control* on a SIMATIC WinCC system, proceed as follows.

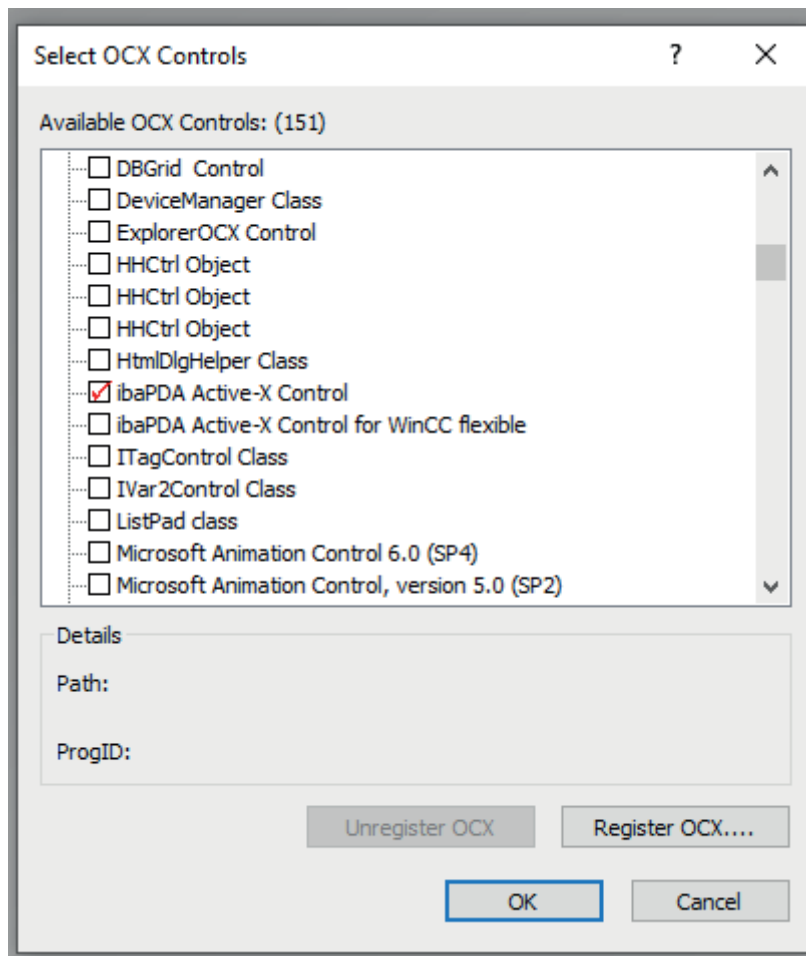
Please note that as a prerequisite, the components *ibaPDA-Client* and *ibaPDA ActiveX-Control* must be installed on the SIMATIC WinCC computer.

Create ActiveX control

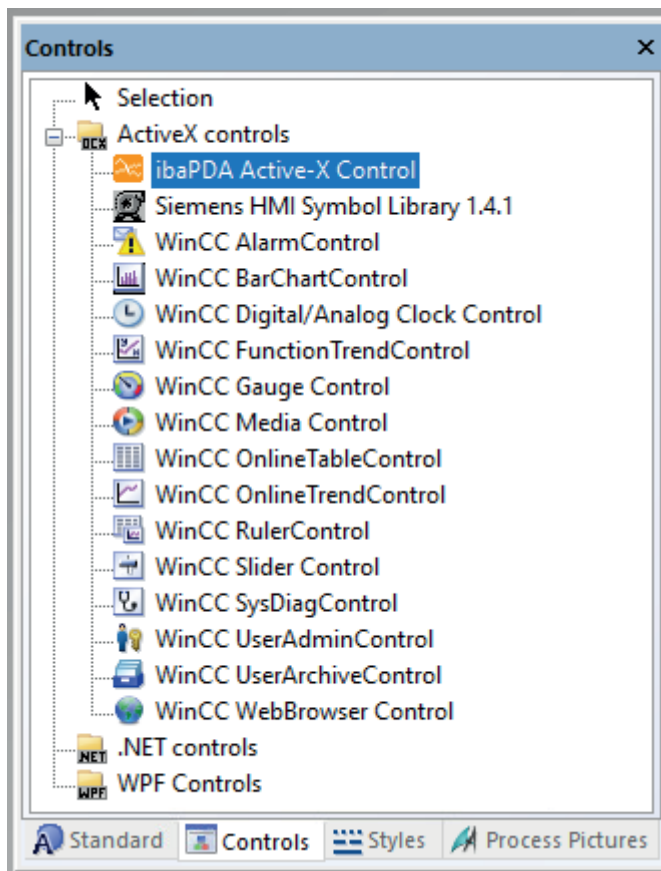
1. Create a new image in the Graphics Designer of SIMATIC WinCC.
2. Open the window with the controls ("Controls"). If *ibaPDA ActiveX Control* is not yet present, you will need to add this first. If the control is already present, proceed to step 5).
3. Right-click in the "Controls" window and select *Add/Remove...* in the context menu.



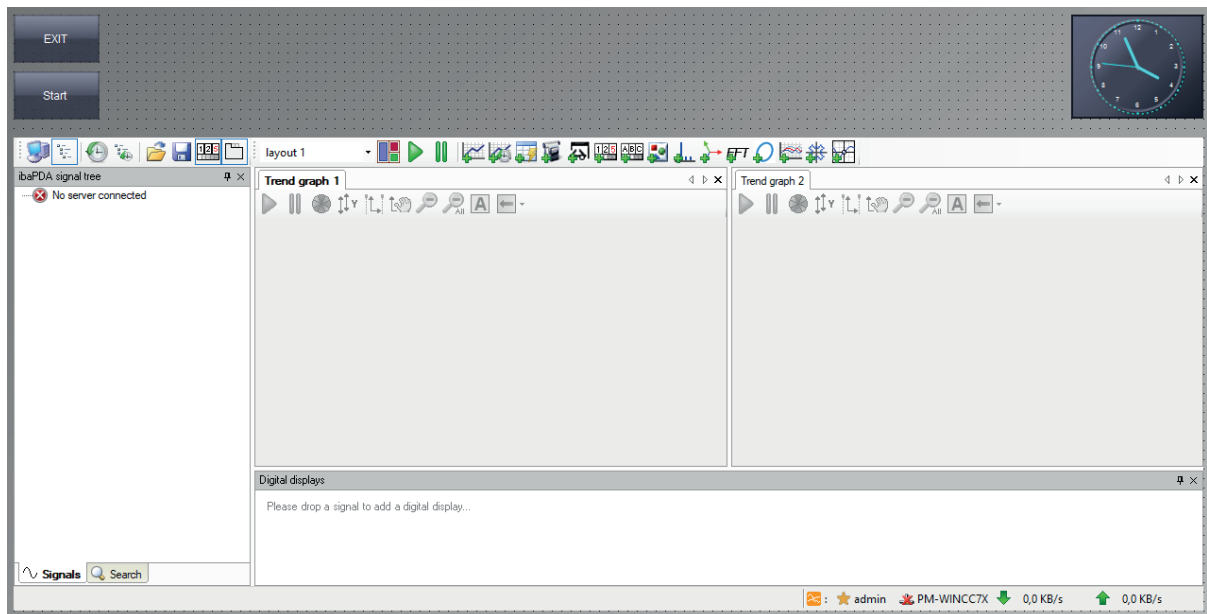
4. In the "Select OCX Controls" dialog, set a check mark next to *ibaPDA ActiveX Control* and close the dialog again with <OK>.



5. Now select the *ibaPDA ActiveX Control* in the "Controls" dialog and insert a new instance into the image.



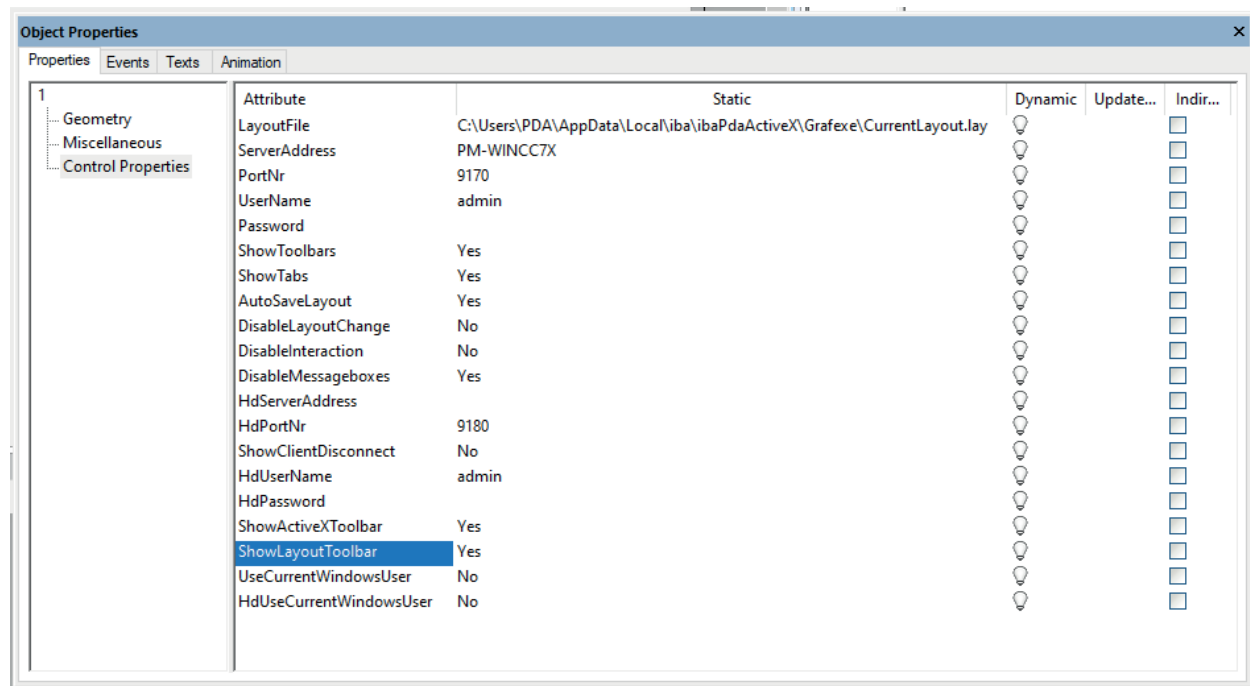
6. The result is a typical *ibaPDA* client interface, whose appearance may differ in design mode and in the subsequent runtime mode, because it is initially a placeholder.



Object properties

The *ibaPDA ActiveX Control* has some properties that you can configure.

In the Graphics Designer, double-click the ActiveX control to access its properties.



A listing of the properties and their description can be found under [➤ Methods and properties](#), page 11.

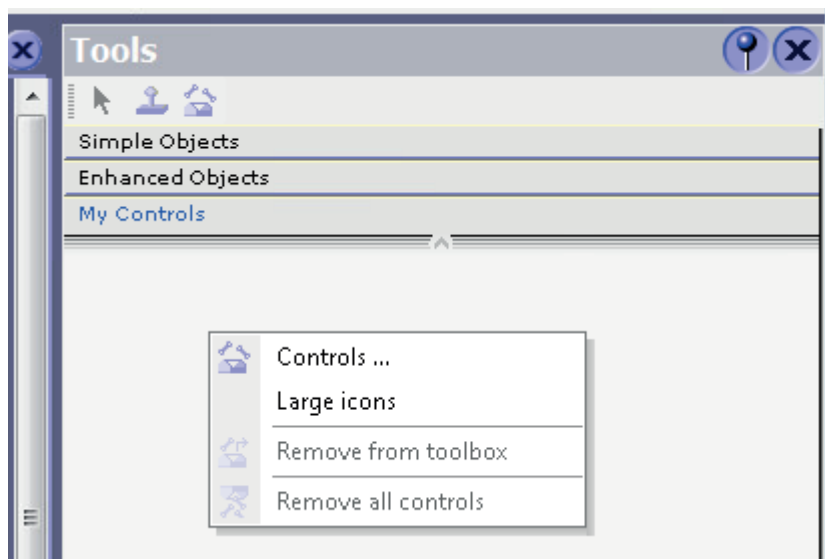
4.3 ibaPDA ActiveX control in SIMATIC WinCC flexible

To set up an ibaPDA *ActiveX Control* on a SIMATIC WinCC flexible system, proceed as follows.

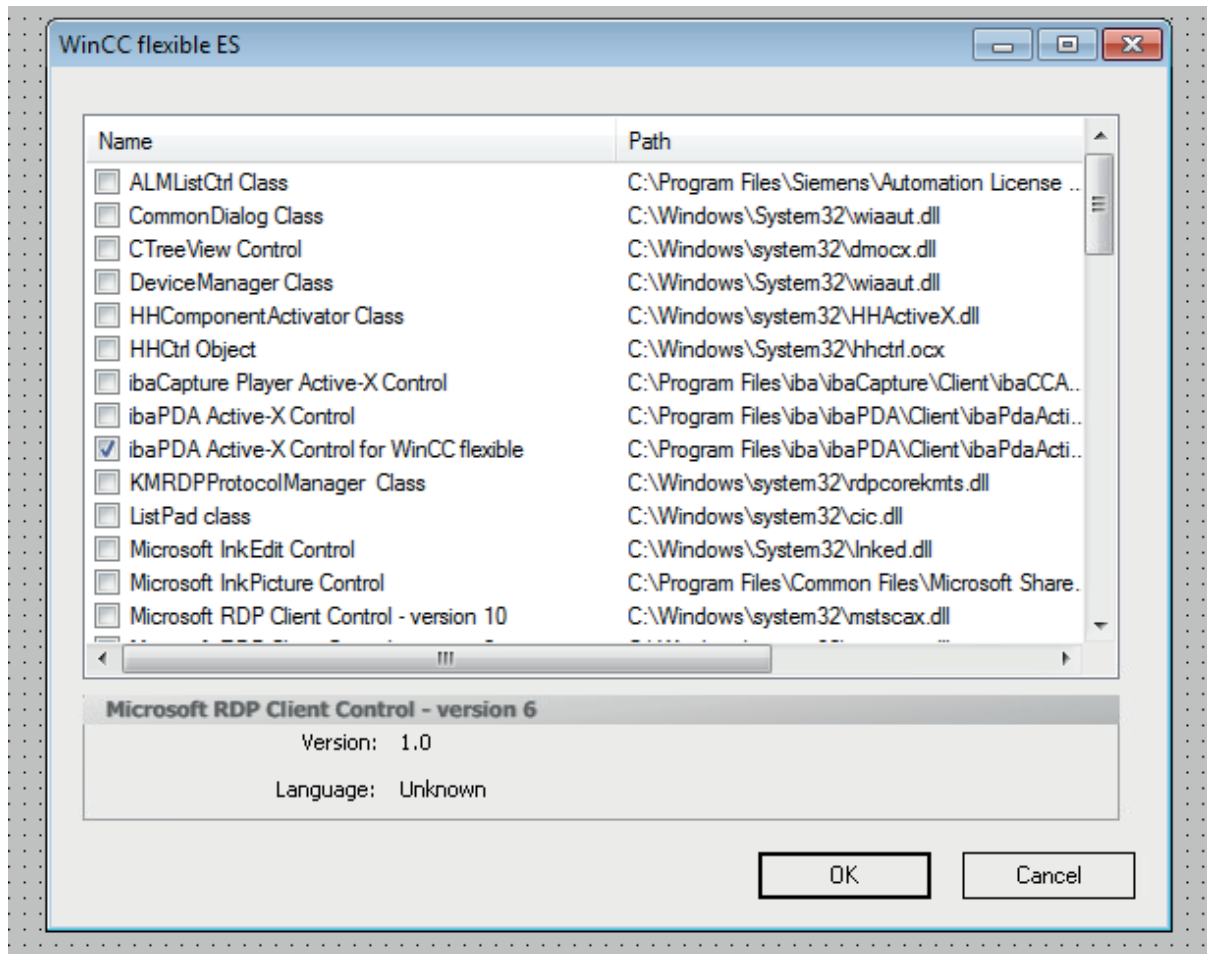
Please note that as a prerequisite, the components *ibaPDA-Client* and *ibaPDA ActiveX-Control* must be installed on the SIMATIC WinCC computer.

Create ActiveX control

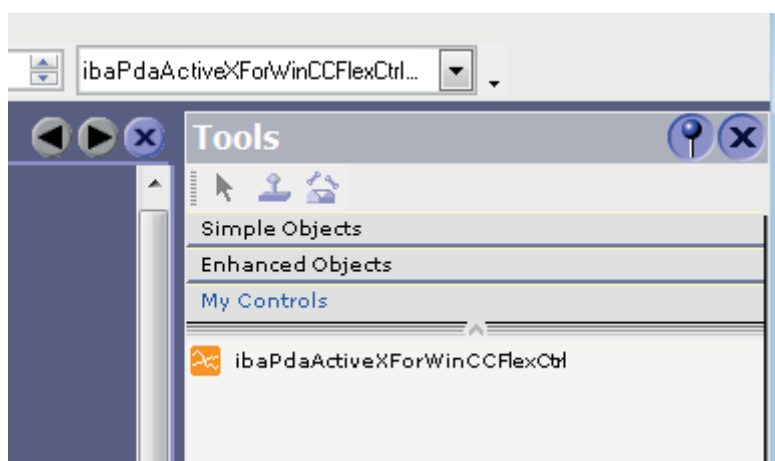
1. Create a new image in SIMATIC WinCC flexible.
2. Open the "Tools" window. If *ibaPdaActiveXForWinCCFlex Control* is not yet listed under *My Controls*, you will need to add this first. If the control is already present, proceed to step 5).
3. Right-click in the "Tools" window and select *Controls...* in the context menu.



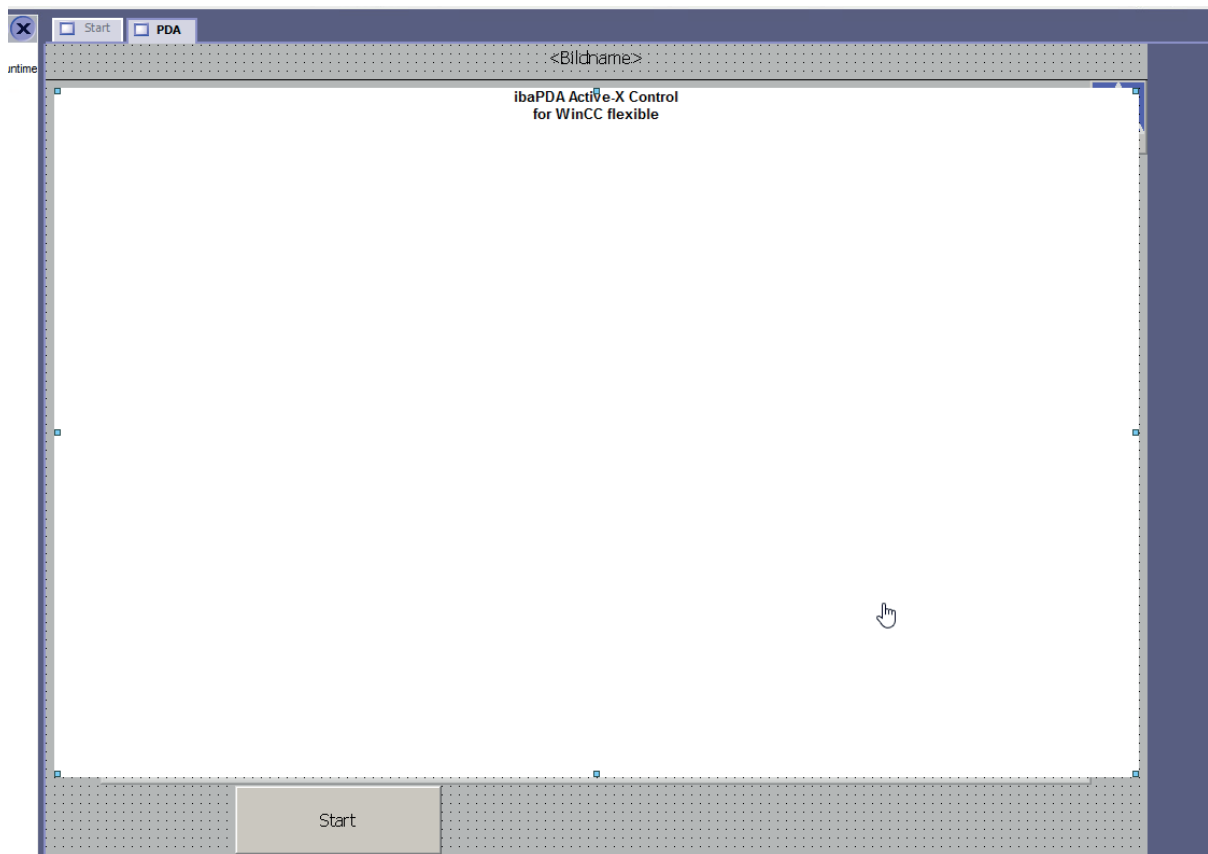
4. In the "WinCC flexible ES" dialog, set a check mark next to *ibaPDA ActiveX Control for WinCC flexible* and close the dialog again with <OK>.



5. Now select the control *ibaPdaActiveXForWinCCFlexCtrl* in the "Tools" window under *My Controls* and insert a new instance into the image. To do this, drag a rectangular frame with your mouse.



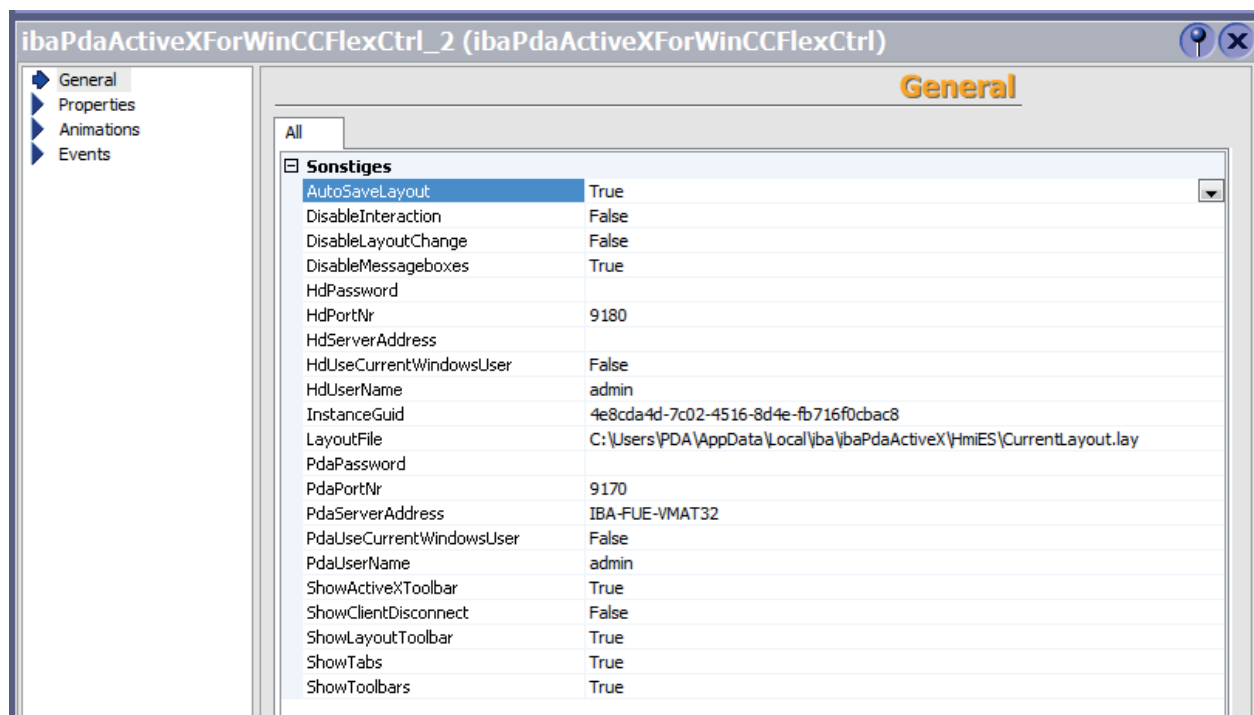
6. The result is a white rectangular box that serves as a placeholder in design mode.



Object properties

The *ibaPDA ActiveX Control* has some properties that you can configure.

Double-click on the ActiveX control to get to its properties.



The properties are similar to those of the ActiveX control for SIMATIC WinCC. They are just listed in a different order, and there is also the Guid of the instance (*InstanceGuid*).

A listing of the properties and their description can be found under ➤ *Methods and properties*, page 11.

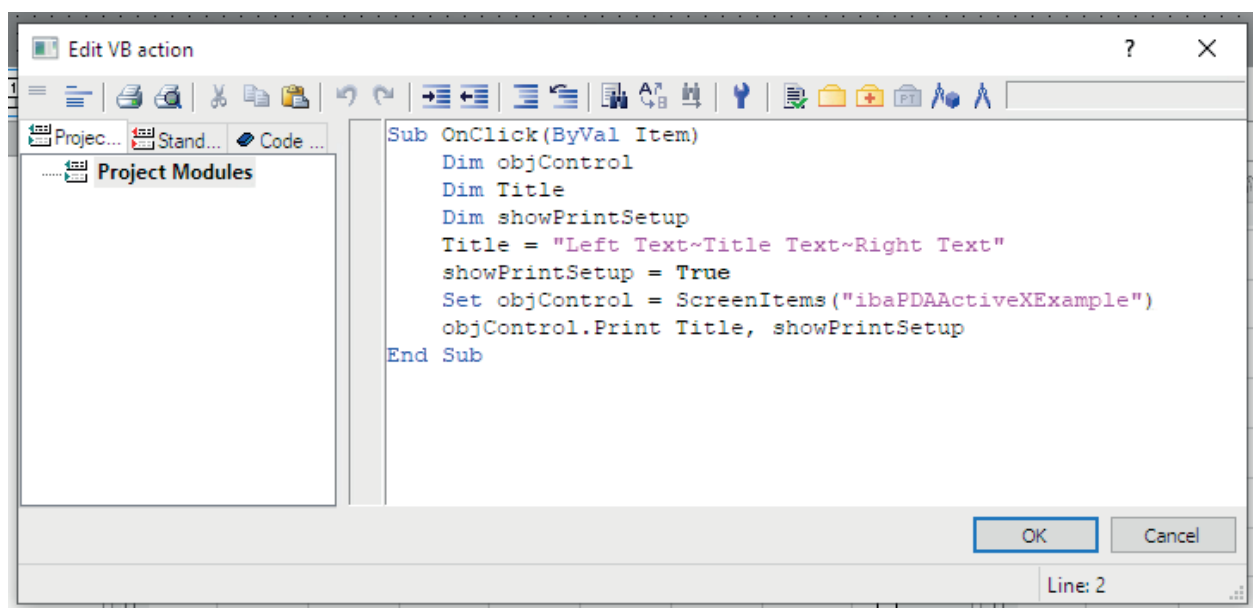
4.4 Sample VB Script for printing out of an ActiveX Control

The *ibaPDA* ActiveX control provides a print function that can be accessed from the ActiveX container (e.g., via VBScript using a button in SIMATIC WinCC).

The print function has 2 arguments:

- **STRING title:** The header of the printed document. It is possible to use the tilde character ("~") to separately specify three texts for the left, middle and right areas of the header.
- **BOOL showPrintSetup:** Set this option to TRUE to run the print settings form before printing. The print settings are saved to the layout file configured in the ActiveX control properties.

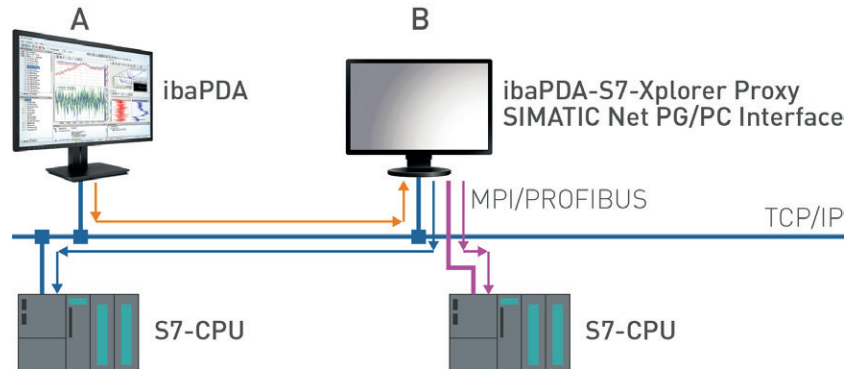
The following is an example of a VBScript in SIMATIC WinCC that can be programmed as a Mouse Click Event using a button, for example. In this case, "ibaPDAActiveXExample" is the object name of the *ibaPDA* ActiveX control in SIMATIC WinCC



The applied method *print* is called by *objControl* in the last line.

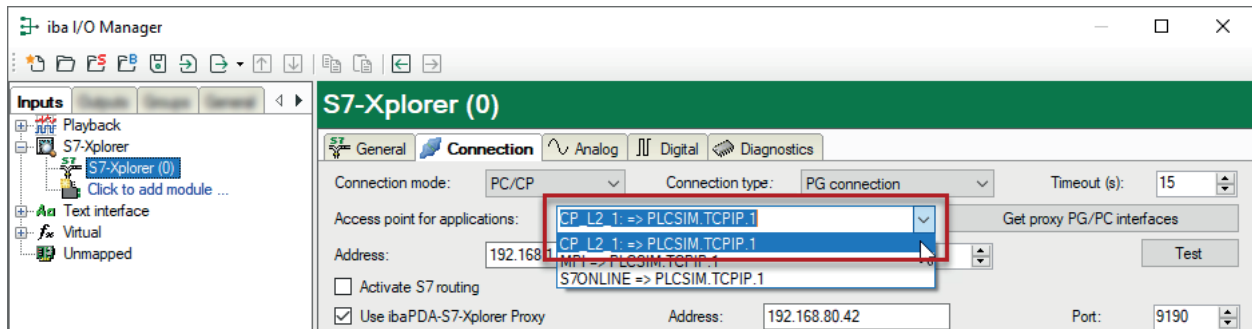
5 ibaPDA-S7-Xplorer Proxy for PC/CP interface

You can also use the *ibaPDA-S7-Xplorer Proxy* when STEP 7 or SIMATIC Net is not installed on the *ibaPDA* computer (computer A), but the CPU still must be accessed via an *S7-Xplorer* module with connection mode PC/CP.



For this purpose, install and start the *ibaPDA-S7-Xplorer Proxy* on the computer with the STEP 7/ SIMATIC NET installation (computer B).

Unlike the configuration with S7-PLCSIM, you must set a real interface (e.g. the network interface card) as access point for applications. Install the access point on computer B.



S7-CPU can be accessed via all installed access points.

6 The I/O configuration file

All configuration data is stored in an XML file.

The *ibaPDA* system uses three different configuration files for I/Os, data storage and display (layout).


For the I/O configuration, *ibaPDA* always uses the *CurrentIoConfig.io* file. The file is stored in a sub-folder of the *ProgramData* folder (path: `c:\ProgramData\iba\ibaPDA\...`).

This file is loaded with each start of the system. The file is updated with each configuration change and when exiting the I/O Manager by pressing the <OK> button. Thus, the current configuration is always stored in this file without the need for explicit saving. Each time a new configuration is applied, the previous one is automatically saved in the `...\Backup` folder. The last ten configurations are always available in the backup folder.


If the most recently used configuration file is not found, the server automatically loads the most recent backup configuration.

If you have applied a valid and functional configuration, it is recommended to also store the configuration file under another name in a protected folder or disk of your choice in order to avoid unintended modifications or loss of configuration data.

6.1 Saving an I/O configuration

1. Open the I/O Manager after you have made sure that the current configuration is correct.
2. Click the button  (save configuration) in the toolbar of the I/O Manager.
3. In the dialog, select the folder where you want to store the file. We recommend choosing the path `c:\ProgramData\iba\ibaPDA`, as this is the default path of the I/O configuration files. Of course, however, you may also choose any other folder.
4. Enter a file name (prefix).
5. Choose the file name extension (default: .io)
 - .io: A simple XML file will be saved.
 - .zip: The XML file will be saved in ZIP file.
6. Click on <Save>.

6.2 Loading an existing I/O configuration

1. Open the I/O manager.
2. Click the button  (open) in the toolbar of the I/O Manager.
3. Using the browser, select the configuration file you would like to use. Usually, you will find the configuration files in the folder `c:\ProgramData\iba\ibaPDA\Server`. The browser shows only files with the extension .io.
The *CurrentIoConfig.io* file contains the latest configuration.

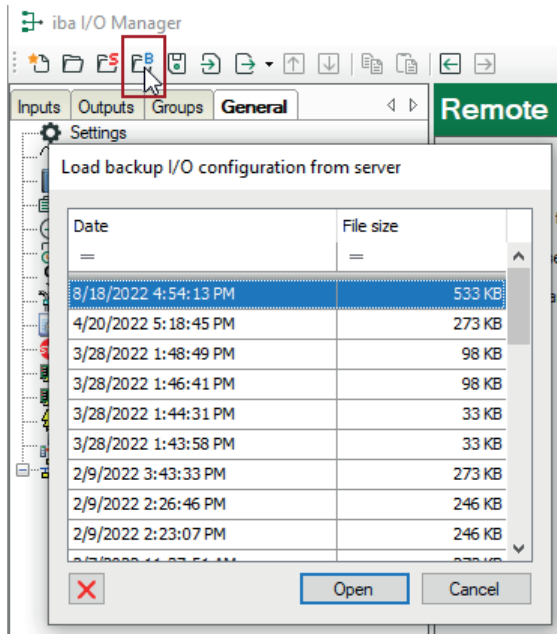
4. Click on <Open>. The configuration will be loaded and verified with the devices connected and the license information in the dongle. If no conflict occurs, the configuration is considered as valid and the acquisition will be started.

Tip

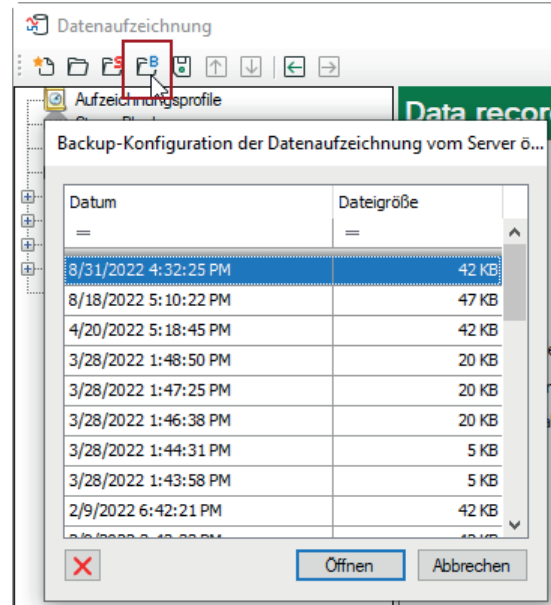
Alternatively, you can store the current configuration in a project. For more information, please refer to the "ibaPDA project management" chapter.

6.3 Loading a backup configuration file

Each time a new I/O configuration or datastore configuration is applied, *ibaPDA* automatically creates a backup of the previous configuration. These backups are stored in the path `%ProgramData%\iba\ibaPDA\Backup` on the *ibaPDA* server computer. In the toolbar of the I/O Manager or data storage manager there is a dedicated button which provides direct access to the backups.



Open backup in I/O Manager



Open backup in datastore manager

If you click on this button, a dialog window will open, which shows a list of the available backups with their date and time of creation and file size.

By selecting a row and clicking on <Open> or by doubleclicking on a row you can open the corresponding backup in the I/O Manager or data storage manager respectively.

If you want to apply the configuration from the backup, then click on <Apply> or <OK>.

In the dialog window with the backup list there is a button to remove a selected backup, if needed.

6.4 Exporting an I/O configuration to a text file

The export function generates a text file which contains modules, groups and profiles. If you have configured profile-based modules (e.g., InSpectra modules), then their profiles will also be exported.

You can only obtain a complete I/O configuration, e.g. including the settings in the *General* branch in the I/O Manager, by saving as an XML file (*.io) or using the project save function.

Modules, groups and profiles can also be exported separately.

You can edit all of the exported files and then reimport them into *ibaPDA*.

This step is sometimes necessary if you update from older *ibaPDA* versions (< v6.0/v7.0) to newer versions.

Even in the case of an initial configuration where many signals need to be configured, editing a text file (e.g., in MS Excel) is usually more convenient and efficient than using the I/O Manager dialogs, especially if you already have the necessary signal information in list form.

Simply create an empty sample configuration which consists of one module of each type.

Next, export this configuration into a text file and then open it with a standard text editor or MS Excel (for example).

You can see the structure of the information in the text file. Copy the module parts as often as needed and adjust the module numbers. You can then edit the signals for each module or insert parts of signal lists from your engineering environment via copy and paste.

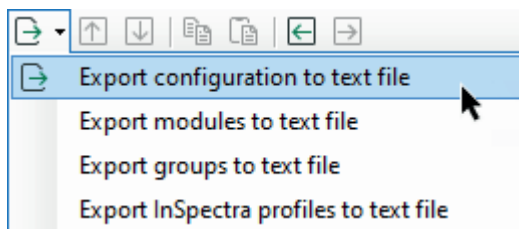
Note

When saving the text file, make sure that a Unicode or UTF-8 data format is selected so that umlauts (mutated vowels) and special characters in signal names and comments are correctly displayed in the I/O manager afterwards.

Once you re-import this text file into *ibaPDA*, the I/O configuration for the modules and signals is complete.

Proceed as follows:

1. Open the I/O manager.
2. Load or create a configuration.
3. Click on the export button and select *Export configuration to text file*.



4. Use the browser to select the desired directory (HDD and folder) in which to store the file, and specify a prefix for the file name. The file suffix *.txt* will be added automatically by *ibaPDA*.
5. Click on <Save>.

Note

The other export options only include the elements specified as part of the overall configuration.

Tip

Export of single modules is possible, too. If there is no need to export an entire I/O configuration, you can export a single module. Simply right-click on the module in the I/O Manager tree and choose *Export* from the context menu.

If you want to export multiple modules from one interface, right-click the interface in the I/O Manager.

Other documentation


More information on exporting InSpectra profiles can be obtained from the *ibaInSpectra* manual.

6.5 Importing a configuration text file

You may import files which have been created by older ibaPDA versions (V5, V6), ibaScope or by other tools (e.g. MS Excel), provided they match the ibaPDA-format.

The files must be simple text files. No XML files!

In case of ibaScope configuration files you need files with the extension ".setup".

1. Open the I/O manager.
2. Click the button  (Import)
3. By using the browser, select the corresponding configuration text file from its storage location (local drive, network drive, floppy disk, CD-ROM etc.). By default, the browser only shows files with the extension .txt. If you are looking for ibaScope files, select the appropriate file type in the lower right corner of the dialog.
4. Click <Open>. The configuration will be loaded and verified with the devices connected and the license information in the dongle. If no conflict occurs, the configuration is considered as valid and the measurement will be started.
5. Store the configuration in the usual way, i.e. as an ibaPDA configuration file, as described above.

Note

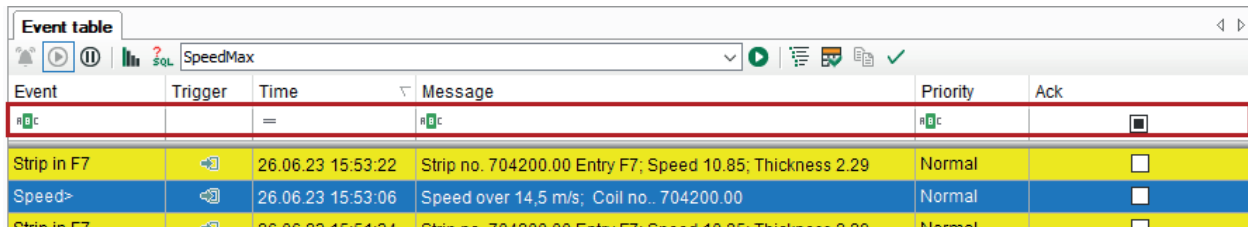


Please note that the module and channel settings are taken over from ibaPDA-V5, however, not the archive profiles! The new data storage profiles should be defined after the import is done and the signals should be assigned to the new profiles.

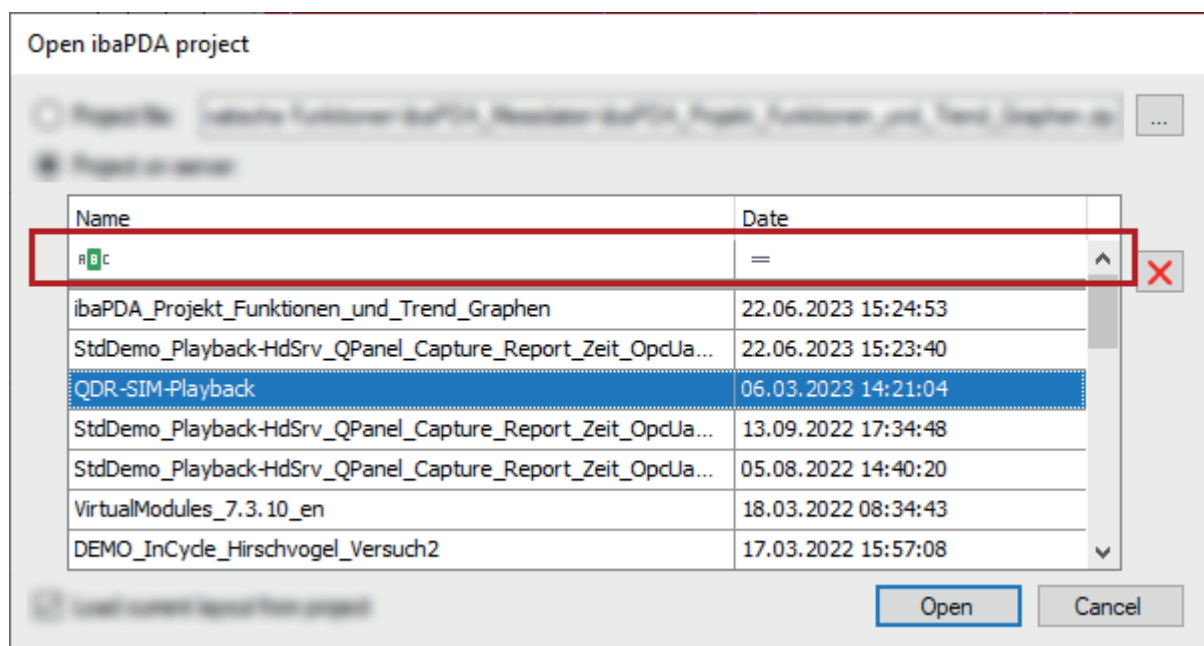
7 Advanced filter in tables

Many tables used in *ibaPDA* provide a filter function. Right below the column headers there is a filter row.

Tables providing such a filter function are for example the HD Event table or the table listing the *ibaPDA* projects in the *Open ibaPDA project* dialog.



Event	Trigger	Time	Message	Priority	Ack
RBC	=			RBC	<input type="checkbox"/>
Strip in F7		26.06.23 15:53:22	Strip no. 704200.00 Entry F7; Speed 10.85; Thickness 2.29	Normal	<input type="checkbox"/>
Speed>		26.06.23 15:53:06	Speed over 14,5 m/s; Coil no.. 704200.00	Normal	<input type="checkbox"/>
Strip in F7		26.06.23 15:54:34	Strip no. 704200.00 Entry F7; Speed 10.85; Thickness 2.29	Normal	<input type="checkbox"/>

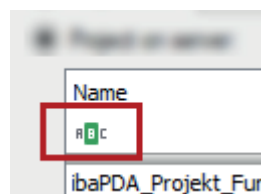


Name	Date
RBC	=
ibaPDA_Projekt_Funktionen_und_Trend_Graphen	22.06.2023 15:24:53
StdDemo_Playback-HdSrv_QPanel_Capture_Report_Zeit_OpcJa...	22.06.2023 15:23:40
QDR-SIM-Playback	06.03.2023 14:21:04
StdDemo_Playback-HdSrv_QPanel_Capture_Report_Zeit_OpcJa...	13.09.2022 17:34:48
StdDemo_Playback-HdSrv_QPanel_Capture_Report_Zeit_OpcJa...	05.08.2022 14:40:20
VirtualModules_7.3.10_en	18.03.2022 08:34:43
DEMO_InCycle_Hirschvogel_Versuch2	17.03.2022 15:57:08

In some of the tables you may immediately see the filter icons, in others you first have to click in the filter row, in order to see the filter icons. Both simple and advanced filter functions are provided.







7.1 Simple filter function

Click in a column of the filter row and you will see the icon of the current filter method.



The default method is *Contains* on columns with text as content and *Equals* on columns with numerical content. If the method has already been altered by a user, the latest selected method is shown.

Clicking on the icon displays a list of the available methods for this column.

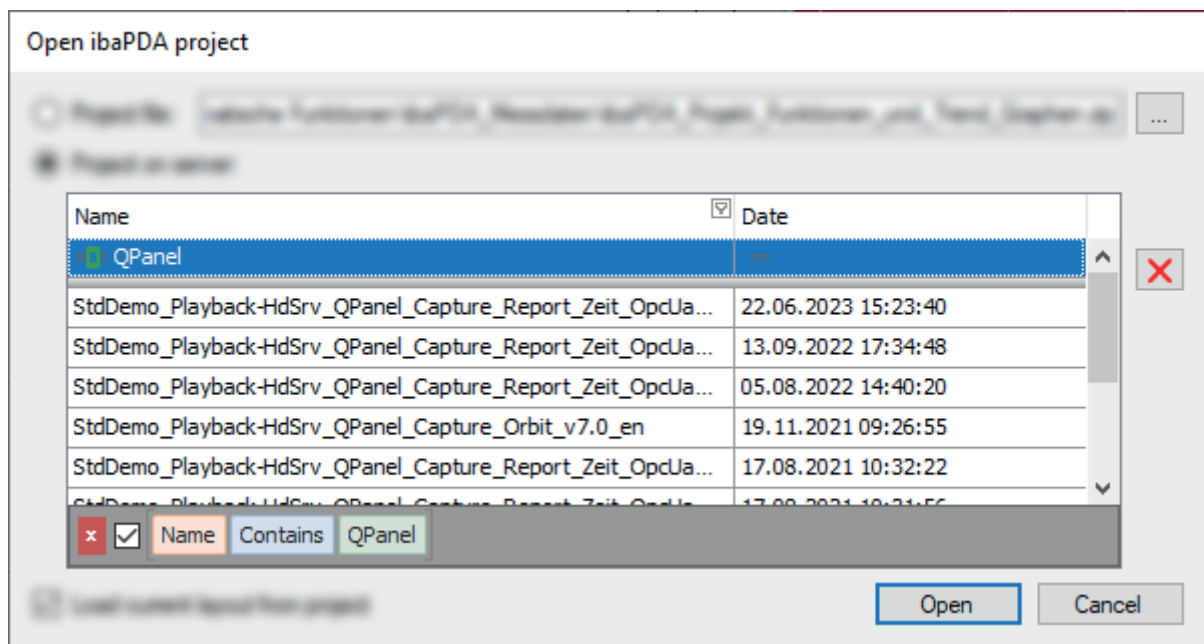
- = Equals
- ≠ Does not equal
-  Contains
-  Does not contain
-  Is like
-  Is not like
-  Begins with
-  Ends with
- > Is greater than
- ≥ Is greater than or equal to
- < Is less than
- ≤ Is less than or equal to

Each method leads to a different filter behavior.

If another filter method suits more your needs, just select it here and enter the filter pattern right into the filter row.

After you have entered a filter pattern the filter is also visualized on the bottom of the table in a human-readable way, showing three fields:

Column name (here *Name*) – filter method (here *Contains*) – filter pattern (here *QPanel*)



The table then only shows the rows which match the filter pattern.

By using the little checkbox you can disable and enable the filter. With the red delete button you remove the filter finally.

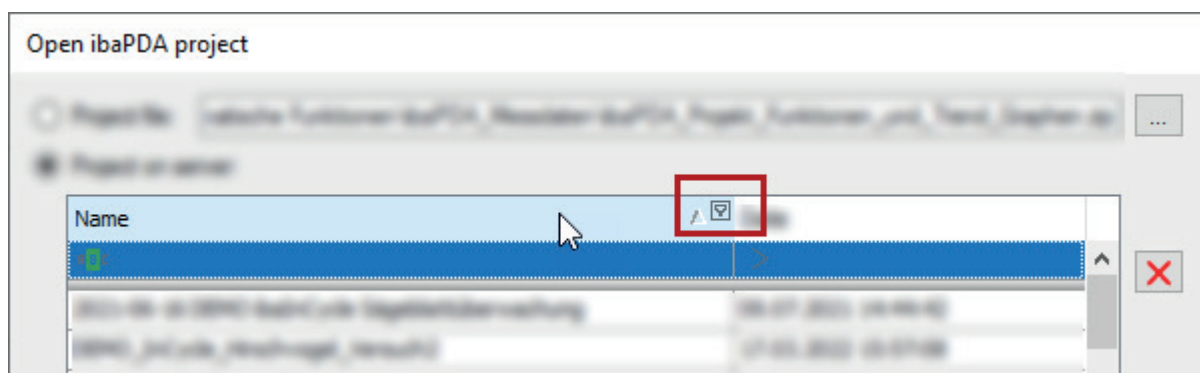
If you filter on multiple columns at the same time, the filters are shown as a logical AND combination.

Name	Date
Qp	18.08.2021 00:00:00
StdDemo_Playback-HdSrv_QPanel_Capture_Report_Zeit_OpcUa_S...	22.06.2023 15:23:40
StdDemo_Playback-HdSrv_QPanel_Capture_Report_Zeit_OpcUa_S...	13.09.2022 17:34:48
StdDemo_Playback-HdSrv_QPanel_Capture_Report_Zeit_OpcUa_S...	05.08.2022 14:40:20
StdDemo_Playback-HdSrv_QPanel_Capture_Orbit_v7.0_en	19.11.2021 09:26:55

☐ ☒ Name Contains Qp And Date > 18.08.2021 00:00:00

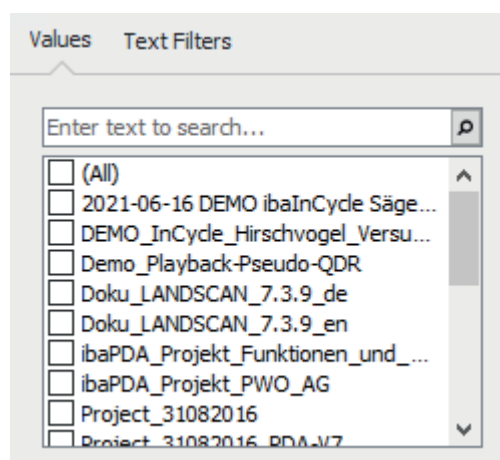
7.2 Advanced filter functions

If you move the mouse cursor over a column header a little filter icon appears at the right end of the column header.



This filter icon, by the way, is always displayed as soon as a filter is enabled.

Clicking on the filter icon opens the advanced filter dialog.

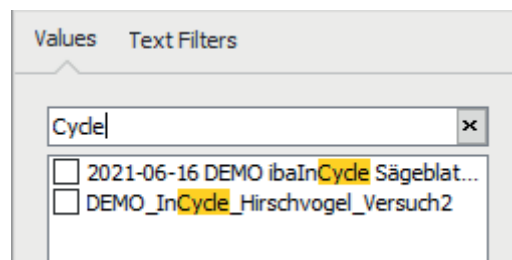


The advanced filter has two tabs:

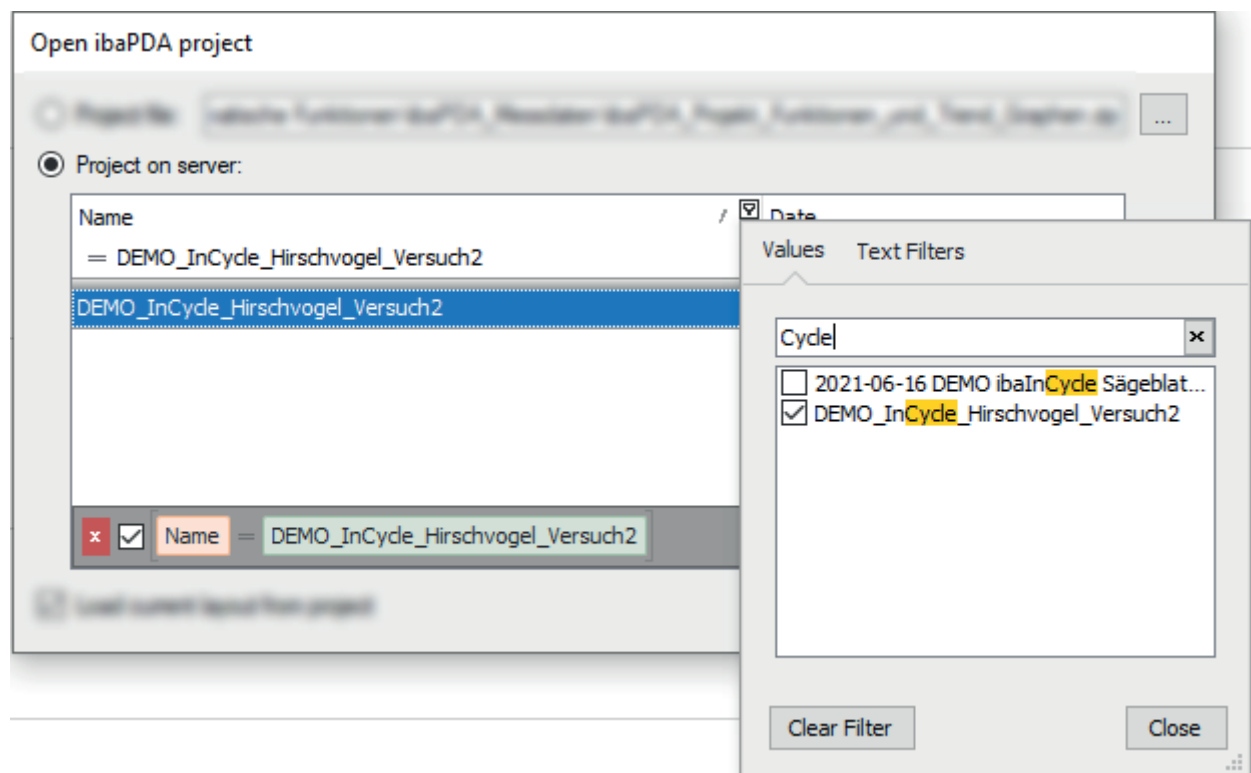
- *Values*, to search for a value or to select one or more values (similar to MS Excel)
- *Advanced filters*, to create a filter based on the value type
In the example above, the value in the column *Name*, is a text and therefore the second tab is *Text Filters*. There would be a *Date Filters* tab if the value is a date and a *Numeric Filters* tab if the value is a number.

Values

The *Values* tab contains a list of all values in the selected column. Above the list, there is a field where you can enter a search pattern. Rows which don't match the pattern are removed from the table. The search pattern is highlighted yellow in every matching row.



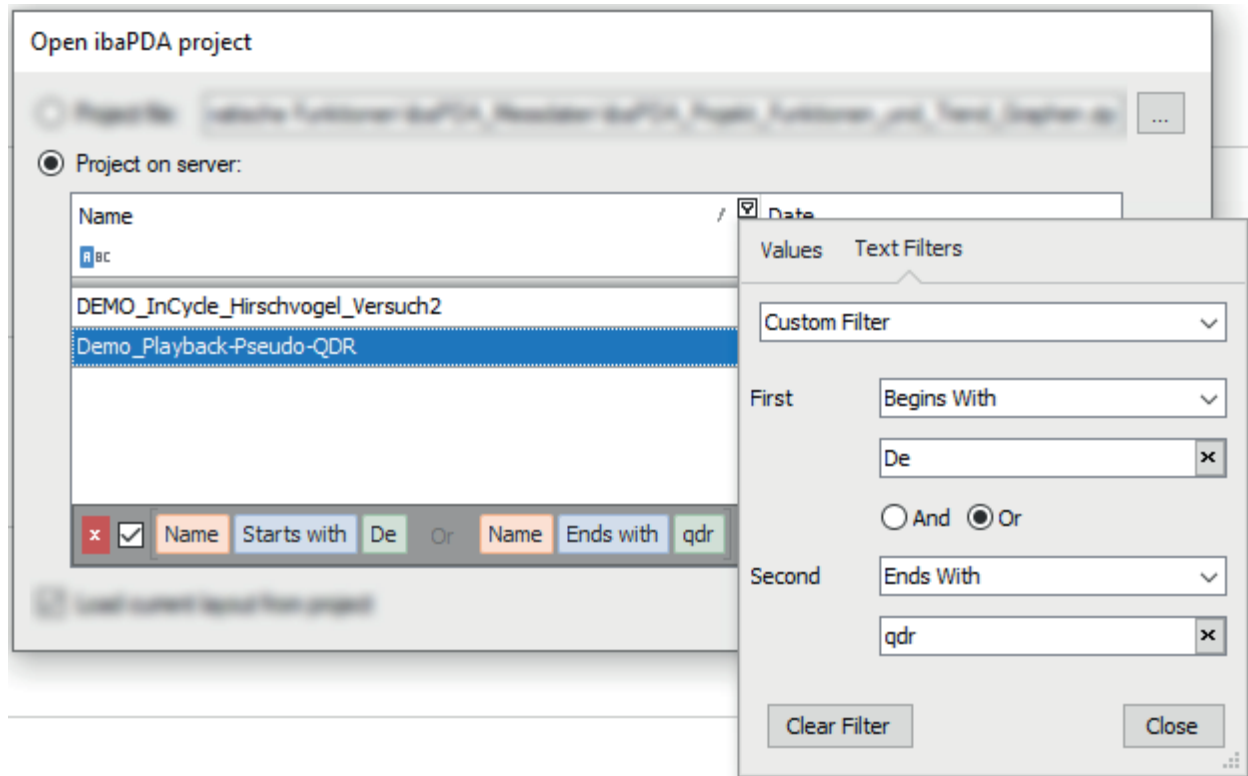
If you select one or multiple search results by a checkmark, only these rows will be displayed in the table.



Advanced filters

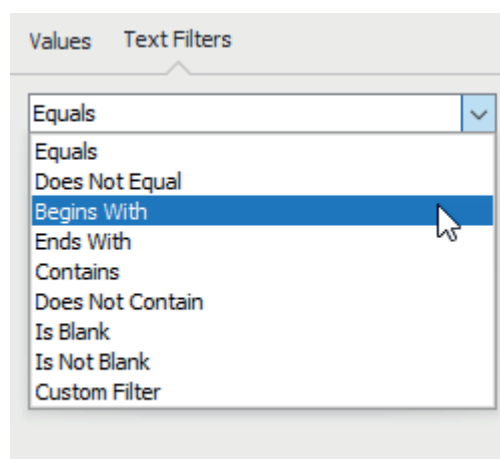
The filter options of the advanced filters depend on the column's value type. A column with numeric value, for example, has different filter options than a column with date values.

Moreover, each filter type provides a custom filter option. In case the standard filters are not sufficient for your needs you can use the custom filter option to create additional filters.



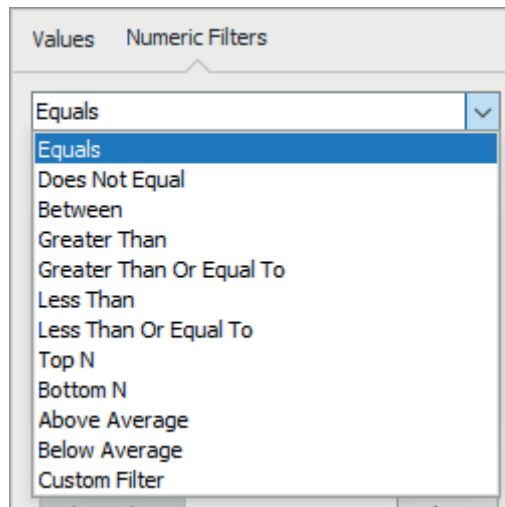
Text filters

Special text filters can be used in columns with text values.



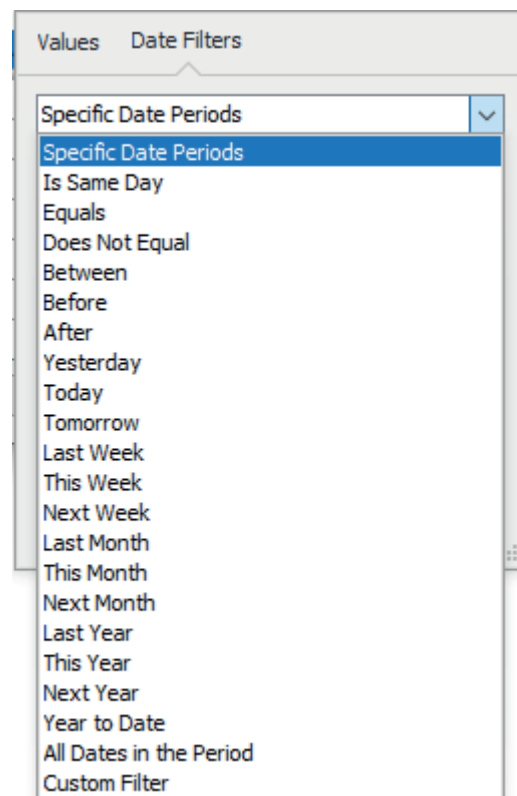
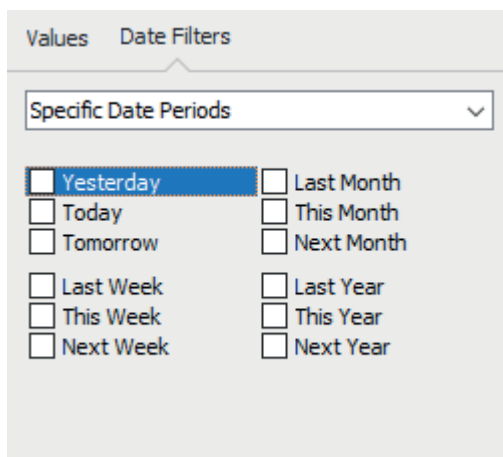
Numeric filters

Numeric filters are only available in columns with numbers.



Date filters

Plenty of filters for time ranges and days are available in columns with date values.



7.3 Search function

Beside the filters you can also use a general search function.

Click into the table and then press <CTRL>+<F>.

A search row will open on top of the table where you can enter a search pattern.

The screenshot shows a dialog box titled "Open ibaPDA project". It has two radio buttons at the top: "Project file" (unselected) and "Project in server" (selected). Below the radio buttons is a list of project names with a search icon on the right. A search bar is open, showing the text "SIM" and a "Find" button. Below the search bar is a table with two columns: "Name" and "Date". The table has one row with the text "QDR-SIM-Playback" in the "Name" column and "06.03.2023 14:21:04" in the "Date" column. At the bottom of the dialog box are "Open" and "Cancel" buttons.

Name	Date
QDR-SIM-Playback	06.03.2023 14:21:04

8 Uninstalling ibaPDA

In case you want to uninstall *ibaPDA*, you have different options as follows:

1. Make a right mouseclick on *ibaPDA* Client in the Windows start menu under *ibaPDA* and select *Uninstall* (you will then be transferred to the Windows control panel where you can find the *ibaPDA* installation).
2. Look for *ibaPDA* entry in the Windows settings under Apps and start uninstalling there.

A dialog for uninstalling opens, providing the option to remove all user data along with the program.

You should not enable this option unless you want to remove *ibaPDA* completely and don't want to use it again.

If you enable the option *Delete all ibaPDA user data* the following items will be deleted:

1. All server data in `%programdata%\iba\ibaPDA`. This folder contains:
 - I/O configuration
 - Data store configuration
 - User management configuration
 - Server access configuration
 - Backups of the configurations
 - Saved projects
 - Address books
 - Certificates
 - Log files
 - File buffer
 - Plugins
2. All user data in `%localappdata%\iba\ibaPDA`. This applies to all users of the system. This folder contains:
 - Layouts
 - Backups of layouts
 - Log files
 - Plugins
 - Cached address books
 - Last connected servers
 - Last main window status

3. The user registry key `HKCU\Software\iba\ibaPDA`. This applies to all users of the system. The registry key contains:
 - Preferences
 - Position and size of the *ibaPDA* client window
 - Client settings
4. *ibaPDA* device driver will be uninstalled.

Data files will not be removed.

9 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