



ibaDatCoordinator

Tool for automated data management

Manual
Issue 4.1

Measurement Systems for Industry and Energy
www.iba-ag.com

Manufacturer

iba AG
Gebhardtstrasse 10-20
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 2026, 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 and can be found in the iba help center docs.iba-ag.com.

Version	Date	Revision	Author	Version SW
4.1	06-2026	Redesigned software interface, updated terminology for jobs and tasks, new "Add task" dialog, new job and task types, email configuration, configuration validation, and other improvements	nm	4.1.0

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	7
1.1	Target group and previous knowledge	7
1.2	Notations	7
1.3	Used symbols.....	8
2	About ibaDatCoordinator	9
2.1	Licenses	9
3	Installation and program start	12
3.1	System requirements.....	12
3.2	Installation and update.....	13
3.2.1	Installing and updating ibaDatCoordinator.....	13
3.2.2	Notation for installation via command line	14
3.2.3	ibaDatCoordinator Status app	16
3.3	Start ibaDatCoordinator client.....	18
3.4	Features of the client-server installation	20
4	User interface and configuration	21
4.1	Menu bar	22
4.2	Navigation.....	24
4.2.1	Jobs and tasks	25
4.2.2	Status	26
4.2.3	Logging	29
4.2.4	TCP/IP Watchdog	30
4.2.4.1	Watchdog message.....	31
4.2.5	SNMP server	32
4.2.5.1	Configuration tab.....	33
4.2.5.2	Objects tab	35
4.2.5.3	System-OIDs 1.3.6.1.2.1.1.....	36
4.2.5.4	Diagnostics tab	37
4.2.6	OPC UA server	38
4.2.6.1	OPC UA server - Configuration.....	38
4.2.6.2	OPC UA server - Tags	41
4.2.6.3	Diagnostics tab	41

4.2.7	Transfer server	42
4.2.7.1	Transfer Server – Configuration	42
4.2.7.2	Transfer Server – Diagnostics.....	43
4.2.8	Settings	44
4.2.8.1	Settings – Processing	44
4.2.8.2	Settings – ibaAnalyzer.....	45
4.2.8.3	Settings – Password protection	45
4.2.8.4	Settings – Script task.....	46
4.2.8.5	Settings – Certificates	47
4.2.8.6	Settings – Databases.....	53
4.2.8.7	Settings – Email accounts	54
4.2.8.8	Settings – Global cleanup	55
5	Configuration of jobs and tasks	56
5.1	Overview of job types.....	56
5.1.1	DAT file job.....	57
5.1.2	Scheduled HD job and scheduled job	58
5.1.3	Event job.....	58
5.1.4	Time period job	58
5.1.5	One-time job.....	59
5.1.6	External file job.....	59
5.2	Configuring a job	59
5.2.1	DAT file job – settings	61
5.2.1.1	Reprocessing DAT files	64
5.2.2	Scheduled HD job and scheduled job – settings.....	65
5.2.3	Event job – settings.....	70
5.2.4	Time period job – Settings	73
5.2.4.1	Properties of the time period table	75
5.2.5	One-time job – settings	77
5.2.6	External file job – settings.....	77
5.2.6.1	Convert external file task – settings.....	80
5.3	Overview of task types	80
5.4	Configuring a task	82
5.4.1	General task settings	83
5.4.2	Copy/move/delete task	90

5.4.3	Transfer task	91
5.4.4	Upload task.....	93
5.4.5	Report task	96
5.4.6	Condition task.....	98
5.4.7	Cleanup task	102
5.4.8	Pause task.....	105
5.4.9	Split task	106
5.4.10	Script task	108
5.4.11	HD offline event task	111
5.4.11.1	HD offline event task – settings	112
5.4.11.2	HD offline event task – Configuration tab.....	114
5.4.11.3	HD offline event task – Client options tab	120
5.4.12	HD import task.....	121
5.4.13	Extract to file task	123
5.4.14	Extract to DB task and publish to DB task.....	124
5.4.15	Kafka publish task	125
5.4.15.1	Kafka publish task – Connection tab.....	126
5.4.15.2	Kafka publish task – Target tab	131
5.4.16	OPC UA publish task	135
5.4.17	SNMP publish task.....	136
5.4.18	Time period update task.....	138
5.4.19	Update task.....	139
5.4.20	S7 writer task.....	141
5.4.21	Merge task.....	144
5.4.22	DB sync task.....	146
5.4.22.1	Configuring a DB sync task.....	147
5.5	Starting and stopping jobs	149
6	Application examples and FAQ	150
6.1	Example: Using the merge task for multistation.....	150
6.2	Example: Using info fields for directory/file names.....	154
6.3	Establishing ibaHD-Server connection.....	157
6.4	Configuring service for other users	158

7 Troubleshooting 159

 7.1 Troubleshooting..... 159

8 Glossary 161

9 Support and contact..... 162

1 About this documentation

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

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 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 especially addresses persons who are responsible for analyzing measurement and process data. As the data is provided and processed using other iba products, the following prior knowledge is required or helpful:

- *ibaPDA* (collection and structure of data files).
- *ibaAnalyzer* (analysis files)
- *ibaHD-Server* (for using tasks with *ibaHD-Server* involved)
- Handling databases (for using database-related tasks)
- Script programming (for using script tasks)

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.

2 About ibaDatCoordinator

ibaDatCoordinator is the central component for the automated processing of measurement data and is therefore essential for the use of the iba system in production systems. With *ibaDatCoordinator*, you can automatically process all measurement files generated by *ibaPDA* or other systems (e.g., *ibaQDR*, *ibaLogic*), as well as data from recordings on the *ibaHD-Server*.

Functions and applications

The main task of *ibaDatCoordinator* is the further processing of measurement data in various formats. In addition to DAT files and data from *ibaHD-Server*, external formats can also be processed (paid add-on). Depending on the data source, create a suitable "job."

You can find an overview of the different job types in chapter [➤ Overview of job types](#), page 56.

While you use the job type to determine the data source, you can configure any number of tasks for each job to control the further processing of the data. The exception is the external file job, for which you can only create one task at a time.

You can find an overview of the different task types in chapter [➤ Overview of task types](#), page 80.

2.1 Licenses

You can basically install and use *ibaDatCoordinator* free of charge. Comprehensive basic functionality is available free of charge. For advanced functions, you must purchase the appropriate licenses for each configured task or job.

The following tasks are available free of charge and enable the configuration of extensive, automated processing sequences:

- Tasks for easy processing of DAT files: copy/move/delete, cleanup, upload, update, report, and split task
- General tasks: pause, condition, script and transfer task
- Tasks for processing HD data: HD import and HD offline event task

Purchased licenses are required for the following additional tasks:

- One *ibaDatCoordinator-DB* license per configured task for automated extraction of data (time series and calculated values) in databases (extract to DB task)
- One *ibaDatCoordinator-File-Extract* license per configured task for extracting data into files (extract to file task) and merging multiple DAT files into one DAT file (merge task)
- One license for *ibaDatCoordinator Convert-CSV-to-dat*, *ibaDatCoordinator Convert-DAS-to-dat*, *ibaDatCoordinator Convert-COMTRADE-to-dat*, or *ibaDatCoordinator Convert-Parquet-to-dat* per configured job for the automated conversion of files from external formats to DAT files, depending on the external format used (DAT file job)

The functionality of *ibaAnalyzer-E-Dat* is integrated into these licenses. A separate license does not need to be purchased.

- One *ibaDatCoordinator-Publish* license per configured task for publishing calculated values via one of the publish tasks (OPC UA, SNMP, or Kafka)

This license is sufficient for extraction in databases, as long as only individual values (info assignments and calculated columns in the *ibaAnalyzer* data extractor) are extracted and no time series.

- A license for *ibaDatCoordinator Update Data Task* or *ibaDatCoordinator S7 Writer* for special tasks for subsequent updating of data files (update task) and for writing to SIMATIC S7 data blocks (S7 writer task).

The number of configurable tasks is not limited for these licenses.

For some of the task types mentioned, bundles of 10 are also available.

License information

Order no.	Product name	Description
34.010550	ibaDatCoordinator	<i>ibaDatCoordinator</i> basic license
34.010510	ibaDatCoordinator-DB	License extension for automatic extraction of data in databases (per task)
34.010520	ibaDatCoordinator-DB-10	License extension for automatic extraction of data in databases (10 tasks)
34.010511	ibaDatCoordinator-File-Extract	License extension for automatic extraction of data in DAT files or other file formats such as CSV, COMTRADE, Parquet, MATLAB, TDMS (per task)
34.010521	ibaDatCoordinator-File-Extract-10	License extension (bundle) for automatic extraction of data in DAT files or other file formats (10 tasks)
34.010512	ibaDatCoordinator Convert-CSV-to-dat	License extension for automatic conversion of CSV files to DAT files (per job)
34.010513	ibaDatCoordinator Convert-DAS-to-dat	License extension for automatic conversion of DAS files from Danieli FDA systems to DAT files (per job) Only available for the 32-bit version of <i>ibaAnalyzer</i> .
34.010514	ibaDatCoordinator Convert-COMTRADE-to-dat	License extension for automatic conversion of COMTRADE files to DAT files (per job)
34.010516	ibaDatCoordinator Convert-PARQUET-to-dat	License extension for automatic conversion of Parquet files to DAT files (per job)
34.010515	ibaDatCoordinator-Publish	License extension for publishing calculated values using OPC UA, SNMP, Apache Kafka, SQL database, or time period updates (per task)
34.010525	ibaDatCoordinator-Publish-10	License extension for publishing calculated values using OPC UA, SNMP, Apache Kafka, SQL database, or time period updates (10 tasks)

Order no.	Product name	Description
34.010552	ibaDatCoordinator-Update Data Task	License extension for the function of subsequent entries in data files
34.010556	ibaDatCoordinator S7 Writer	License extension for writing values from a data file to data blocks (DB) of an S7 PLC

License extensions for ibaAnalyzer

33.010000	ibaAnalyzer	Offline analysis tool
33.010003	ibaAnalyzer-DB-Read	Read data from SQL databases
33.010445	ibaAnalyzer-E-Dat	Offline analysis for external data formats

3 Installation and program start

Here you will find information about system requirements, installation, and starting the software.

3.1 System requirements

The following system requirements are necessary for using *ibaDatCoordinator*:

Hardware

- At least PC, Multicore CPU 2 GHz, 2048 MB RAM, 100 GB HDD

Software

You can open the version history via the *Help - Version history* menu in the *ibaDatCoordinator* client. In the *Version compatibility* tab, you will find all relevant requirements regarding the operating system and other iba components for the respective version of *ibaDatCoordinator*.

Please note the following requirements for *ibaDatCoordinator* v4.0.0 or higher:

- Operating system Windows 10 (x86/x64), Windows 11 (x64), Windows Server 2016 (x64), Windows Server 2019 (x64), Windows Server 2022 (x64), Windows Server 2025 (x64)
- .NET framework v4.8
- *ibaAnalyzer* v8.3.0 or higher
- *ibaHD-Server* v3.2.0 or higher
- *ibaPDA* v7.0.0 or higher

Note



A connection from an *ibaDatCoordinator* server v4.0.0 is only possible with an *ibaDatCoordinator* client v4.0.0 or higher.

In addition, you can only use an *ibaDatCoordinator* client v4.0.0 for the configuration of an *ibaDatCoordinator* server v4.0.0 or higher.

Compatibility with other iba products

ibaAnalyzer is a central component for the function of *ibaDatCoordinator*. Many tasks, such as extraction and report generation, are performed in the background by *ibaAnalyzer*. For this reason, *ibaAnalyzer* must also be installed. iba always recommends using the latest available version of *ibaAnalyzer*.

In some cases, certain versions of other software products such as *ibaPDA* or *ibaHD-Server* are also required. Due to these cross-dependencies, iba recommends that you always install the latest version of the software products involved.

3.2 Installation and update

If you have a ZIP file of the new software version available (e.g. after a download), unzip it into any (temporary) directory. You will find the file for *ibaDatCoordinator* on the "iba Software & Manuals" data medium in the [01_iba_Software\ibaDatCoordinator](#) directory.

You have the following options for installing *ibaDatCoordinator*:

■ Installation with installation wizard

See [🔗 Installing and updating ibaDataCoordinator, page 13](#).

■ Installation via command line

This approach is helpful for central software administration or when using deployment systems. You can control how the installation is performed and which components are installed using the command line switches that are set after calling the installation program.

You can find an overview in [🔗 Notation for installation via command line, page 14](#).

If an older software version is already installed, you can simply install the new version. The older version of the program is automatically removed after a query and confirmation. Settings and configurations made with the older version are retained.

3.2.1 Installing and updating ibaDataCoordinator

Below you will learn how to install *ibaDatCoordinator* using the installation wizard.

Note



Some options only appear during the initial installation and not during the update. If you uninstall *ibaDatCoordinator* manually before installing the new version, the installation behaves in the same way as the initial installation.

1. Execute the file [ibaDatCoordinatorSetup_vx.y.z.exe](#) and follow the instructions of the installation wizard.
2. If necessary, check the version history in advance.
To do this, click *View version history* on the welcome page of the wizard.
3. Read and accept the license agreement.
4. Select the installation directory (initial installation only).
5. Select the components to be installed:

- Client and server (standard installation, recommended)

ibaDatCoordinator will be installed and executed as a service. Configuration takes place via the client.

Please also note the information in [🔗 Features of the client-server installation, page 20](#).

- Client only

Select this option if you do not want to run the *ibaDatCoordinator* service on the PC, but only need the client for remote configuration on another system.

- Stand-alone executable file

In this case, *ibaDatCoordinator* does not run as a service, but locally in a separate process as an application. This type of installation is recommended for test purposes.

optional:

- ibaManagementStudio Agent

If an *ibaManagementStudio* server is available in the network, you can manage this installation with *ibaManagementStudio*. Select this option to install the *ibaManagementStudio* agent. If this software is already installed on your computer, you can deactivate this option.

6. Update of "WIBU Codemeter Runtime" (only displayed if required)

If you are using the "WIBU CodeMeter Runtime" software for licensing, the installer will check the currently installed version. If the version is out of date, the installation program will offer to update CodeMeter Runtime. Please note that other iba applications also use the software for licensing and can be stopped and restarted during the update. If you do not want to update CodeMeter Runtime, deselect the option.

7. Select the user account (initial installation only):

- Local system account
- Own user

8. Select whether a desktop shortcut should be created during installation.

9. Once the installation is complete, you can choose whether *ibaDatCoordinator* should start automatically after closing the wizard.

→ When installing as a service, the *ibaDatCoordinator* Status app is also installed for the installation and diagnosis of the service. The corresponding status icon appears in the Windows taskbar, see ↗ *ibaDatCoordinator* Status app, page 16.

Note



During installation as a service, the system automatically checks whether the resource allocation in the operating system is optimally set in the Windows registry file. If tasks with high resource requirements are to be performed, optimization is recommended. You can perform this process manually later via the *ibaDatCoordinator* server status.

3.2.2 Notation for installation via command line

Below is an overview of the command line switches for installing *ibaDatCoordinator* via the command line.

Example

```
ibaDatCoordinatorSetup_v3.1.0.exe /COMPONENTS= ibaDatCoordinator\server /VERYSILENT /SUPPRESSMSGBOXES /TASKS=nodesktopicon /PORT=8800
```

/HELP	This switch shows a message box containing all supported command line parameters.
/SILENT	This switch installs the software in "silent mode", i.e., interaction is not required during installation. All installer pages are skipped except the Installing page showing the installation progress. The installer runs without specifying additional parameters, as if <Next> were always pressed in the dialog.
/VERYSILENT	Works like "/SILENT" but nothing is shown.
/SUPPRESSMSGBOXES	Suppresses all pop-up messages. Use this option especially in combination with /VERYSILENT to guarantee a completely GUI free installation.
/TASKS=	This switch allows you to specify whether a desktop icon should be created during installation. Examples: /TASKS=desktopicon /TASKS=nodesktopicon
/DIR=	Use this switch to specify the installation folder, i.e. the program folder. Example: /DIR="C:\Programme (x86)\iba\ibaProduct"
/LANG=	This switch sets the language of the installer (by default, the language of the operating system is used). Possible options: de, en, es, fr, it, ja, pt, ru, zh. Examples: /LANG=en /LANG=de
/COMPONENTS=	Insert the names of the components to be installed directly after the equal sign in a comma-separated list. Only the components specified in the list are installed. Possible options: ibadatcoordinator\server, ibadatcoordinator\client, ibadatcoordinator\standalone, ibamanagementstudio Examples: /COMPONENTS=ibadatcoordinator\client /COMPONENTS=ibadatcoordinator\server,ibamanagementstudioagent
/PORT=	Use this switch to select the TCP port via which the <i>ibaDatCoordinator</i> service communicates with the client. The port can be set later in the <i>ibaDatCoordinator</i> Status application. This switch can only be used in conjunction with the installation of Client/Server. Default setting: 8800 or setting of the previous installation.

3.2.3 ibaDatCoordinator Status app

The *ibaDatCoordinator* Status app is only available if the *ibaDatCoordinator* service is installed on the PC.

You can open the status app in several ways:

- Double-click the *ibaDatCoordinator* icon in the taskbar
- Right-click the *ibaDatCoordinator* icon in the taskbar and select *Restore* in the context menu.
- Via the Windows Start menu: *ibaDatCoordinator Server Status*

When the status app is running, the corresponding status icon appears in the Windows taskbar. The icon indicates the following statuses:



ibaDatCoordinator service is running

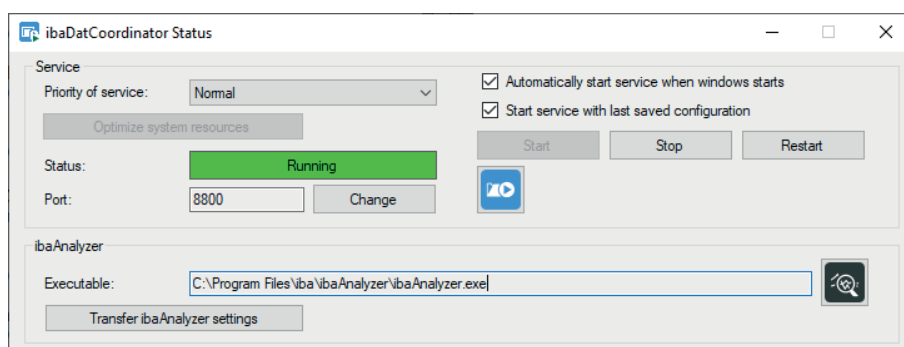


ibaDatCoordinator service stopped

In the context menu of the status icon in the taskbar, you will find additional options to start, stop, or completely exit the status app.

Configuration options in the status app

The *ibaDatCoordinator* Status app offers functions for monitoring and configuring the *ibaDatCoordinator* service.



Priority of service

You can define different priorities for the *ibaDatCoordinator* service. The default value is "Normal".

If you choose a higher priority the process is given more CPU-time. Lowering the priority makes more CPU time available for other processes. Leave the setting at "Normal" if you are only using one or a few jobs and there is sufficient time to complete the tasks.

If less time is available and you want to avoid other processes interrupting the execution of the jobs, select a higher priority. If the priority of service is changed, the *ibaDatCoordinator* service will need to be restarted.

Automatically start service when windows starts

If this option is selected, the *ibaDatCoordinator* Status application starts automatically when you log on to Windows.

Start the service using the most recently saved configuration

When enabled, *ibaDatCoordinator* starts with the most recently saved configuration. If this option is not enabled, the client starts with an empty configuration.

<Optimize system resources>

If you click on this button, *ibaDatCoordinator* checks the resource allocation in the operating system and corrects the settings in the Windows registry file if necessary.

If this optimization is not carried out, the number of simultaneously available *ibaAnalyzer* instances is reduced to a maximum of 6.

When you perform this action, *ibaDatCoordinator* saves a backup of the affected registry entries in the program folder.

Status

In the status display, you can see if the service is running or not.

<Start>, <Stop> and <Restart>

With these buttons, you can start and restart the *ibaDatCoordinator* service.

Stopping all jobs and the service can take a long time, especially with extensive configurations. If necessary, all processes involved can be manually forced to stop immediately using the <Force stop> button.

Port

The port number through which the *ibaDatCoordinator* service communicates with the client. If necessary, you can adjust the port number using the <Change> button. The default port is set to 8800.

Note

When installing in existing networks, the port number might be used by other devices or processes, or the port number might be blocked by the firewall. In such cases, the operation of the *ibaDatCoordinator* service, e.g. starting and stopping jobs, is not possible (no connection to the service, red header bar in the client). In this case, change the port number so that it can be used only by *ibaDatCoordinator* and/or is approved in the firewall.

Open ibaDatCoordinator client

To open the *ibaDatCoordinator* client directly from the status app, click on . The client then automatically connects to the correct server.

ibaAnalyzer – Executable

Path to the [ibaAnalyzer.exe](#) application (read-only)

If no *ibaAnalyzer* installation was found, this is also displayed. To test the path and open *ibaAnalyzer*, click on the button with the *ibaAnalyzer* icon.

<Transfer ibaAnalyzer settings>

Settings of *ibaAnalyzer*, like default settings, macros and filters are usually stored in relation to the user. If *ibaDatCoordinator* is installed as service, the user accounts of *ibaDatCoordinator* client and service can be different. If *ibaAnalyzer*-settings for the tasks that are to be fulfilled have been finalized and saved under the client user, these settings are not available to the service for processing tasks as the service starts *ibaAnalyzer* under a different user account. Executing a macro, for example, would not work.

Click the button to transfer the *ibaAnalyzer* settings from the local user account to the system account so that they are available for *ibaDatCoordinator* or *ibaAnalyzer* to perform the analyses.

This applies in particular to the printer settings. It may be necessary to run the service under a different user in order to be able to use network printers.

3.3 Start ibaDatCoordinator client

To start the *ibaDatCoordinator* client and connect to the correct server, proceed as follows.

Note

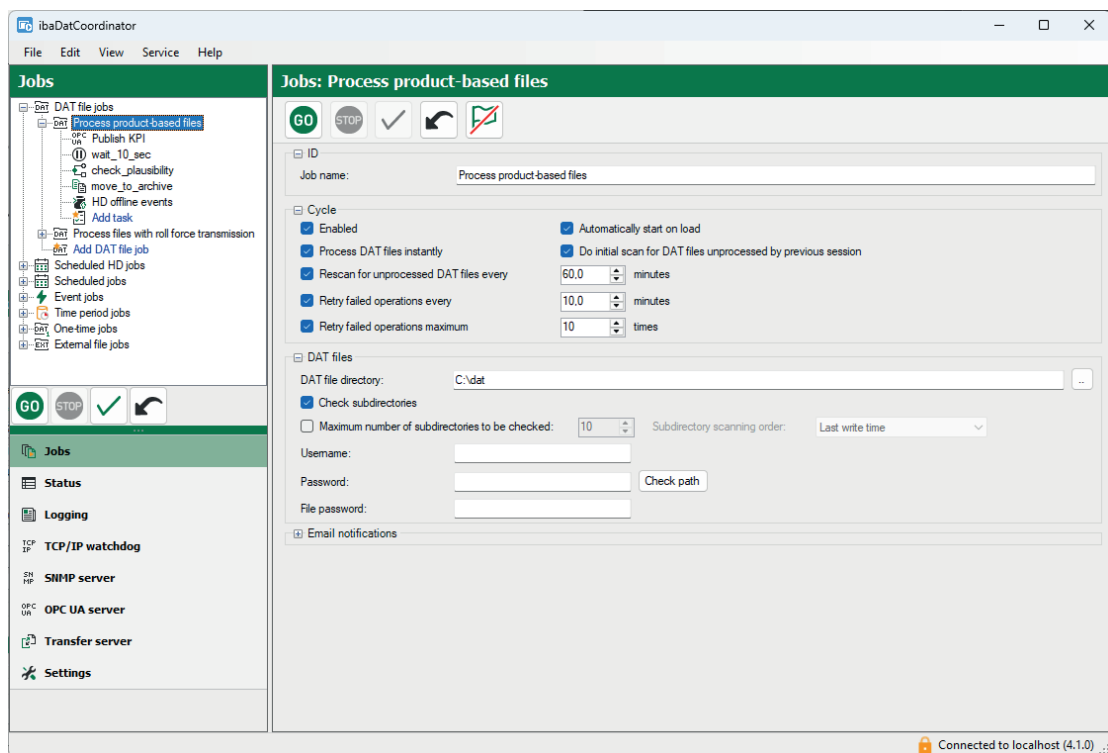


When installing *ibaDatCoordinator* as a standalone executable file, it is not necessary to establish a connection to a server.

1. Open the *ibaDatCoordinator* client via the corresponding path in Windows or via the *ibaDatCoordinator* status app.

See ↗ *ibaDatCoordinator Status app*, page 16.

- The client automatically attempts to connect to the last connected server. The currently connected server is displayed in the status bar of the application (bottom right).



The color of the window title indicates the connection status to the server:

- Green: The client is connected to the server and processing is active.
- Orange: The client is connected to the server and is in read-only mode.
- Red: The client is not connected to any server.

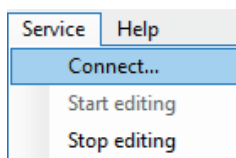
See [➤ User interface and configuration, page 21](#).

Note

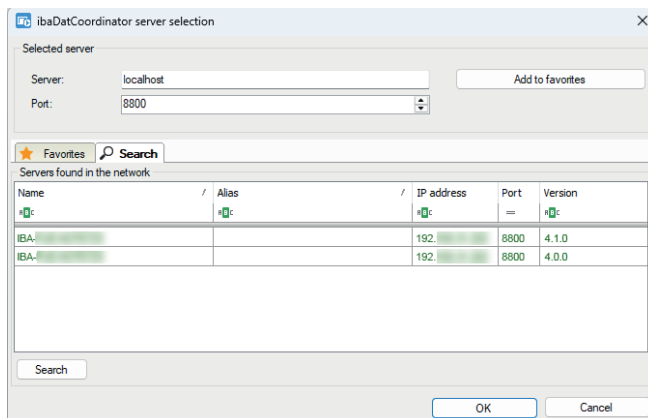


Communication is normally encrypted. The configuration is only transferred unencrypted if the *ibaDatCoordinator* service version does not support this.

2. To connect the client to another server, in the menu click on the *Service – Connect*.



3. In the following dialog, select the server that is to be connected to the client.



Use the <Search> button to start or stop searching for active *ibaDatCoordinator* servers on the computer and in the network.

If the server is not found in the network, you have the option of entering the IP address (*Server*). and port manually. The standard port is 8800.

4. Confirm your selection with <OK>.
- The client attempts to connect to the selected server via the specified port.
5. If you exit the dialog with <Cancel>, the client remains connected to the previously connected server.

3.4 Features of the client-server installation

Please note the following specific features when using client-server installations with regard to access by multiple clients to a server, running the service on a remote system, and the use of license extensions.

Multiple client connections

It is possible for multiple clients to connect to the same *ibaDatCoordinator* server. This means that several clients can change the jobs and tasks of *ibaDatCoordinator* on the server, but not simultaneously. If a client is connected to the server in edit mode, other clients that attempt to connect receive a message that editing by another client is blocked. In this case, you have the option of disconnecting the other client or opening your own client in read-only mode.

You cannot make any changes to the configuration in read-only mode. The header in the client is orange:



Read-only mode is also active if password protection is activated or you have entered an incorrect password.

You can also switch between edit mode and read-only mode. To do this, click on *Start editing* or *Stop editing* in the menu under Service.



Executing the service on a remote system

If the service is running on a remote system, analysis files, scripts, source and target directories should be located on the remote system. When specifying a local file system path, it should be a local path on the server system. If the path leads to a shared network folder, *ibaDatCoordinator* converts the path to a UNC path.

If you click on the Browse button, where you can specify such paths, a dialog opens that emulates a browse dialog running on the server system. For tasks where you can specify an analysis file (on the remote system or on a network share), a copy of the analysis file is downloaded to the client system. If changes have been made, the file can be manually uploaded to the server.

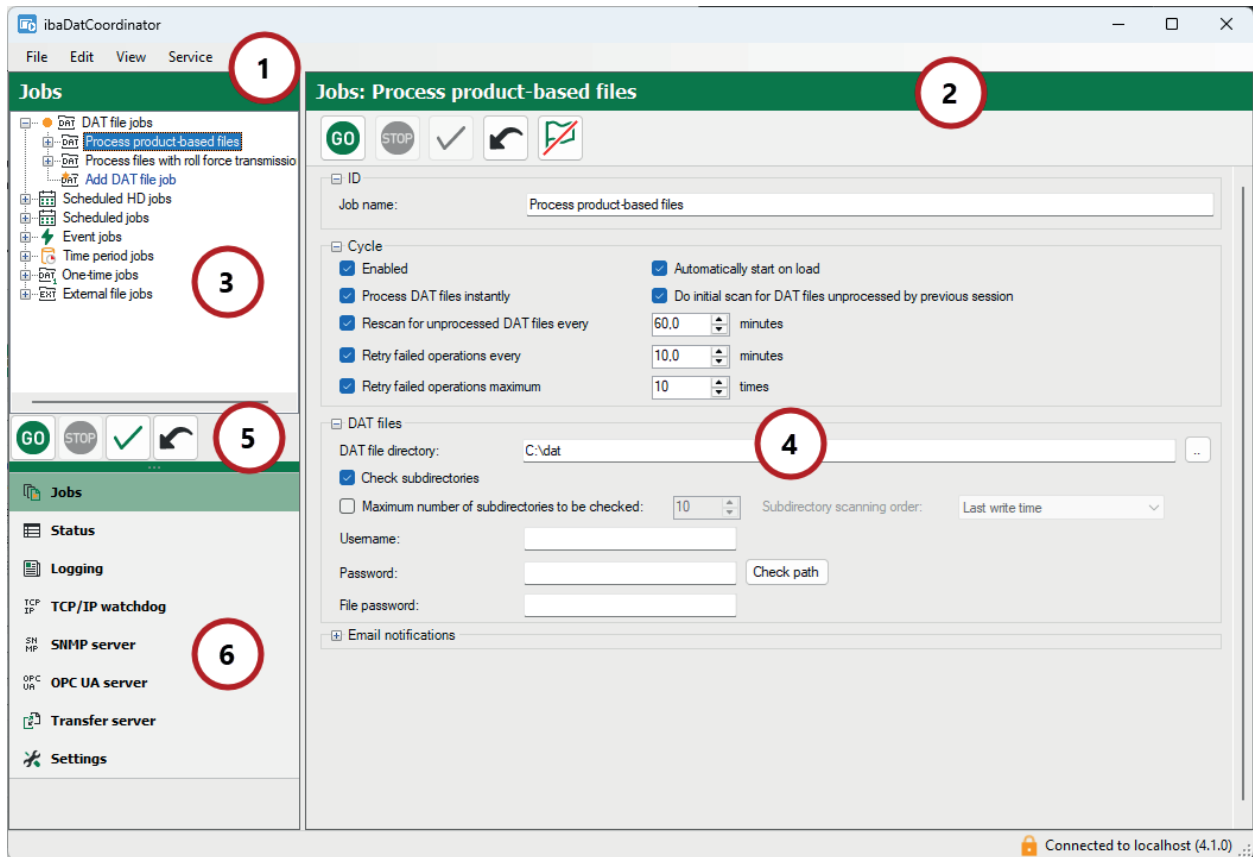
License extensions

If you want to use additional functions that require a license as plugins in *ibaDatCoordinator*, you must install them on the server system. If you attempt to install the plugins on a system on which only the client is installed, you will receive a warning and the installation will be canceled.

When a client connects to a server for the first time, the files needed for the execution of the plugin will be transferred from the server to the client. If the client connects to another server, the already loaded plugin components might not be compatible with the version of the new server to be connected. If this is the case, *ibaDatCoordinator* will have to be restarted so that the plugins can be copied correctly.

4 User interface and configuration

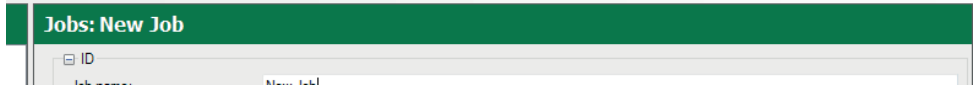
When you open *ibaDatCoordinator*, you will see the main window.

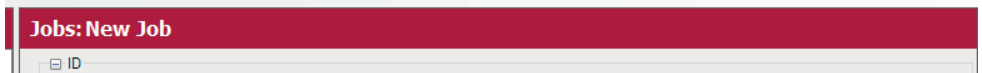


1	Menu bar
2	Header of the window
3	Navigation area, e.g. list of jobs and task
4	Display and navigation area
5	Buttons for starting and stopping (all jobs), as well as for accepting and discarding changed configurations
6	Navigation menu

Connection status of the client

The header of the window shows the connection status of the client to the server:

Color	Meaning
Green	 The client is connected to the server and is in edit mode.

Color	Meaning
Orange	 The client is connected to the server and is in read-only mode. See ↗ Multiple client connections, page 20.
Red	 The client is not connected to any server.

4.1 Menu bar

The menu bar contains the main control and navigation functions.

File Edit View Service Help

File

The *File* menu is mainly used for managing configuration files.

New

Create new *ibaDatCoordinator* project

All the configured jobs are removed and a new project will be created.

Open

Open previously saved *ibaDatCoordinator* project (XML file)

Open from backup

Open saved configuration status

Backups are saved with every change to the configuration. A window with all existing backups opens, from which you can select the desired backup or delete backups.

Save as...

Save project as XML file

Exit

Close the *ibaDatCoordinator* client

If *ibaDatCoordinator* is installed as service, the service and hence all jobs continue running.

If you have made changes to the configuration that have not yet been transferred to the server, a message appears.

Edit

Apply all, Undo all

The *Apply all* action applies all changes (complete configuration) to the server, *Undo all* discards the changes and reloads the last saved server configuration.

You can apply the changes to all jobs, individual jobs, or server settings. See [↗ Jobs and tasks](#), page 25.

Cut, Copy, Paste, Delete

You can apply these actions to jobs as well as to tasks. Some of these actions are not available when the job is running. These options are also available in the context menu of the job tree.

View

The *View* menu contains all functions that are also available in the navigation bar. Also see chapter ↗ *Navigation*, page 24.

Language switching is also possible there.

Service

The *Service* menu is only available if *ibaDatCoordinator* was installed as a service (client server).

Connect...

Connect the *ibaDatCoordinator* client to another server, see ↗ *Start ibaDatCoordinator client*, page 18.

Start editing

Starts the editing mode

You can only select the menu item if the client is in read-only mode. For further information see ↗ *Multiple client connections*, page 20.

Stop editing

Releases the server for processing by another client and sets the client to read-only mode.

Help

In the *Help* menu, you can open the *ibaDatCoordinator* documentation as an HTML file. If you press the <F1> key, the help page for the current view opens.

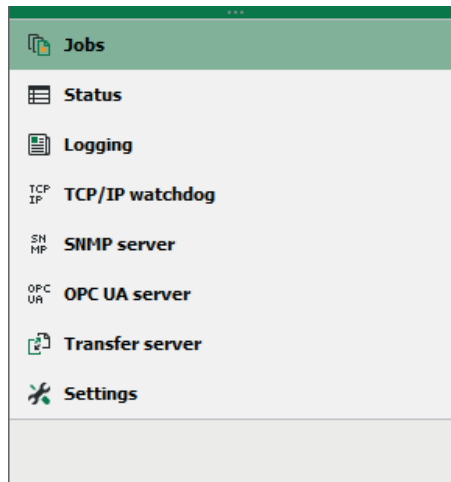
In case of support, you can also create a ZIP file in which all relevant settings and configurations are saved. If *ibaDatCoordinator* was installed as a service, this ZIP file contains two separate ZIP files: `client.zip` and `server.zip`. The latter is generated by the server and transferred to the client site.

You can send this ZIP file to the iba support team in order to get help during troubleshooting.

You will also find information about the application itself and its version history here. Clicking “Version history” opens a file that documents new features, improvements, and bug fixes.

4.2 Navigation

You can use the navigation menu to switch between the most important views of *ibaDatCoordinator*. Alternatively, you can select the views from the menu bar.



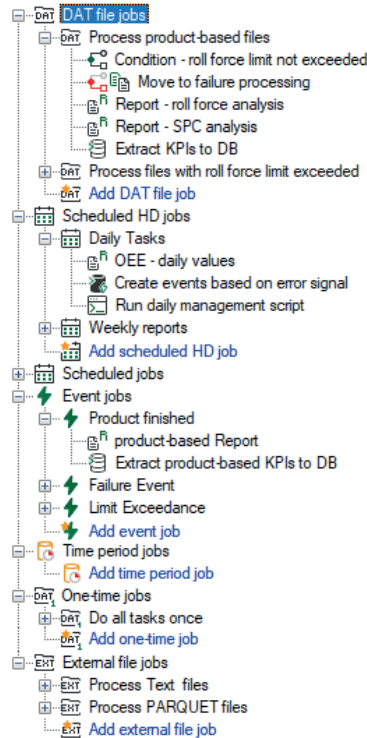
The following views are available:

Jobs	Allows the configuration of jobs and tasks Also see Jobs and tasks, page 25 for the configuration of the individual jobs and tasks.
Status	Displays the license information and the status of the individual jobs in tabular form See Status, page 26 .
Logging	Displays all chronologically sorted log entries See Logging, page 29 .
TCP/IP Watchdog	Configuration of watchdog telegrams for monitoring See TCP/IP Watchdog, page 30 .
SNMP server	Configuration of the integrated SNMP server See SNMP server, page 32 .
OPC UA server	Configuration of the integrated OPC UA server See OPC UA server, page 38 .
Transfer server	Configuration of the transfer server See Transfer server, page 42 .
Settings	General settings Also see chapter Settings, page 44 .

4.2.1 Jobs and tasks

The *Jobs* view in the navigation menu displays a tree structure showing all configured jobs and their tasks. The tree contains individual job nodes and the "Add job" node for creating new jobs.

Click on a job to display and adjust its settings in the right-hand pane.



Each job can have one or more task nodes subordinate to it. These are displayed as soon as you have added tasks in the job configuration. Configuration options are also available for tasks in the right-hand pane.

You will find the following buttons above the navigation menu:

	Saves the configuration and starts all jobs in the tree
	Stop all jobs in the tree
	Applies all configuration changes to the jobs and the task on the server
	Discards all configuration changes to jobs and the task

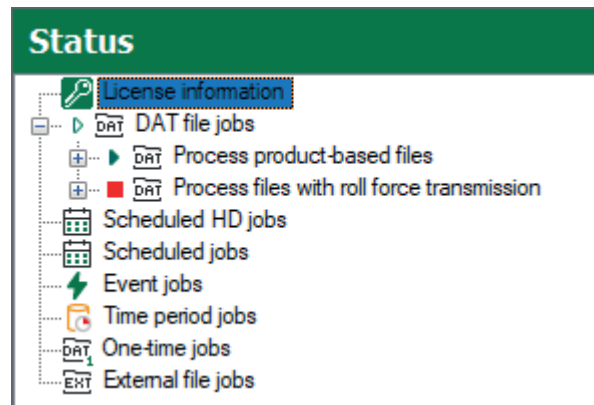
For more information on adding and configuring jobs, see [➤ Configuration of jobs and tasks, page 56](#) and [➤ Configuring a job, page 59](#).

4.2.2 Status

The *Status* view displays license information for *ibaDatCoordinator* as well as the current status of the configured jobs and tasks.

License information

If you select the *License information* node, you will receive details on the licenses used and the available license containers.



The *Used licenses* tab shows all tasks and the licenses required for them. In addition, information about the associated job, license, and license container is displayed for each task.

Tasks that do not require a license can be hidden above the table with the option *Show only tasks that require a license*. You can also group the tasks by job or container, for example. To do this, drag and drop the desired column header into the gray area above the table.

The *License containers* tab shows the license containers available on the server. The ID, the host and a list of the available *ibaDatCoordinator* and *ibaAnalyzer* licenses are displayed for each container.

For further information see ➤ *Licenses*, page 9.

Job status

The icons next to the configured jobs in the tree indicate the status of each running job:

	Job is running/all jobs of the job type are running
	Job is stopped/all jobs of the job type are stopped
	The job type includes running and stopped jobs

If you select a job in the tree, the processed data records, e.g. the tasks per DAT file, are displayed in a table.

The columns for the file name and the number of previous attempts are always visible. For each task configured in the job, an additional column with the respective task name as the title is displayed. You can hide a task column by unchecking the corresponding checkbox in the tree. The <Show all columns> and <Hide all columns> buttons are also available above the table.

The icons in the columns indicate the status of the respective tasks:

Empty	The task has not yet been executed or has not been started due to a defined condition.
Task icon	The task is currently being executed.
Task icon with a blue check-mark	The task was successfully executed or the condition was TRUE.
Task icon with a red cross	The task could not be executed or the condition was FALSE.
Task icon with "No access"	The DAT file currently cannot be accessed.

The screenshot shows the 'Status: process_dat_files' window in the ibaDatCoordinator application. The window title is 'Status: process_dat_files'. Below the title bar, there is a sub-header 'This is the statusdata for' followed by a link 'Process product-based files'. The main area contains a table with the following columns: Processing end time, iba DAT file, Attempts, Publish_KPI, check_plausibility, and move_to_archive. The table lists 16 rows of data, each representing a task execution. The status bar at the bottom indicates 'The job is started' and 'Connected to localhost (4.1.0)'.

Processing end time	iba DAT file	Attempts	Publish_KPI	check_plausibility	move_to_archive
11/28/2023 11:41:48 AM	pda004.dat	1	OPC UA?	✓	📁
11/28/2023 11:41:48 AM	pda005.dat	1	OPC UA?	✓	📁
11/28/2023 11:41:48 AM	pda006.dat	1	OPC UA?	✓	📁
11/28/2023 11:41:49 AM	pda000.dat	1	OPC UA?	✓	📁
11/28/2023 11:41:49 AM	pda001.dat	1	OPC UA?	✓	📁
11/28/2023 11:41:49 AM	energy_test000.dat	1	OPC UA?	✓	📁
11/28/2023 11:41:49 AM	energy_test001.dat	1	OPC UA?	✓	📁
11/28/2023 11:41:49 AM	energy_test002.dat	1	OPC UA?	✓	📁
11/28/2023 11:41:49 AM	energy_test003.dat	1	OPC UA?	✓	📁
11/28/2023 11:41:49 AM	energy_test004.dat	1	OPC UA?	✓	📁
11/28/2023 11:41:50 AM	energy_test005.dat	1	OPC UA?	✓	📁
11/28/2023 11:41:50 AM	energy_test006.dat	1	OPC UA?	✓	📁
11/28/2023 11:41:50 AM	energy_test007.dat	1	OPC UA?	✓	📁
11/28/2023 11:41:50 AM	energy_test008.dat	1	OPC UA?	✓	📁
11/28/2023 11:41:50 AM	energy_test009.dat	1	OPC UA?	✓	📁
11/28/2023 11:41:50 AM	energy_test010.dat	1	OPC UA?	✓	📁
11/28/2023 11:41:50 AM	energy_test011.dat	1	OPC UA?	✓	📁

You can also see the following information on the condition task icon:

	The task was executed successfully, the result is TRUE.
	The task was successfully executed, the result is FALSE.
	The task could not be executed.

Above the table you will find the job name as a link that leads directly to the configuration of the respective job. Below the table, the status of the selected job is indicated.

You can sort and filter the columns in the table as usual. For example, you can filter the task status column to display only files for which the task failed, was processed successfully, caused a timeout, etc.

The context menu of the table contains the following commands:

Open in ibaAnalyzer

Opens the processed file with *ibaAnalyzer* for further analysis.

This command is only available if the file can be found in its original location.

Open in ibaAnalyzer with PDO

Opens the processed file with *ibaAnalyzer*; in addition, the analysis file configured for the task is loaded. The PDO file offered depends on the column you click on.

Show full path

When enabled, the full path is displayed for all files shown in the table. Otherwise, only the file name will be displayed.

Clear filters

Resets active filters in the table.

Records with persistent errors are added to the *Persistently failed DAT files* list and displayed below the job in the tree. This list is only created if the *Retry failed operations maximum* option is enabled in the job configuration window and the set number of repetitions has been exceeded, see also [DAT file job – settings, page 61](#).

In addition to the normal status view, you have the option of selecting one or more files and deleting or reprocessing them.



Deleting files that show a constant failure.



Cleaning up the files that show a constant failure.

Status: Process product-based files - Permanently failed DAT files

The following files have reached the maximum number of job attempts

Show all columns Hide all columns

Process product-based files

☒	Processing end time	iba DAT file	Attempts	Publish KPI	wait_10_sec	check_plausibility
☒	—	C:\dat\ProductGroup1\ProduktNr_000800.dat	10	—	—	—

Connected to localhost (4.1.0)

4.2.3 Logging

The *Logging* view displays warnings, info messages, and error messages that were generated during job execution and written to the log file.

In the logging settings, you can adjust the number of entries displayed and filter the messages as needed.

Use the <Clear all events> button to clear the displayed list. However, the underlying log file remains unchanged.

Time	Job	Task	DAT file	Message
23.06.2026 13:43:51	New job	SNMP publish	C:\DatCoordinator_Tests\triggered\ID_20250217_00055.dat	SNMP-Server ist deaktiviert
23.06.2026 13:43:51	New job	SNMP publish	C:\DatCoordinator_Tests\triggered\ID_20250217_00055.dat	Berechnete Werte aktualisiert
23.06.2026 13:43:51	New job	SNMP publish	C:\DatCoordinator_Tests\triggered\ID_20250217_00055.dat	Start der Auswertung der berechneten Werte
23.06.2026 13:43:51	New job	SNMP publish	C:\DatCoordinator_Tests\triggered\HRM_Produkt_ID_20250217_00040.dat	SNMP-Server ist deaktiviert
23.06.2026 13:43:51	New job	SNMP publish	C:\DatCoordinator_Tests\triggered\HRM_Produkt_ID_20250217_00040.dat	Berechnete Werte aktualisiert
23.06.2026 13:43:51	New job	SNMP publish	C:\DatCoordinator_Tests\triggered\HRM_Produkt_ID_20250217_00040.dat	Start der Auswertung der berechneten Werte
23.06.2026 13:43:51	New job	SNMP publish	C:\DatCoordinator_Tests\triggered\ID_20250217_00040.dat	SNMP-Server ist deaktiviert
23.06.2026 13:43:51	New job	SNMP publish	C:\DatCoordinator_Tests\triggered\ID_20250217_00040.dat	Berechnete Werte aktualisiert
23.06.2026 13:43:51	New job	SNMP publish	C:\DatCoordinator_Tests\triggered\ID_20250217_00040.dat	Start der Auswertung der berechneten Werte
23.06.2026 13:43:51	New job	SNMP publish	C:\DatCoordinator_Tests\triggered\HRM_Produkt_ID_20250217_00031.dat	SNMP-Server ist deaktiviert
23.06.2026 13:43:51	New job	SNMP publish	C:\DatCoordinator_Tests\triggered\HRM_Produkt_ID_20250217_00031.dat	Berechnete Werte aktualisiert
23.06.2026 13:43:51	New job	SNMP publish	C:\DatCoordinator_Tests\triggered\HRM_Produkt_ID_20250217_00031.dat	Start der Auswertung der berechneten Werte
23.06.2026 13:43:51	New job	SNMP publish	C:\DatCoordinator_Tests\triggered\HRM_Produkt_ID_20250217_00023.dat	SNMP-Server ist deaktiviert
23.06.2026 13:43:51	New job	SNMP publish	C:\DatCoordinator_Tests\triggered\HRM_Produkt_ID_20250217_00023.dat	Berechnete Werte aktualisiert
23.06.2026 13:43:50	New job	SNMP publish	C:\DatCoordinator_Tests\triggered\HRM_Produkt_ID_20250217_00023.dat	Start der Auswertung der berechneten Werte
23.06.2026 13:43:50	New job		C:\DatCoordinator_Tests\triggered\ID_20250217_00007.dat	Currently no access to the file because the file is in use by the following processes: ibaAnalyzer (25924)
23.06.2026 13:43:49	New job			Finished scanning directory for unprocessed .dat files
23.06.2026 13:43:49	New job			Start der Verarbeitung der Konfiguration
23.06.2026 13:43:49	New job			Started scanning directory for unprocessed .dat files
23.06.2026 13:43:49	New job	SNMP publish		Lizenz ibaDatCoordinator-Publish (14025) erworben von Container 5HXV8-2FBNN-NBBRX 130-3091155814 (IBA-FUE-NOTE795)

Each event in the list contains the following information:

- Time at which the entry was created in the log file
- Job that *ibaDatCoordinator* has just executed (empty if not applicable)
- Task that the job performed (blank if not applicable)
- Dataset that was loaded by the task (empty, if not applicable)
- Message text in the color of the respective message type:
 - Green: Information
 - Orange: Warning
 - Red: Error

Table options

You can sort and filter all columns in the log table. If a filter is active in the table, an additional line with the filter definition is displayed at the bottom of the table.

Right-click in the table to open a context menu with the following commands:

Open in ibaAnalyzer

Opens the processed file with *ibaAnalyzer* for further analysis.

This command is only available if the file can be found in its original location.

Show full path

When enabled, the full path is displayed for all files shown in the table. Otherwise, only the file name will be displayed.

Clear filters

Resets active filters in the table.

Log files

While executing a task, *ibaDatCoordinator* creates a log of the internal processes and saves it in the form of log files on the hard disk.

Use the <Open log file> button to open the current log file with the standard text editor. The log files can be found in the following directory:

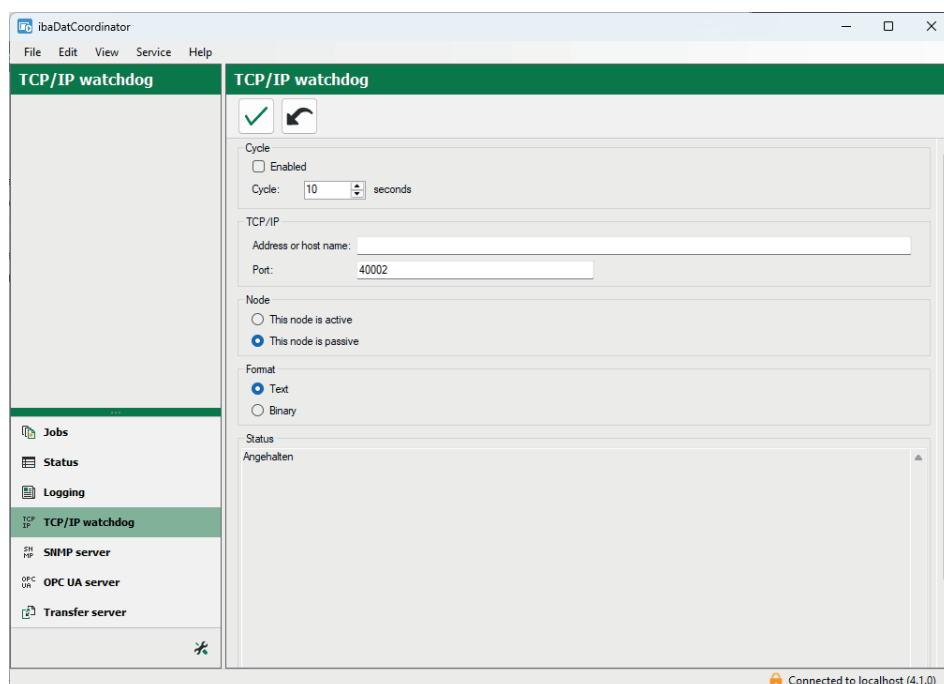
`C:\ProgramData\iba\ibaDatCoordinator\Log`

If you have installed *ibaDatCoordinator* as a standalone executable file, the name of the current log file is `ibaDatCoordinatorLog.txt`. If *ibaDatCoordinator* is installed as a service, you will find two log files `ibaDatCoordinatorLog_client.txt` and `ibaDatCoordinatorLog_server.txt` in the directory.

Previous log files have names with date and time included.

4.2.4 TCP/IP Watchdog

In the *TCP/IP Watchdog* view, you can configure a watchdog that sends a TCP/IP message at regular intervals via a specified port. These messages can be monitored using appropriate applications.



The messages configured here are sent regardless of whether a job is currently running or not. You can establish an active or passive connection to the recipient of the watchdog message. In passive mode, several systems can receive the messages. You can change the transmission cycle in the *cycle* field. The transmission cycle is 10 seconds by default.

Note



The difference between an active and a passive node is that *ibaDatCoordinator* establishes a connection to the recipient if it is configured as active node. If *ibaDatCoordinator* is configured as passive node, the program waits for the recipient to establish a connection.

The watchdog messages can be sent as text or in binary format. In binary format, the message length is limited to 464 bytes, so that up to 16 jobs can be recorded. For further information on the format of the watchdog message, see [Watchdog message, page 31](#).

The *Status* section shows all connections from or to *ibaDatCoordinator*.

4.2.4.1 Watchdog message

Text

The watchdog messages begin with a date in the following format:

"dd/MM/yy HH:mm:ss.fff".

The date and time information is followed immediately by the text ",DatCo".

For each job a string is added with the following format:

"{Job name} : {w} to do, {x} done, {y} erroneous, {z} perm. failed;"

The meaning of the tags used in this format can be found below:

- {Job name} is the name of the job
- {w} is the number of the .dat files in the "to be processed" list of the job.
- {x} is the number of the .dat files in the "processed" list of the job.
- {y} is the number of the .dat files, which could not be executed for one or more tasks.
- {z} is the number of .dat files which have shown a constant failure for one or more tasks.

Example

```
30/04/09 16:12:10.045,DatCo:First Job:300 todo,150 done,0 erroneous,0 perm. failed;Second
Job:200 todo,100 done,1 erroneous,0 perm. failed;
```

Binary

The binary version of the watchdog consists of 4 general integer values and for each job another 7 integer values. Each integer value is 32 bits long (4 byte).

The message length is limited to 464 byte, comprising the information for 16 jobs.

$(4 * 4 \text{ byte} + 16 * 7 * 4 \text{ byte} = 464 \text{ byte})$

Bytes for unused jobs contain zero values.

The table below contains the description for each integer value.

Structure of binary watchdog message:

	Byte	Description
	0	Telegram counter; every time a telegram is sent, this value is incremented by 1.
	4	Current version number; the current version number is set to 1.
	8	Reserved integer, currently with the value 0
	12	Reserved integer, currently with the value 0
Repeated for each job max. 16 jobs	16	Job status; the value is 1 when the job is started and 0 when the job is stopped.
	20	The number of DAT files in the "to be processed" list of the job
	24	The number of DAT files in the "processed" list of the job
	28	The number of DAT files, that could not be executed for one or more tasks
	32	The number of DAT files that show a permanent failure for one or more jobs
	36	Reserved integer, currently with the value 0
	40	Reserved integer, currently with the value 0
	44	Job 2
	72	Job 3
	100	Job 4
	...	...
	436	Job 16

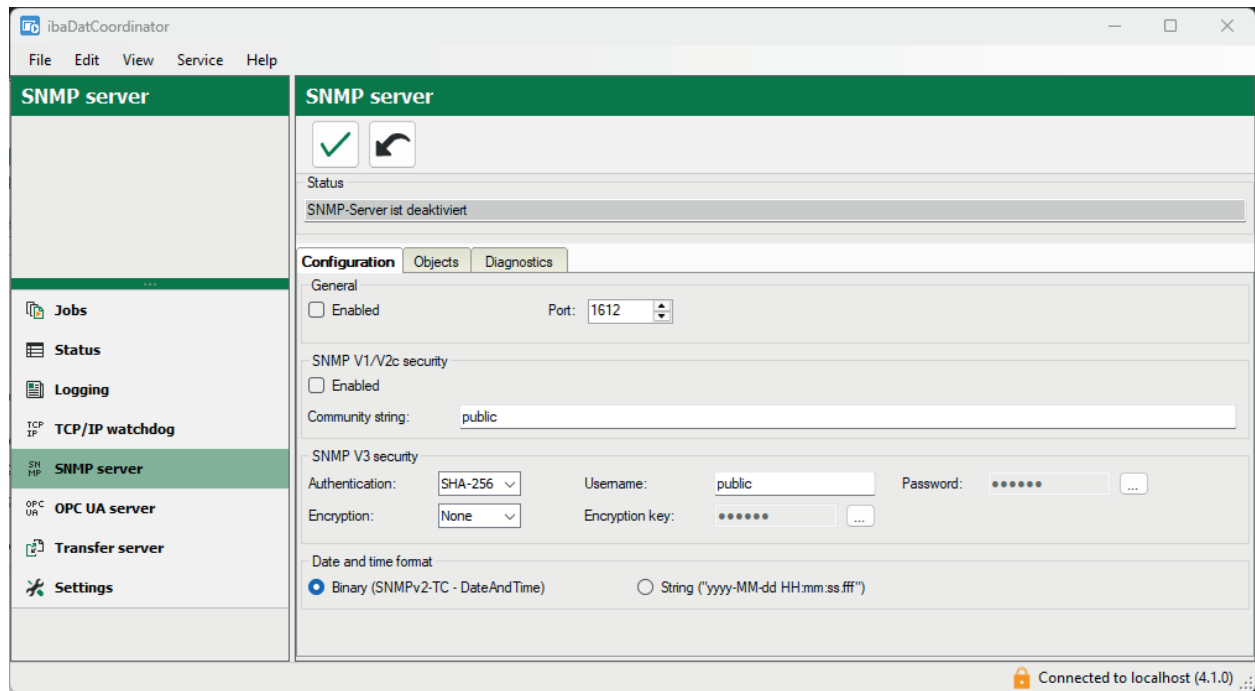
4.2.5 SNMP server

SNMP (Simple Network Management Protocol) was developed to monitor and control network elements (e.g. routers, switches, printers, computers) from a central station. Using the SNMP server, *ibaDatCoordinator* can be monitored by a network monitoring tool such as the Paessler PRTG or Nagios.

SNMP versions V1, V2c and V3 are supported. The SNMP server provides a set of diagnostic signals by default, which provide feedback on the status of the system and the various data recordings.

4.2.5.1 Configuration tab

In the *Configuration* tab you enable the SNMP server and make the basic settings, such as for data security.



Enabled

Here you can activate and deactivate the SNMP server.

Port

The port via which the SNMP server should communicate.

The default value is 1612.

Community string

In the protocol variants V1 and V2c, only a community string can be submitted for authentication. This serves as a password for the data transfer between the SNMP server and SNMP client. The default value for the community string is "public".

Allowed characters are a-z, A-Z, 0-9 and special characters.

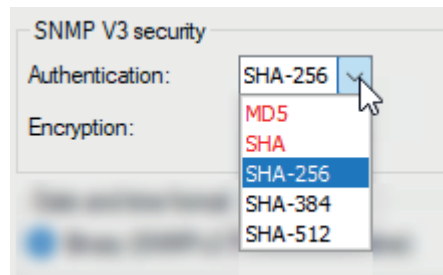
Authentication, user name, password

In protocol version V3, the security standards are more stringent and there are multiple methods available for authentication and encryption. For authentication, the following standards are supported:

- SHA256 (secure hash algorithm, SHA2 class, hash function, 256-bit)
- SHA384 (secure hash algorithm, SHA2 class, hash function, 384-bit)
- SHA512 (secure hash algorithm, SHA2 class, hash function, 512-bit)

In addition to selecting the authentication algorithm, you must also enter a user name and password.

Older standards like MD5 or SHA are not supported anymore. If these standards are still set in older configurations, a warning symbol appears nearby and those standards are highlighted in red in the drop-down list.



Encryption, key

In terms of encryption, you will have the following options:

- No encryption
- AES128 (Advanced Encryption Standard, 128-bit)
- AES192 (Advanced Encryption Standard, 192-bit)
- AES256 (Advanced Encryption Standard, 256-bit)

Older standards like DES are not supported anymore. If these standards are still set in older configurations, a warning symbol appears nearby and those standards are highlighted in red in the drop-down list.

Note



It is best to clarify the settings to be made here with your network administrator. Make sure that the set port is not already being used by any other SNMP servers running on the computer.

The <...> button is used to change the password.

Text encoding

For the correct transmission of ASCII data (text channels), set the appropriate text encoding for the region or language here.

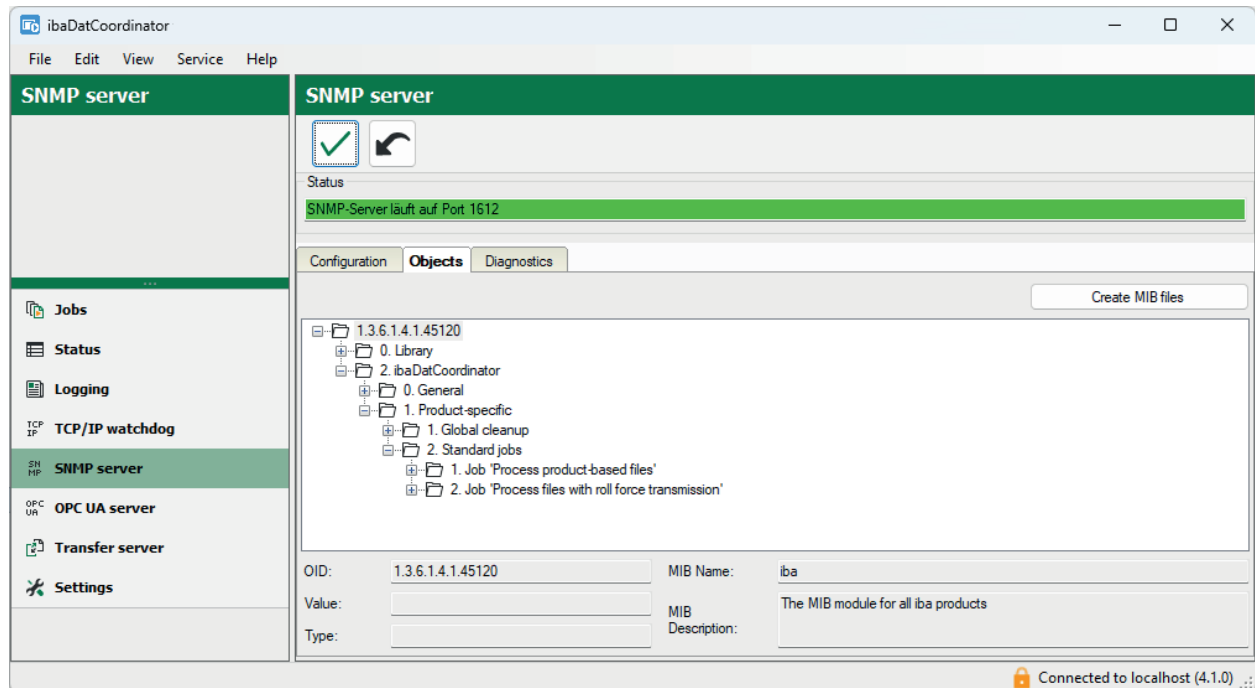
Date and time format

Here you can set the date/time format (if applicable).

- Binary (as per SNMPv2 TC standard encoded , in multiple bytes)
- String (YYYY-MM-dd HH:mm:ss:ffff)

4.2.5.2 Objects tab

Configure the settings for the SNMP server in the *Objects* tab.



Objects

The tree structure displays the objects that are available in the SNMP server. The tree structure shows the OID tree (object identification tree). The objects are divided into a number of groups:

- Objects of the SNMP library
- Objects for the general properties of *ibaDatCoordinator*, such as the version and dongle information
- Objects for global cleanup settings
- Objects for standard tasks
- Objects for scheduled tasks
- Objects for one time tasks

Button <Generate MIB files>

With this button you can start the generation of the MIB (management information base) files.

After clicking the button, select a folder in which the files will be saved. Two MIB files are always created:

- IBA-GENERAL-MIB.txt: This file contains the objects which all iba products have in common.
- IBA-PRODUCT-IBADATCOORDINATOR-MIB.txt: This file contains the objects which are specific to *ibaDatCoordinator*.

The object currently selected is displayed under the OID tree:

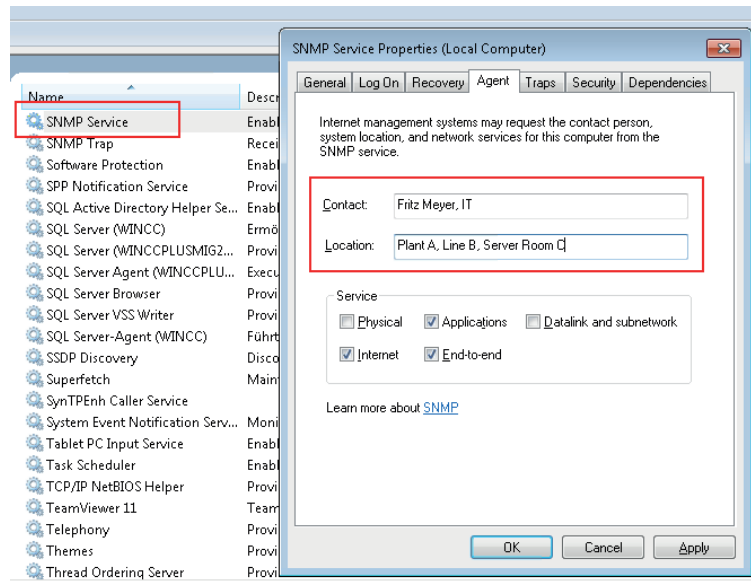
- Full OID
- MIB name
- MIB description
- Current value
- Data type

4.2.5.3 System-OIDs 1.3.6.1.2.1.1.

iba supports the system-OIDs 1.3.6.1.2.1.1.x, which are used by SNMP tools for automatic server detection.

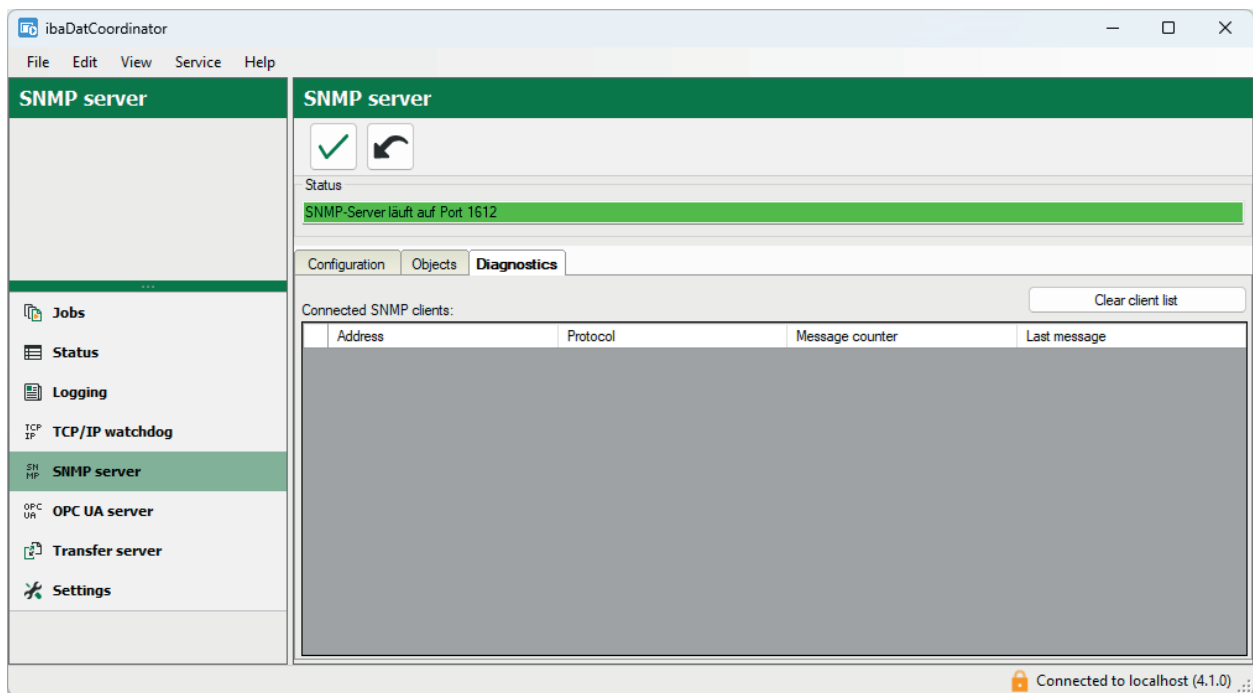
OID	Description
1.3.6.1.2.1.1.1	sysDescr: sysDescr: fix value "ibaSNMP agent for ibaPDA"
1.3.6.1.2.1.1.2	sysObjectID: fix value "1.3.6.1.4.1.45120.1". This is iba AG OID +"1" for the product ibaPDA fix value "1.3.6.1.4.1.45120.2". This is iba AG OID +"2" for the product ibaDatCoordinator fix value "1.3.6.1.4.1.45120.4". This is iba AG OID +"4" for the product ibaHD-Server fix value "1.3.6.1.4.1.45120.6". This is iba AG OID +"6" for the product ibaCapture
1.3.6.1.2.1.1.3	sysUpTime: Dynamic: Time since starting ibaPDA service (in 1/100 seconds)
1.3.6.1.2.1.1.4	sysContact: Dynamic: Must be manually configured in the SNMP-service settings.
1.3.6.1.2.1.1.5	sysName: automatically generated: computer name plus domain suffix, e.g. iba-fue-wks123.iba-ag.local
1.3.6.1.2.1.1.6	sysLocation: Dynamic: Must be manually configured in the properties of the Windows SNMP-service
1.3.6.1.2.1.1.7	not supported

Contents of the system-OIDs [sysContact](#) and [sysLocation](#) are set by default in the *Agent* tab in the Windows service "SNMP-service".



4.2.5.4 Diagnostics tab

The *Diagnostics* tab displays information about the status of the SNMP server.



Connected SNMP clients

List of all clients that have accessed the SNMP server within the past hour.

Clients that have accessed the SNMP server in the last 10 minutes are marked with a green background, all others with a gray background.

Using the <Clear client list> option you can remove all entries from the list.

4.2.6 OPC UA server

In the *OPC UA server* view, you can configure the OPC UA server integrated in *ibaDatCoordinator*. This provides status information from *ibaDatCoordinator* for other systems. If an OPC UA publish task (subject to license) is also configured, calculated values are also published in this way.

Note



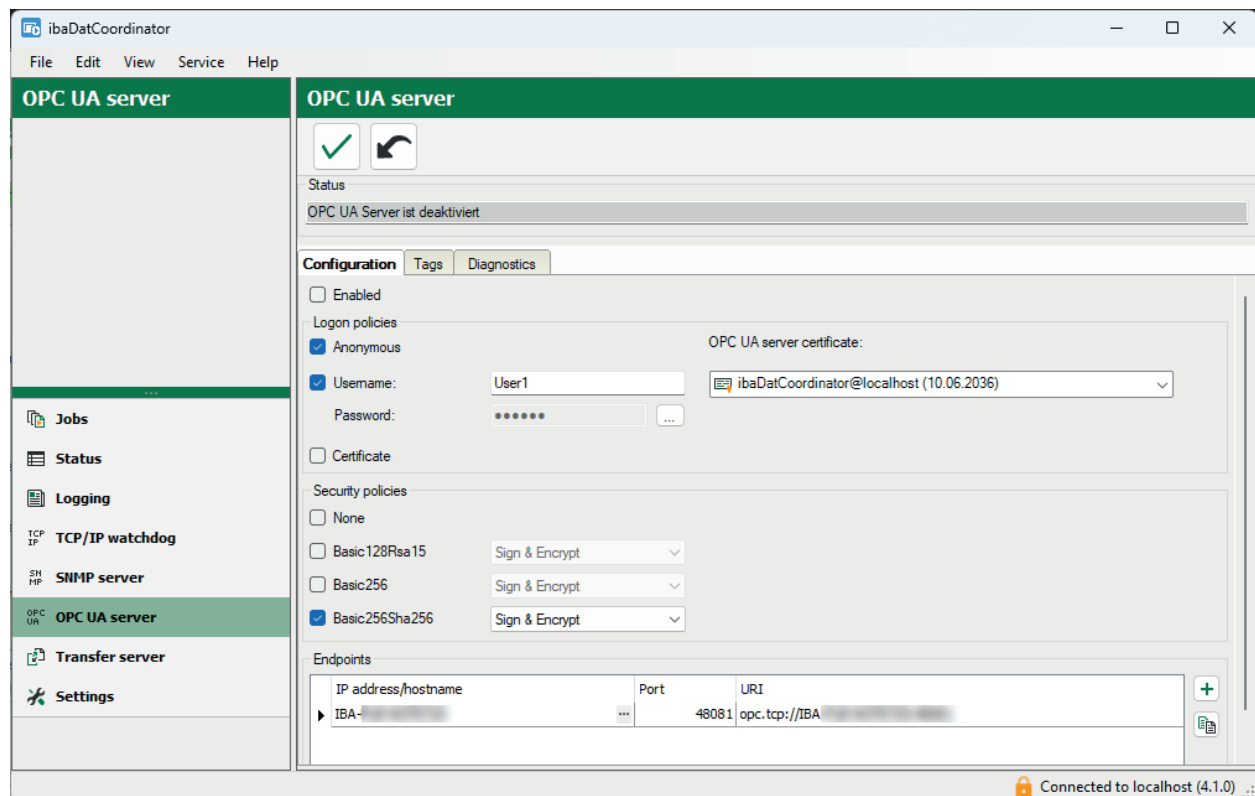
The basic function for status monitoring of *ibaDatCoordinator* via OPC UA server can be used without an additional license. A license is required to publish values calculated with *ibaAnalyzer*.

Before an OPC UA client can connect to *ibaDatCoordinator* (OPC UA server), an application certificate must be configured first. Certificates can be provided from both the server and client side. OPC UA communication can only take place if both partners trust the certificate.

Certificates can be exchanged spontaneously during connection setup, or you can register the certificates in advance as trusted certificates. If a previously unknown certificate is offered during connection setup, this certificate must be manually accepted or rejected by the user. Accepted certificates are automatically entered into the *Certificates* table and marked as trusted, see [➤ Central certificate store, page 47](#). If the certificate is rejected, no communication will be established.

4.2.6.1 OPC UA server - Configuration

In the *Configuration* tab, activate the OPC UA function and define login rules, security rules, and endpoints for communication.



Enabled

Check this box to enable the OPC UA server function.

OPC UA server certificates

Select the certificate that the OPC UA server should use.

If you have not yet generated or imported a certificate, you can do so by selecting *Create new certificate* or *Manage certificates* from the list. You will then be routed to the central certificate store.

The handling of certificates is described in chapter ↗ *Settings – Certificates, page 47*

Login policies

At least one of the following login rules should be applied:

Anonymous

If this option is enabled, clients can log on to the OPC UA server without login credentials (user/password).

User name/password

If this option is enabled, clients can only log in if they can authenticate themselves by entering their user name and password in the dialog.

The <...> button is used to change the password.

Certificate

If this option is enabled, clients can log in if they can authenticate themselves by using a trusted certificate.

Security policies

At least one of the options must be enabled.

If you enable the option *None*, then connections without encryption are also supported.

For each of the other options or encryption types, you can select a security rule with signature and/or encryption:

- Sign
- Sign & Encrypt
- Sign + Sign & Encrypt

Note

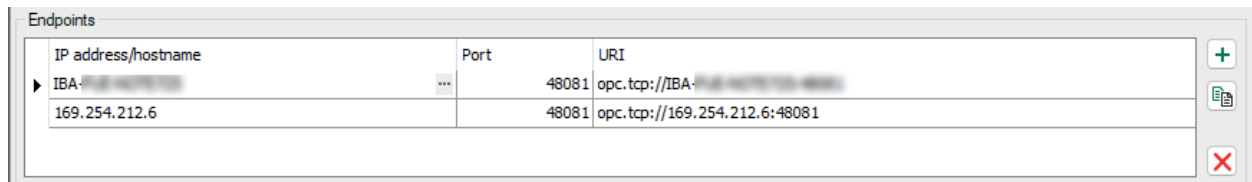
Basic128Rsa15 and Basic256 encryption are now obsolete.

For security reasons, the use of Basic256Sha256, Aes128-Sha256-RsaOaep or Aes256-Sha256-RsaPss encryption is preferred.

Endpoints

This area of the dialog lets you configure the local endpoints that the server will provide.

An endpoint is a combination of an IP address and a port number. Instead of entering a specific IP address, it is also possible to enter the computer name of the OPC UA server. This applies to all IP addresses of all network interfaces in the system. The URI (Uniform Resource Identifier) is formed from the IP address or computer name and the port number.



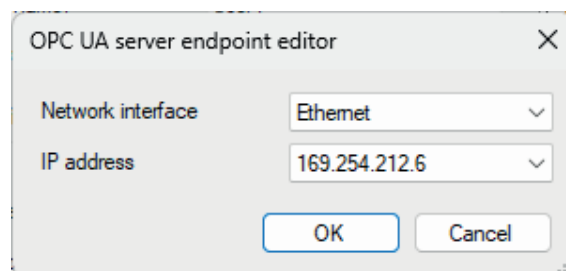
In the example above, OPC UA clients can connect to the OPC UA server from any network using port 48081 or 48082. In addition, clients can also connect to IP address 172.29.0.100 via port 49000.

The list of endpoints includes some controls:

Button	Function
	Use this button to add a new endpoint. Initially, the new endpoint always has the same data as the local computer, which then needs to be edited.
	This button lets you duplicate the selected endpoint, which you can then edit.
	Use this button to delete the selected endpoint from the list.

By default, the name of the local computer is already entered in the list.

If you click on an endpoint in the list, the button will appear. . Click on the button to edit the endpoint.



Select the desired network interface via which the OPC UA clients should communicate. The dropdown list shows all of the computer's available network interfaces with all IP addresses.

Click <OK> and the selected IP address will be displayed in the list.

Note



Currently only IPv4 addresses are supported.

4.2.6.2 OPC UA server - Tags

This tab displays the tags that can be published by the OPC UA server.

- Dongle-specific information in the *Licensing* folder
- Information on global cleanup for each drive of the computer
- Status information on all configured jobs and the task, divided by job types

The day currently selected is displayed under day tree:

- Node ID: corresponds to a path specification of the tag
- Value: Numerical or boolean value of the tag
- Type: Data type of the tag (Boolean, different numerical data types, Enum, string, Date Time)
- Description: Short description of the meaning of the tag

4.2.6.3 Diagnostics tab

The current status of the OPC UA server is displayed in the *Diagnostics* tab. You will also see a list of connected OPC UA clients and subscriptions there.

Name	ID	Last message time
ibaPDA [OPC UA Client]	ns=4;i=1966665814	11:08:36
urn:DEVPC-HARRISON:UnifiedAutomation:U...	ns=4;i=1966665902	11:08:35

ID	Monitored items count	Publishing interval	Next sequence number
3	2	50 ms	949
4	1	100 ms	1

For each connection, the name and ID of the session are displayed along with the time stamp of the most recent communication.

For each subscription, the subscription ID, the number of monitored tags, the publication interval and the next sequence number are displayed. The latter value is increased each time new data is sent for a particular subscription.

If you select an OPC UA client in the upper list, the subscriptions are filtered so that only subscriptions that are relevant for this client are displayed in the lower list.

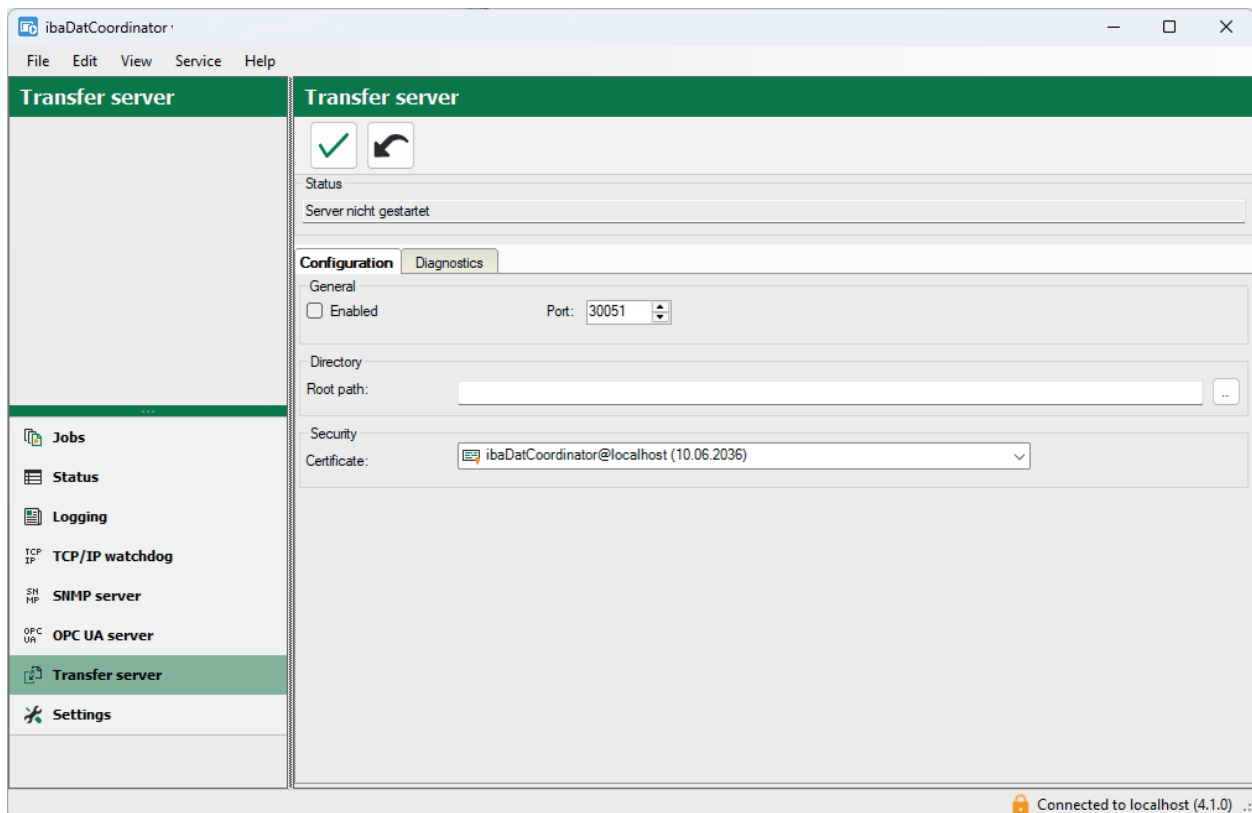
4.2.7 Transfer server

In the *Transfer server* view, you can configure *ibaDatCoordinator* to receive files from another *ibaDatCoordinator* instance.

Note



In this menu, you can only configure the receiving of files. To send files to another *ibaDatCoordinator* instance, see chapter [Transfer task, page 91](#).



The *Status* area shows whether the server has already been started or stopped.

4.2.7.1 Transfer Server – Configuration

On the Configuration tab, configure the following settings for the transfer server. .

Enabled

Enable or disable transfer server

Port

Selection of any TCP port for the communication to this *ibaDatCoordinator* instance

The port 30051 is preset.

Root path

Path where the sent files are to be saved.

The path setting of the respective sending transfer task is relative to the path specified here, see [Transfer task, page 91](#). You can navigate to the desired directory using the <...> button.

Certificate

Here you can select a certificate for the encryption of the connection. If no certificate is selected, the connection is established unencrypted. You can also create a new certificate.

Note

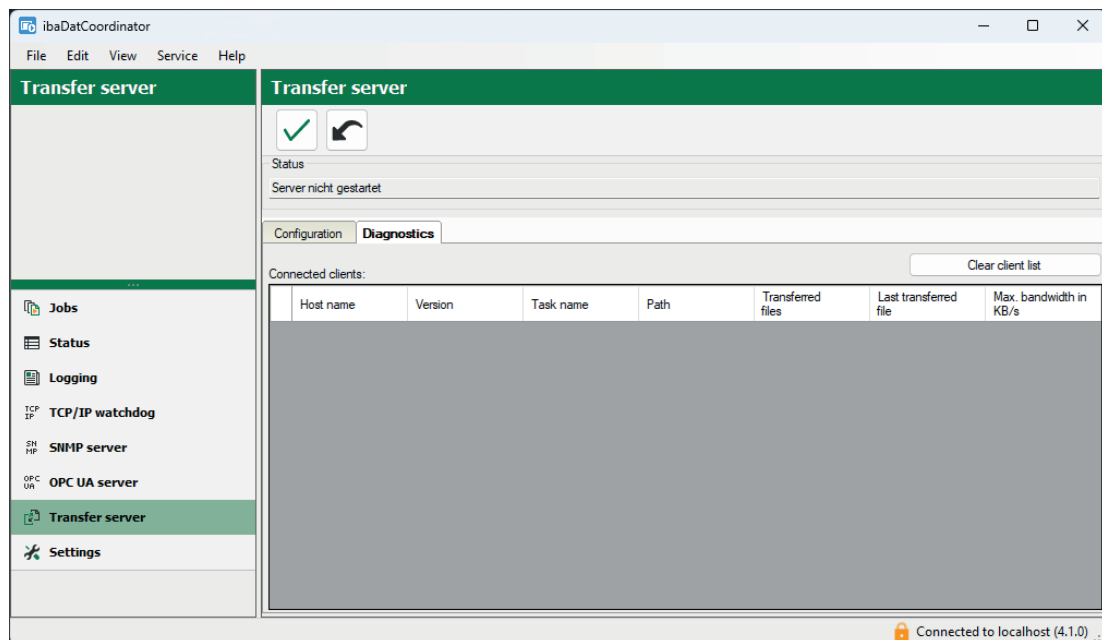


The same certificate must also be used in all sending transfer tasks, see [↗ Transfer task, page 91](#).

If you select the “Manage certificates” option, the *Settings* view opens. You can also import and export existing certificates in the certificate management. When exporting the certificate, the private key does not have to be exported. See chapter [↗ Settings – Certificates, page 47](#).

4.2.7.2 Transfer Server – Diagnostics

The *Diagnostics* tab displays all incoming connections to the transfer server and related information for connections that have transmitted data to the server since the server was last started.



In addition to the host name and the *ibaDatCoordinator* version, the name of the task, the absolute target path, the number of files transferred, the last file transferred and the configured maximum bandwidth are displayed.

Note

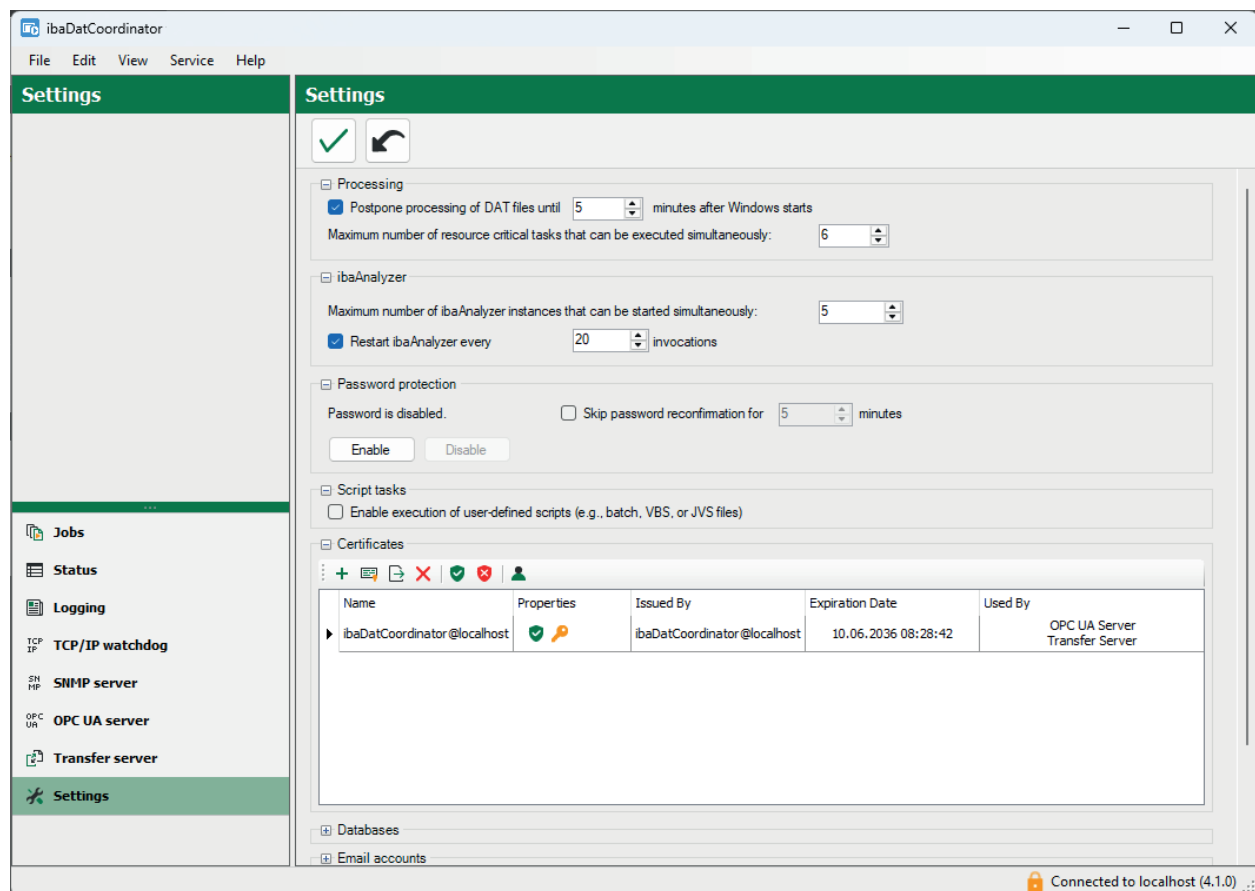


A value of "0" for the maximum bandwidth means "no restriction".

You can reset the display with the <Clean client list> button.

4.2.8 Settings

In the *Settings* view, you can adjust a range of basic settings for the *ibaDatCoordinator* service and other functions.



4.2.8.1 Settings – Processing

In the *Processing* section, you can set how long *ibaDatCoordinator* waits after Windows starts before processing DAT files and how many resource-critical tasks are executed simultaneously.

Postpone processing of DAT files until... minutes after Windows starts

This option ensures that Windows and all important applications and services are started before *ibaDatCoordinator* scans the DAT file directory for unprocessed DAT files.

When Windows starts, all services are loaded first. When the *ibaDatCoordinator* service starts, *ibaDatCoordinator* waits for the defined time before the application starts searching for unprocessed DAT files.

This prevents connection problems with services that have not yet been started, such as database and network connections or printer drivers.

Maximum number of resource critical tasks that can be executed simultaneously

With this setting, you can limit the number of the tasks to be processed in parallel that are resource critical. There is a high demand on resources if you process complex analyses with high volumes of data in the tasks, e.g. comprehensive reports with FFT analyses.

Because multiple instances of *ibaAnalyzer* can be started when processing tasks in parallel, the number should be limited so that the computer does not reach its performance limits. The default setting is 6.

If you want to process more tasks in parallel, you must activate the <Optimize system resources> option in the *ibaDatCoordinator Status* app, see [🔗 ibaDatCoordinator Status app](#), page 16.

4.2.8.2 Settings – ibaAnalyzer

ibaAnalyzer path

Path to the [ibaAnalyzer.exe](#) application (read-only)

To test the path and open *ibaAnalyzer*, click on the button .

Note



This option is only available if *ibaDatCoordinator* has been installed as a stand-alone executable file. In the case of a client-server installation, this option is located in the *ibaDatCoordinator status* app, see [🔗 ibaDatCoordinator Status app](#), page 16.

Maximum number of ibaAnalyzer instances...

With this setting, you can limit the maximum permissible number of *ibaAnalyzer*-instances running in parallel. This setting is also used to adapt the system resources in order to prevent an overload of the computer. Depending on the performance of the computer (processor) you can run more or fewer instances in parallel. The default setting is 5.

When the permissible number of instances in operation is reached, additional tasks that require *ibaAnalyzer* for processing must wait.

Restart ibaAnalyzer every... invocations

When enabled, *ibaAnalyzer* will restart after processing a fixed number of data blocks.

4.2.8.3 Settings – Password protection

With password protection for *ibaDatCoordinator*, you can prevent unauthorized persons from changing settings.

After the first successful authorization, you can activate the option *Skip password reconfirmation for ... minutes*. This allows you to specify how long protected functions can be executed without requiring the password to be re-entered.

If the password protection is enabled, a password will be needed for the following actions:

- Starting/stopping the *ibaDatCoordinator* service
- Starting/stopping jobs (<STOP> and <GO> buttons)
- Dragging jobs or tasks within the tree
- Accessing the *ibaDatCoordinator* settings

Enabling password protection (for the first time)

1. Click on <Enable>.
2. Enter the password in the *Password* field.
3. Enter the same password once again in the *Confirm password* field.
4. Click <OK>.

Changing the password

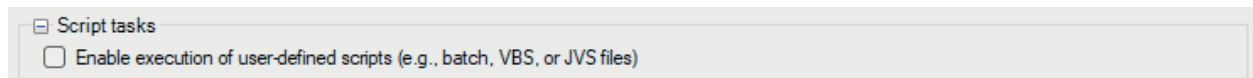
1. Open the *Settings* section in the navigation bar.
2. Click on <Modify> in the *Password protection* area.
3. Enter the current password.
4. Enter the new password in both fields.
5. Click <OK>.

Disabling password protection

1. Open the *Settings* section in the navigation bar.
2. Click on <Disable> in the *Password protection* area.
3. Enter the current password.
4. Click <OK>.

4.2.8.4 Settings – Script task

To minimize security risks, the use of script tasks is disabled by default. You can enable them in the settings.

**Enable execution of user-defined scripts (e.g., batch, VBS, or JVS files)**

If you enable this option, a message will appear that you must confirm. After that, you can use the task type in the jobs. For information on configuring script tasks, see [➤ Script task, page 108](#).

4.2.8.5 Settings – Certificates

For secure and encrypted TLS/SSL communication between a client and a server, certificates are used because they enable secure authentication. Certificates used by iba applications 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 be exchanged spontaneously during connection setup, or you can register the certificates in advance as trusted certificates. If a previously unknown certificate is offered when establishing a connection, the user must interactively 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, no communication will be established.

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. have been 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".

4.2.8.5.1 Central certificate store

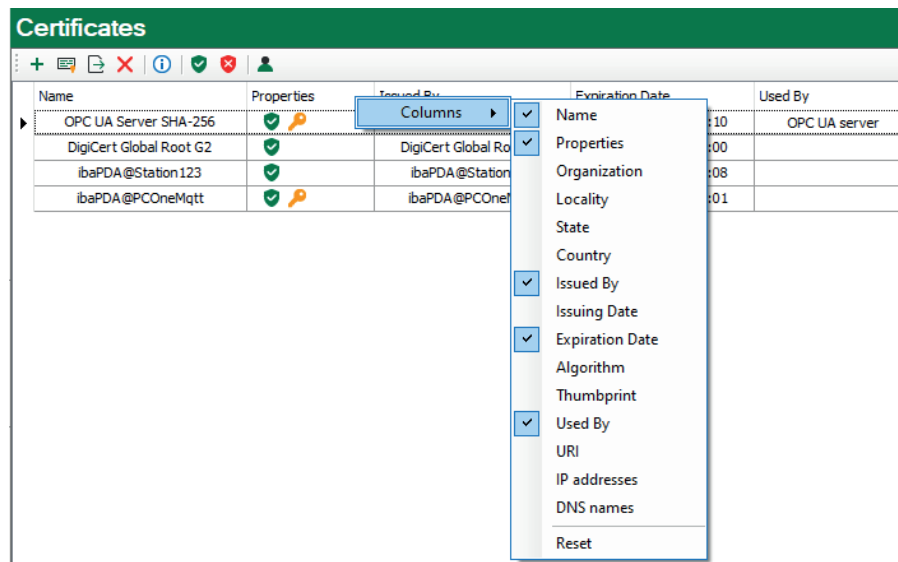
All registered certificates are listed in a table. The following figure shows the certificate store in the I/O Manager of *ibaPDA* as an example.

Certificates				
				
Name	Properties	Issued By	Expiration Date	Used By
OPC UA Server SHA-256	 	OPC UA Server SHA-256	20.12.2031 12:29:10	OPC UA server
DigiCert Global Root G2		DigiCert Global Root G2	15.01.2038 13:00:00	
ibaPDA@Station123		ibaPDA@Station123	07.02.2032 10:26:08	
ibaPDA@PCOneMqtt	 	ibaPDA@PCOneMqtt	11.02.2032 10:16:01	

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 certificates 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 1: 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 *OPC UA Server SHA-256* is used by the OPC UA server in *ibaPDA*. This means that this certificate was selected during configuration of the OPC UA server.

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.

4.2.8.5.2 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 open a window with all detailed information about the selected certificate.
	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.

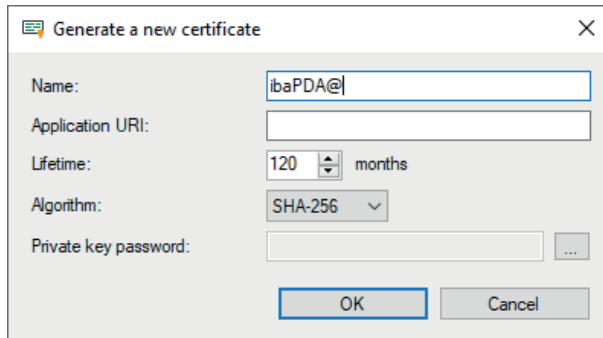
Table 2: Buttons in the toolbar for certificate management

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.

4.2.8.5.2.1 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).

4.2.8.5.2.2 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.

4.2.8.5.2.3 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.

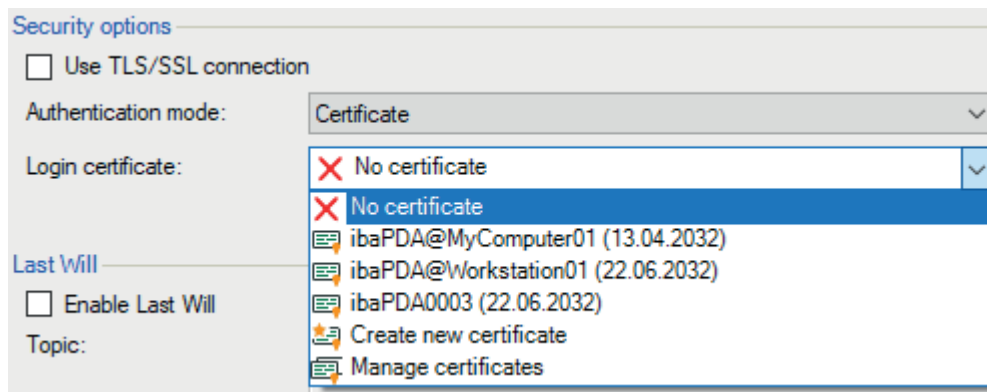
If the password is incorrect, the certificate will not be exported.

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.

4.2.8.5.3 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*, *ibaDatCoordinator* etc.

4.2.8.5.4 Save and protect certificates

The certificates are stored in the `settings.xml` file, which is located in the folder `c:\ProgramData\iba\Name of application\Certificates`. This file is automatically encrypted.

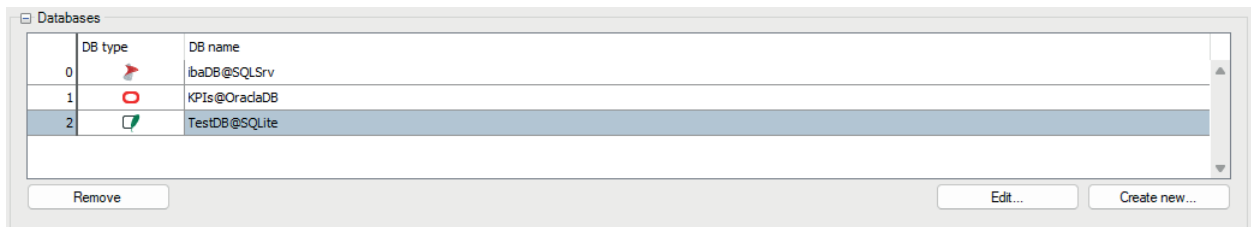
There are a number of measures whereby certificates with private keys can be used to protect your identity or that of your organization. Specifically, these are measures that make their simple export and reuse in Windows or other applications more difficult.

- 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.

4.2.8.6 Settings – Databases

You can configure and manage connections to databases in the *Databases* section.



Use the <Create new ...> button to open the dialog to create a database connection.

To edit an existing connection, select the connection from the table and click on the <Edit ...> button.

Settings in the database connection dialog

DB type

Select the database type used.

Connection name

This line contains the name of the database connection. The name is automatically formed and entered according to the scheme `username@database` as soon as you have filled in the fields below. You can also overwrite the connection name.

Other documentation



A detailed description of the other connection-specific settings can be found in the *ibaAnalyzer* documentation, part 4 "Database Interface".

You can use the <Test connection> button to check access to the selected database on the set DB server.

You can remove selected connections in the table with <Remove>.

4.2.8.7 Settings – Email accounts

In the email accounts section, you can configure the accounts that will be used to send notifications. You can then select the email accounts you create here in the job configurations for sending notifications.

Click the button to add an account and enter the required information. You can duplicate an account in the list by clicking the button. This is helpful if, for example, you want to set up multiple senders.

To use a standard SMTP server, select "Other" in the *Provider* field. Alternatively, you can select other SMTP server types in this field, such as "Microsoft Exchange Online". The input form

changes accordingly so that you can configure the settings for the selected provider. In this context, *ibaDatCoordinator* supports the OAuth 2.0 authentication protocol.

4.2.8.8 Settings – Global cleanup

The *Global cleanup* feature ensures that a defined minimum amount of storage space is always available on a computer's local drives.

Drive - [Volume Name]	Size	Curr. free space	Min. free space	Rescan time	Active
C:\ - [OS]	473.96 GB	143.87 GB	10%	47.4 GB	1 min ☐

Please observe the following rules when using this function:

- The cleanup function only processes data files (DAT files).
- Only local drives are considered, including external drives and USB media. Network drives are not included.
- The cleanup function must be activated separately for each drive. To do this, select the *Active* option for the desired drive.
- For use on the system drive (C:\), a subdirectory must be specified that is to be monitored by the cleanup function.
- The minimum free storage space is specified as a percentage per drive. Consider the "growth rate" of the storage space for data files and the setting of the rescan duration in order to choose the correct value for the free storage space.
- When the defined minimum is reached, the function deletes the oldest data files on the respective drive, regardless of the directory in which they are located.
- It is checked periodically in configurable time intervals (rescan duration) if the minimum has been reached. The minimum rescan time is 1 min.

Here, too, take into account the "growth rate" of storage space for data files and the setting for minimum free storage space.

- To clean up other file types, you can create a cleanup task and define the corresponding file types there, see [🔗 Cleanup task, page 102](#).

Note



To avoid overlaps in cleanup operations, disable task-specific cleanup strategies if you want to use global cleanup.

5 Configuration of jobs and tasks

With the job type you specify which data source is to be processed. Within the job, you then configure tasks and determine how the data is processed.

All jobs that you configure in a project run in parallel and can contain multiple tasks. The tasks within a job run sequentially.

You can adjust the order of the job nodes in the job tree using drag & drop. Thus, the sequence changes in which the nodes are displayed in the job tree and the status tree.

You can delete, cut, copy, or paste a job using the context menu in the job tree. Some of the commands are only available if the job is not currently running. You can also move the task nodes using drag & drop to change the order in which they are processed.

Note



The order of the tasks in the tree is important because they are processed from top to bottom. Therefore, pay particular attention to correct sequencing when a task depends on the result of a previous one.

Tip



If you perform drag and drop while holding down the <Ctrl> key, a duplicate of the moved job or task node is created.

For an overview of the available job types, see [➤ Overview of job types, page 56](#).

A detailed description of how to configure jobs can be found in [➤ Configuring a job, page 59](#).

5.1 Overview of job types

Below you will find an overview of the available job types in *ibaDatCoordinator*.

Job	Description
DAT file job	Automatic processing of DAT files in one or more directories See ➤ DAT file job, page 57 .
Scheduled HD job	Regular processing of measurement data in a configurable cycle See ➤ Scheduled HD job and scheduled job, page 58 .
Scheduled job	Regular processing of tasks that do not require measurement data. See ➤ Scheduled HD job and scheduled job, page 58 .
Event job	Processing of data when a defined event occurs (HD event store) See ➤ Event job, page 58 .
Time period job	Processing of time periods from a time-based HD store See ➤ Time period job, page 58 .

Job	Description
One-time job	One-time processing of a data file See ➤ One-time job, page 59 .
External file job	Conversion of external file formats (CSV, DAS, COMTRADE, PARQUET) to DAT format (additional license required) See ➤ External file job, page 59 .

5.1.1 DAT file job

With this job type, *ibaDatCoordinator* monitors one or more directories in which DAT files are stored. Monitoring is performed either via file system monitoring or by regularly scanning the directories. As soon as *ibaDatCoordinator* detects a new, unprocessed DAT file, the job starts processing the defined tasks.

During execution, *ibaDatCoordinator* monitors the status of tasks within a job and marks DAT files that have already been processed. For this purpose, info fields are added to each DAT file, which show whether a job has been fully processed or has failed, and how many attempts at processing have already been made. When you open a processed DAT file in *ibaAnalyzer*, you will find the info fields \$DATCOOR_status and \$DATCOOR_times_tried in the info section of the data file.

A job is considered failed if at least one of its tasks could not be executed. As long as a job is not fully completed, *ibaDatCoordinator* will try to repeat the job.

Note



When a job is repeated, all of its tasks are executed again, including those that have already been successfully completed.

In the job configuration, you can specify how long and how often DAT files whose processing has failed are reprocessed. If the maximum number is exceeded, *ibaDatCoordinator* permanently marks the files as failed. You can then decide whether to delete these files or reset the status information to enable reprocessing.

You can find out how to configure jobs in *ibaDatCoordinator* in [➤ Configuring a job, page 59](#). A description of the specific configuration options for the DAT file job can be found in [➤ DAT file job – settings, page 61](#).

5.1.2 Scheduled HD job and scheduled job

The job type **scheduled HD job** type was specifically designed to access data records generated by the *ibaHD-Server*. *ibaDatCoordinator* executes the defined tasks in a configurable cycle. The measurement data for the job is not provided as a classic data file, but is retrieved cyclically from the HD store for an adjustable time interval. Based on the configured HD store and the time period, *ibaDatCoordinator* generates an HD query (HDQ file), which is transferred to *ibaAnalyzer* for processing the tasks. Then the actual data query is performed. The HD query can be processed in *ibaAnalyzer* like a normal data file. The period queried is relative and always refers to the time when the task was executed (trigger).

With the **scheduled job**, on the other hand, you can run tasks cyclically that do not require measurement data, such as the cleanup or script task. In this case, no HD query is required and *ibaDatCoordinator* behaves like a simple scheduler.

For scheduled HD jobs and scheduled jobs as well, *ibaDatCoordinator* monitors the processing status and retries failed tasks at configurable intervals or up to a defined maximum number of retries.

You can find out how to configure jobs in *ibaDatCoordinator* in [➤ Configuring a job, page 59](#). A description of the specific configuration options for the scheduled HD job and the scheduled job can be found in [➤ Scheduled HD job and scheduled job – settings, page 65](#).

5.1.3 Event job

This job type is executed when a defined event occurs that has been configured in *ibaHD-Server* for event-based recording. To configure and execute the included tasks, *ibaDatCoordinator* therefore requires access to *ibaHD-Server*. To detect the occurrence of events, *ibaDatCoordinator* periodically queries the HD server.

You can find out how to configure jobs in *ibaDatCoordinator* in [➤ Configuring a job, page 59](#). A description of the specific configuration options for the event job can be found in [➤ Event job – settings, page 70](#).

5.1.4 Time period job

Time periods mark a time range within a time-based HD store and are grouped in time period store. A time-based HD store can contain multiple time period stores, each of which contains multiple time periods.

Similar to DAT files, time periods can be processed by *ibaDatCoordinator* when finished.

You can find out how to configure jobs in *ibaDatCoordinator* in [➤ Configuring a job, page 59](#). A description of the specific configuration options for the time period job can be found in [➤ Time period job – Settings, page 73](#).

5.1.5 One-time job

With this job type, you can execute defined tasks exactly once, i.e., based on a data file. Every reprocessing has to be started manually. This is especially helpful for test purposes, commissioning or in rare individual cases.

You can find out how to configure jobs in *ibaDatCoordinator* in [↗ Configuring a job, page 59](#). A description of the specific configuration options for the time period job can be found in [↗ One-time job – settings, page 77](#).

5.1.6 External file job

The external file job offers the option of automatically converting external files in supported formats into DAT files. You can process the resulting DAT files as usual with a DAT file job. This allows such files to be integrated directly into the iba workflow.

To use the external file job, you need an additional license depending on the file format to be converted. If you want to convert different file formats, you need one license per format.

Note



The *ibaAnalyzer-E-Dat* license is normally required to import external data formats into *ibaAnalyzer*. Since the functionality is already included in the corresponding licenses for using the external file job, there is no need to purchase a separate *ibaAnalyzer-E-Dat* license.

Detailed information about the licenses can be found in [↗ Licenses, page 9](#).

You can find out how to configure jobs in *ibaDatCoordinator* in [↗ Configuring a job, page 59](#). A description of the specific configuration options for the external file job can be found in [↗ External file job – settings, page 77](#).

5.2 Configuring a job

To create and configure a job in *ibaDatCoordinator*, proceed as follows.

1. First, decide what type of job you need, see [↗ Overview of job types, page 56](#).
2. Click on the corresponding link in the tree structure:
 - Add DAT file job
 - Add scheduled HD job
 - Add scheduled job
 - Add event job
 - Add time period job
 - Add one-time job
 - Add external file job
3. Click on the newly added job node.
→ In the display and configuration area, the basic settings for this job are shown.

4. Set a name for the job.

The default name is "New job". For better comprehensibility, iba recommends choosing an application-specific name.

5. Add one or more tasks to the job, see [🔗 Configuring a task, page 82](#).

Please note that the tasks will later be processed in the order in which they are listed in the job tree.

6. In the *Notifications* section, specify whether email notifications should be sent whenever the job runs or at specific intervals. You can further specify the settings later for each subordinate task.

Select the account for the sender and enter the recipient's email address.

For more information on configuring email accounts, see [🔗 Settings – Email accounts, page 54](#).

7. Configure the job-specific settings, see

- [🔗 DAT file job – settings, page 61](#)
- [🔗 Scheduled HD job and scheduled job – settings, page 65](#)
- [🔗 Event job – settings, page 70](#)
- [🔗 Time period job – Settings, page 73](#)
- [🔗 One-time job – settings, page 77](#)
- [🔗 External file job – settings, page 77](#)

8. Add one or more tasks to the job, see [🔗 Configuring a task, page 82](#).

Please note that the tasks will later be processed in the order in which they are listed in the job tree.

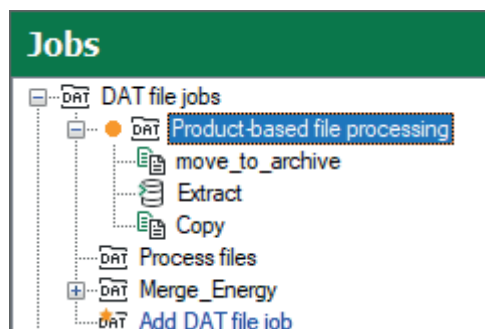
9. If you want to undo the configuration changes, click .

→ The most recently saved and currently active configuration will be downloaded and applied.

10. To apply changes to the job configuration, click .

Please note that the button for individual jobs is only active while the job is running. Alternatively, apply the configuration changes to all jobs by clicking the same button above the navigation menu.

Jobs with unsaved configuration changes are marked with an orange dot in the job tree:

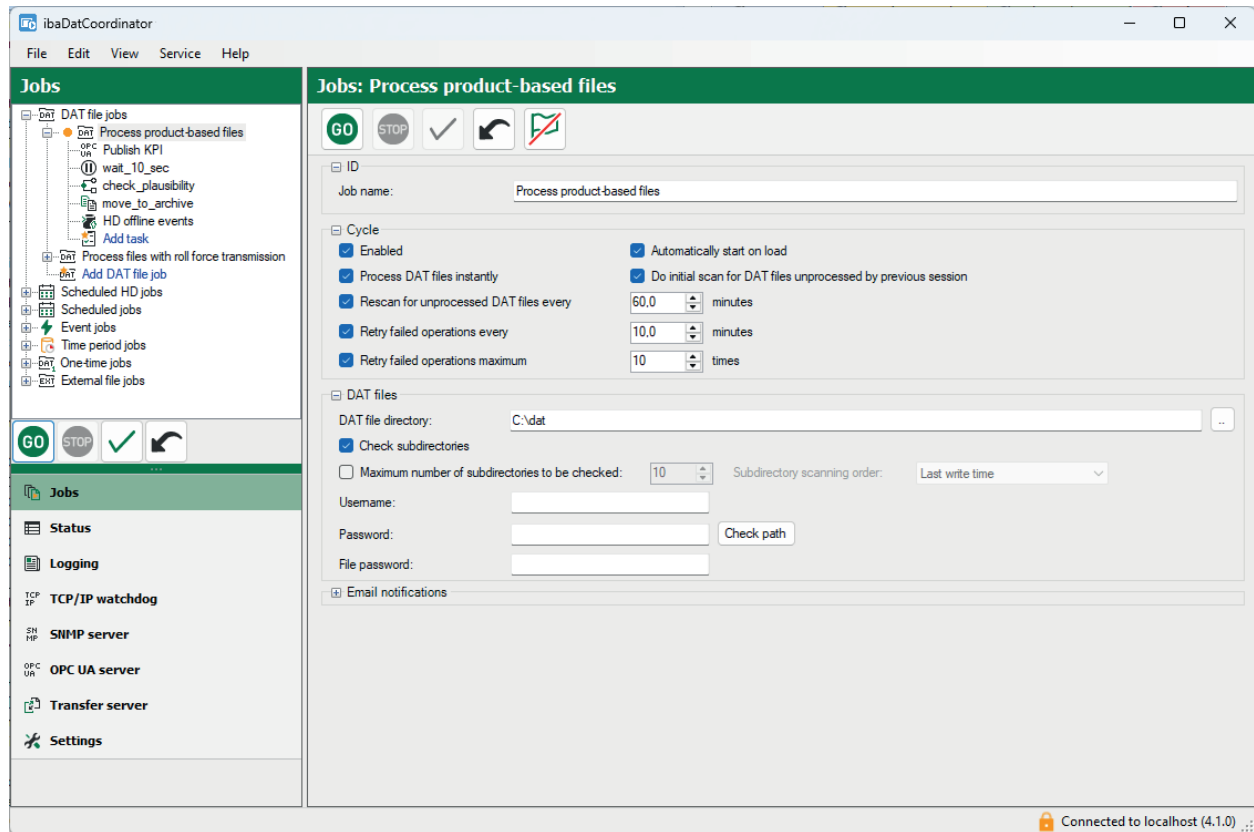


→ *ibaDatCoordinator* validates the configuration and displays the result in a window.

5.2.1 DAT file job – settings

For DAT file jobs, configure the cycle and the directory to be monitored in addition to the default settings.

For information on the default settings, see [↗ Configuring a job, page 59](#). For general information about the DAT file job, see [↗ DAT file job, page 57](#).



Cycle

Here you can set the parameters for job timing. Define when and how often *ibaDatCoordinator* scans the directory for new files. You can select or deselect the following options.

Enabled

The job will only be executed if this option is enabled.

Automatically start on load

When enabled, the job starts automatically as soon as the XML configuration file has been loaded. This is the case if you load the configuration manually via the *File – Open* menu, or if the *ibaDatCoordinator* client or service starts.

Process DAT files instantly

When enabled, *ibaDatCoordinator* processes a new DAT file immediately as soon as it is closed by *ibaPDA*, regardless of when the next scan is scheduled. If you disable this option, new, unprocessed files will only be processed when the directory is scanned again (see field *Rescan for unprocessed DAT files every... minutes*).

If DAT files are saved in the network, *ibaDatCoordinator* might miss newly created DAT files when the network is not active. In this case, iba recommends doing a cyclic or one-time test for non-processed DAT files so that these missing files can be processed.

Note

If you use the *ibaPADU-S-IT* device for writing DAT files on its own hard disk and monitor this hard disk using *ibaDatCoordinator*, the *Process DAT files instantly* option must be disabled.

Do initial scan for DAT files unprocessed by previous session

When enabled, the DAT file directory is scanned for unprocessed DAT files as soon as *ibaDatCoordinator* starts. If you disable this option, new, unprocessed files will only be processed when the directory is scanned again (see field *Rescan for unprocessed DAT files every... minutes*).

If the processing of unprocessed files has no priority, you should disable this option, as this might overload the CPU when the process is started.

Rescan for unprocessed DAT files every ... minutes

This option allows you to specify the interval at which *ibaDatCoordinator* regularly scans the directory for unprocessed DAT files. The time must be at least 0.1 minutes (equivalent to 6 seconds).

If you disable this option, only DAT files that are picked up by the *Process DAT files immediately* option will be processed.

Tip

Disable this option if you only want to process the current DAT files, e.g., to generate a report or publish calculated values.

Activate this option if you want to process all DAT files (possibly with a delay).

Retry failed operations every ... minutes

Set the time after which tasks on DAT files marked as failed will be performed again.

ibaDatCoordinator scans the directory at regular intervals to perform the configured tasks on files marked as failed.

If you disable this option, only one attempt will be made to process a file and no directory scan will be performed.

Retry failed operations maximum ... times

Here you can define how often a DAT file is reprocessed after a failure. If the maximum number of retry attempts has been made without success, the DAT file is added to a file list of *Continuously failed DAT files*.

If you disable this option and the *Retry failed operations every ... minutes* option is enabled, *ibaDatCoordinator* will not stop trying to process the files marked as failed.

DAT files

DAT file directory

Specify the directory where the data files to be processed are located. You can enter the name of the directory or select the directory using the browser button. If a network drive is to be accessed, you can enter the network path in UNC notation with user name and password here, for example. To test whether *ibaDatCoordinator* can access the specified directory, click on the <Check path> button.

Note



If the software runs in service mode under the local system account (default), you normally have to enter the user name and password for network drives so that the *ibaDatCoordinator* service process has access to the network.

Note



If you want to use network shares, iba recommends entering the UNC path description (\\Host\network share or directory) to access the file directory.

To store files on the disk of an *ibaPADU-S-IT* device you will have to enter the following:

```
\\<IP-adr. or host name of ibaPADU-S-IT>\RamDisk
```

Optionally, enter "\dat" if it has been configured in the DAT_FILE_WRITE function block of the *ibaLogic* project with the default directory path = C:\dat (from *ibaLogic* V4_v4.2.4 onwards).

The username is "padu" and the default password is "1234."
(You can change the password via the *ibaPADU-S-IT* web interface.)

iba also recommends setting a time interval for the start of DAT file processing after booting, see chapter [➤ Settings](#), page 44.

Check subdirectories

To also monitor the subdirectories of the source directory, enable the *Check subdirectories* option. This is required if the DAT files are stored in subdirectories, that are separated e.g. by year, month, week, day and/or hour.

Maximum number of subdirectories to be checked

You can optionally limit the number of subdirectories to be scanned. In the *Subdirectory scanning order* field, select which subdirectories to include if there are more than the specified number:

- Last write time: The specified number of subdirectories containing the most recent data will be scanned (default setting).
- Numeric directory names: If directories are named according to a numerical scheme, such as "YYYY2 – "MM" – "DD" (multi-level scheme) or "YYYYMMDD" (flat scheme), the specified number of subdirectories with the most recent names or the largest integers in their names will be scanned. Only directories whose names contain an integer are included.

File password

To be able to process password-protected measurement files with *ibaDatCoordinator*, enter the appropriate password for the data files here. The password is automatically saved in the job.

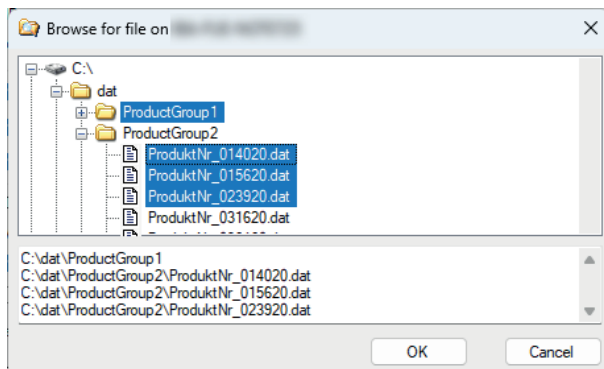
5.2.1.1 Reprocessing DAT files

When DAT files are processed using a DAT file job, *ibaDatCoordinator* marks these DAT files with an info field. To reprocess the files, you must remove the flags.

1. Open the desired DAT file job.

2. At the top of the configuration area click .

→ A file browser will open, displaying the available folders and files on your network.



3. Select one or more folders or files and click <OK>.

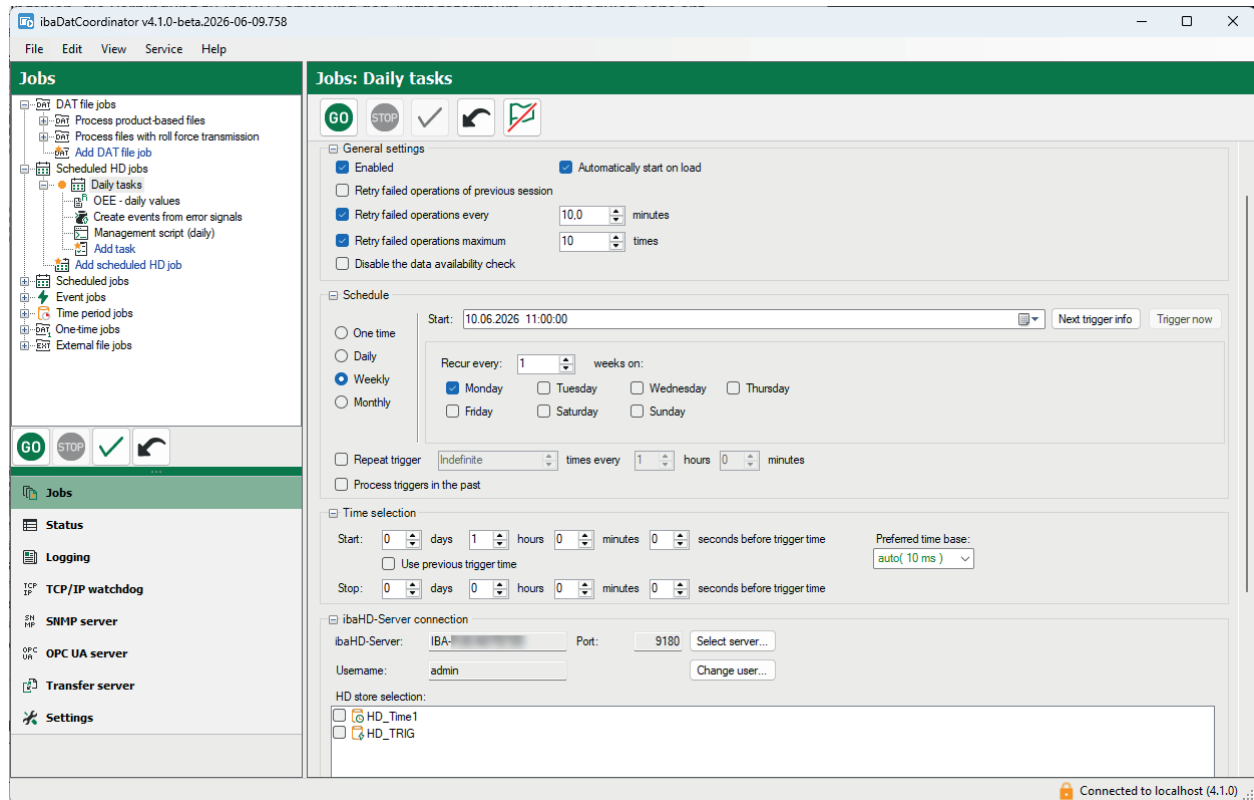
To select multiple items, hold down the <CTRL> key. If the *Check subdirectories* option is enabled in the job configuration, files in subfolders will also be included.

→ *ibaDatCoordinator* removes the corresponding flag from the DAT files and treats them as if they were newly created.

5.2.2 Scheduled HD job and scheduled job – settings

For scheduled HD jobs, configure the processing schedule, the connection to the *ibaHD-Server*, and the query period in addition to the default settings. For scheduled jobs, the server connection settings are not required.

For information on the default settings, see [↗ Configuring a job, page 59](#). Detailed information about the job types can be found in [↗ Scheduled HD job and scheduled job, page 58](#).



General settings

In the *General settings* area, you can make the settings for the starting time and the period of execution.

Enabled

The job will only be executed if this option is enabled.

Automatically start on load

When enabled, the job starts automatically as soon as the XML configuration file has been loaded. This is the case if you load the configuration manually via the *File – Open* menu, or if the *ibaDatCoordinator* client or service starts.

Note



"Starting" does not mean that the tasks configured in the job are executed immediately, but that the job waits for the trigger to start the execution.

Retry failed operations of previous session

Enable this option if you want to ensure that operations that could not be completed in the previous session are repeated.

Retry failed operations every ... minutes

Set the time after which failed tasks, including HD queries, that are marked as failed will be performed again. *ibaDatCoordinator* only performs these tasks again.

Retry failed operations maximum ... times

Here you can define how many times the attempt will be made to reprocess an HD query. If the maximum number of attempts has been performed without success, the HD query is added to a file list of *continuously failed DAT files*.

If you disable this option, *ibaDatCoordinator* will not stop trying to process HD queries marked as failed.

Disable the data availability check

By default, *ibaDatCoordinator* checks whether data is available in the configured HD stores and ignores empty stores. If you enable this option, empty HD stores will also be included in processing.

Schedule

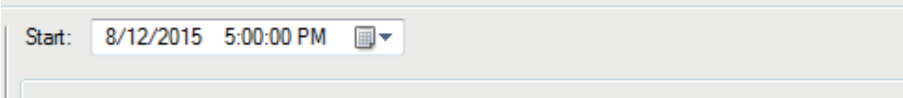
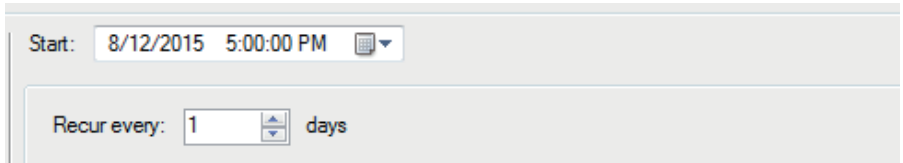
With these settings allow you to specify when *ibaDatCoordinator* should execute a job and process the configured tasks.

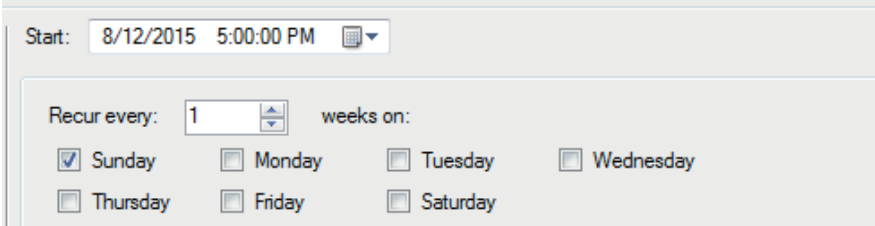
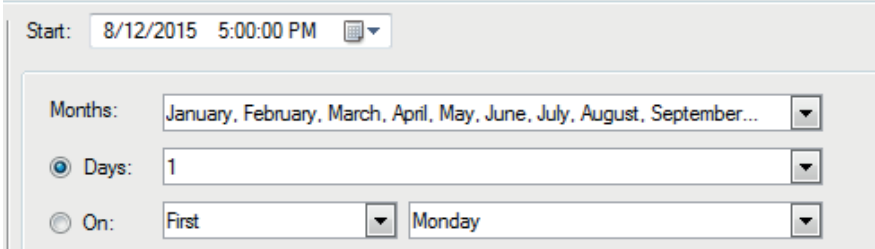
Start

Enter the time for the first job execution here. A calendar function is available for the date; you must enter the time manually in the format HH:MM:SS.

This point in time is decisive for all periodic executions, whether on a daily, weekly or monthly basis.

Then define whether the job is to be executed just once or every day, every week or every month. Depending on the period, additional settings options are displayed:

One time	
Daily	 Enter the intervals, calculated in days, at which the trigger is to be tripped. Number = 1 means every day.

Weekly	 Enter the intervals, calculated in weeks, at which the trigger is to be tripped. Number = 1 means every week. Select the day of the week on which the trigger is to be tripped. You can select multiple options, e.g. on Monday, Wednesday and Friday. Thus, you can configure daily execution week by week.
Monthly	 Select the month(s) in which you want the files to be processed. You can choose between execution on one or more days of the month or on certain weekdays of the month.

Repeat trigger

This option allows you to additionally schedule the job to run on an hourly or minute basis. If this option is enabled, execution will take place according to this fine-tuning within the higher-level planning (daily, weekly, or monthly).

In the first field, select the number of repetitions for the trigger. In the second and third fields, select the time interval at which the trigger is to be repeated.

With the number, determine how often the trigger will be repeated after the first trigger of the day/week/month.

- Number of repetitions = 1: After the first trigger, the trigger will be tripped once more in the set interval.
- Number of repetitions = 2: After the first trigger, the trigger will be tripped twice, each time at the set interval.
- More values corresponding
- Number of repetitions = infinite: After the first trigger, the trigger will be tripped again and again at the set interval until a new (main) trigger is tripped.

This setting is useful if, for example, you want to run a task every 10 minutes over a 24-hour period on a daily basis. In this case, you do not have to calculate the number of repetitions (1440).

Example

With the *Repeat trigger* option, you can have a job executed several times a day, e.g. for creating shift reports.

Select a daily execution, starting at 6:00 a.m. and repeating the trigger every 8 hours. In combination with the time range of 8 hours for the HD query and a report task, you will receive 3 shift reports a day for the times between 10 p.m. and 6 a.m., 6 a.m. and 2 p.m., and 2 p.m. and 10 p.m.

Process triggers in the past

If activated, *ibaDatCoordinator* restarts processing after the last processed time range instead of from the configured start time. If *ibaDatCoordinator* cannot process certain time ranges (e.g. because the server is not available due to maintenance work), processing is continued with the next "unprocessed" trigger instead of waiting for the next trigger and skipping the missed triggers.

To reset the saved time range, click on the  button. Processing starts from the configured start time when the job is restarted.

<Next trigger info>

This button opens a window listing the next 10 calculated trigger times with the corresponding time ranges of the HD queries.

<Trigger now>

For test purposes or if you want to trigger spontaneously, you can click on the <Trigger now> button. This button is only available if the job has been started (<GO>).

Time selection

With the setting in the "Time selection" area, you can determine the time period from which the measurement data are originated.

Start and Stop

Set the start and end of the query period under *Start* and *Stop*. The *Start* must be further in the past than the *Stop*.

Both entries refer to the trigger time.

Use previous trigger time

Select this option to use the time of the last trigger as the start time for the query period. This way, you can reliably implement a complete query. Additionally, set "0" for all fields at *Stop* (from trigger to trigger).

Tip

Activate this option for monthly triggers, for example, if you want to query data for the entire past month. Due to the different numbers of days per month, the query period will vary.

Preferred time base

Similar to HD queries with *ibaAnalyzer*, you can select a preferred time base. Depending on the setting of the time ranges and the number of samples contained therein for the HD query, the range of time bases changes.

If you do not need a specific time base for the query results, use the default setting "auto". This ensures that the query delivers the highest possible resolution, which can be particularly important for subsequent calculations.

ibaHD-Server connection (scheduled HD job only)

Here you can configure the settings for the connection to the *ibaHD-Server* and the HD store. *ibaDatCoordinator* requires this information and the following settings for the time selection for generating the correct HD query (HDQ file), which is then transferred to *ibaAnalyzer* for execution of tasks.

To open the server selection dialog, click the <Select server...> button. Select the desired *ibaHD-Server* from the table and close the dialog by clicking <OK>.

If user management is enabled for the selected *ibaHD-Server*, you must then enter a user name and the correct password.

For more information on *ibaHD-Server connection* and user management, see ➤ *Establishing ibaHD-Server connection, page 157*

Now select the desired HD store where the measurement data is stored under *HD store selection*. If you select multiple stores, the query criteria are applied to all selected stores. All data in the result of the HD query is processed by the job.

Note

If you want to execute cyclical tasks that do not require measurement data, such as cleanup or script tasks, leave the *ibaHD-Server* field blank in the *ibaHD-Server connection* area.

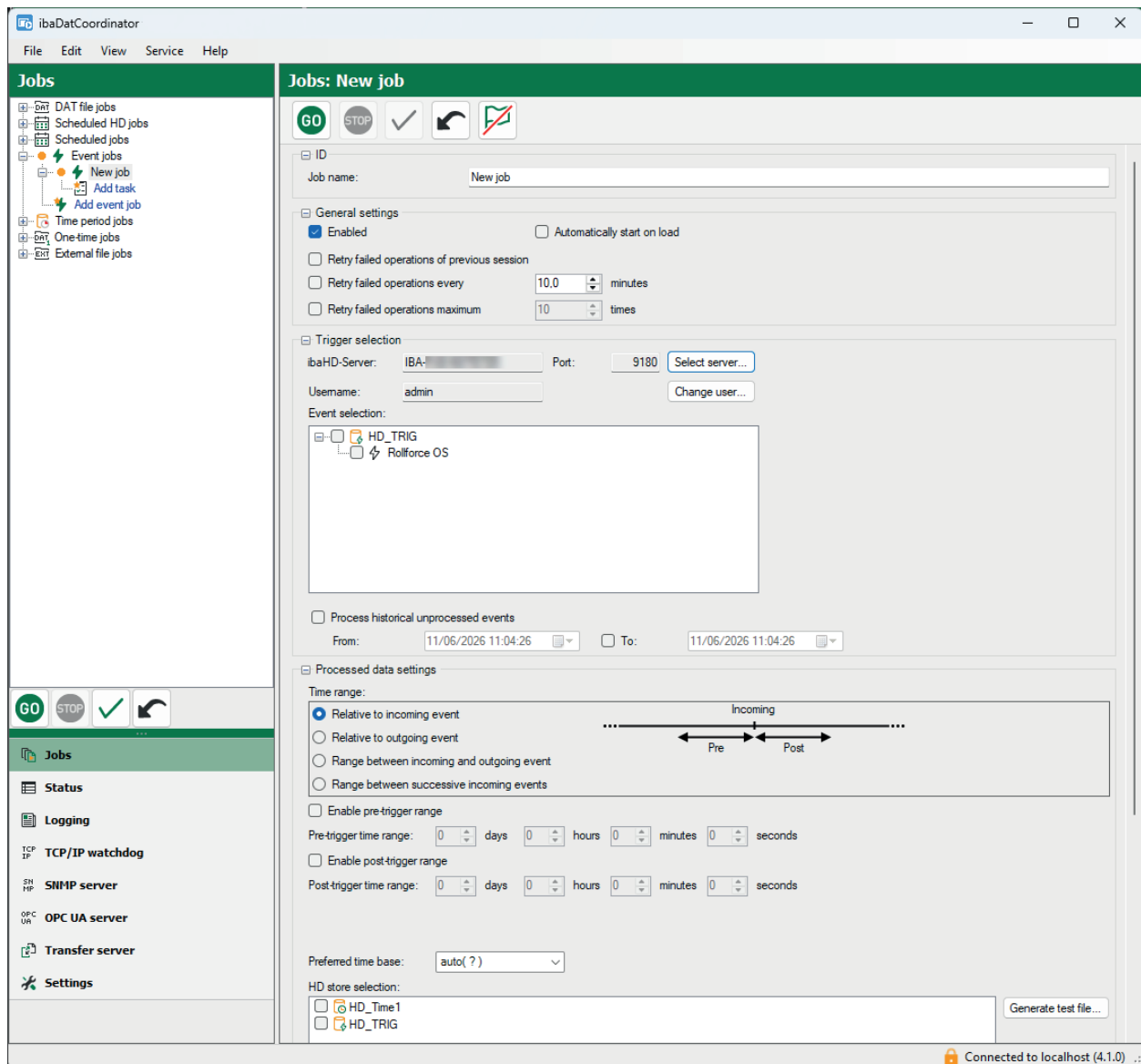
Other documentation

Detailed information on the function, configuration, and querying of an *ibaHD-Server* can be found in the corresponding documentation.

5.2.3 Event job – settings

For event jobs, configure the connection to the *ibaHD-Server*, the triggers, and the source of data processing in addition to the default settings.

For information on the default settings, see [↗ Configuring a job, page 59](#). General information about the job type can be found in [↗ Event job, page 58](#).



General settings

Enabled

The job will only be executed if this option is enabled.

Automatically start on load

When enabled, the job starts automatically as soon as the XML configuration file has been loaded. This is the case if you load the configuration manually via the *File – Open* menu, or if the *ibaDatCoordinator* client or service starts.

Note

"Starting" does not mean that the tasks configured in the job are executed immediately, but that the job waits for the trigger to start the execution.

Retry failed operations of previous session

Enable this option if you want to ensure that operations that could not be completed in the previous session are repeated.

Retry failed operations every ... minutes

Set the time after which tasks marked as failed will be retried.

Retry failed operations maximum ... times

Here you can define how many times the attempt will be made to reprocess time periods.

If you disable this option and the *Retry failed operations every ... minutes* option is enabled, *ibaDatCoordinator* will not stop trying to process the data marked as failed.

Trigger selection**ibaHD-Server**

To open the server selection dialog, click the <Select server...> button. Select the desired *ibaHD-Server* from the table and close the dialog by clicking <OK>.

If user management is enabled for the selected *ibaHD-Server*, you must then enter a user name and the correct password.

For more information on the *ibaHD-Server* connection and user management, see [🔗 Establishing ibaHD-Server connection, page 157](#).

Once the connection to the *ibaHD-Server* has been established, you will see the event-based HD stores and the available events under *Event selection*. Select at least one event for the configuration to be valid. The job is executed each time a selected event occurs.

Process historical unprocessed events

When enabled, all unprocessed events that occur after the defined start time (*From*) or within the defined time range *From/To*.

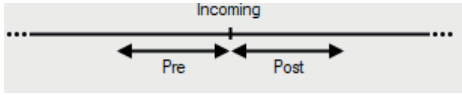
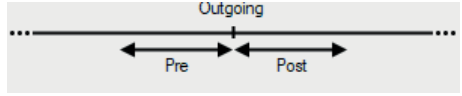
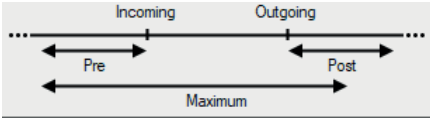
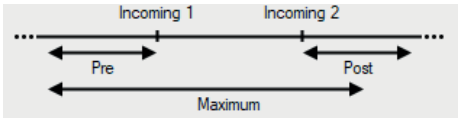
To reset the saved time range, click the button . Processing begins at the configured start time when the job is restarted.

Processed data settings

Here you specify the source for data processing, i.e., the HD store with the desired data, response behavior, and time range.

Time range

Here, you can specify exactly when the job should be executed for an event. Choose from the following options:

Option	Description
Relative to incoming event 	The job is executed at the "start" of a selected event (e.g., rising edge).
Relative to outgoing event 	The job is executed at the "end" of a selected event (e.g., falling edge).
Range between incoming and outgoing event 	This option only makes sense if the job is to generate HDQ files. In this case, the HDQ files also contain the time range between incoming and outgoing events.
Range between successive incoming events 	The job runs from the "start" of one event until the "start" of the next event.

Similar to conditional HD queries in *ibaAnalyzer*, you can also set pre-trigger times and post-trigger times here. The times are relative to the monitored event moments.

Enable pre-trigger range

Select this option if you want to query a period of time before an event. You can set the length of the period directly below. When the job is executed, the data from the period up to the specified event (incoming or outgoing) is then queried.

Enable post-trigger range

Select this option if you want to query a period of time after an event. You can set the length of the period directly below. When the job is executed, the data from the period starting from the specified event (incoming or outgoing) is queried.

Maximum time range

The setting for this time range is only available if you have selected the trigger option *Range between incoming and outgoing events* or *Range between successive incoming events*.

The maximum time range serves as a limit in the event that an incoming event has occurred but the outgoing or subsequent incoming event has not occurred before the maximum duration has expired. Note that the maximum time range includes the pre-trigger time and post-trigger time.

Preferred time base

Similar to HD queries with *ibaAnalyzer*, you can select a preferred time base. Depending on the setting of the time ranges and the number of samples contained therein for the HD query, the range of time bases changes.

If you do not need a specific time base for the query results, use the default setting "auto". This ensures that the query delivers the highest possible resolution, which can be particularly important for subsequent calculations.

HD store selection

Select the HD stores from which data is to be retrieved. If available, you can select multiple stores.

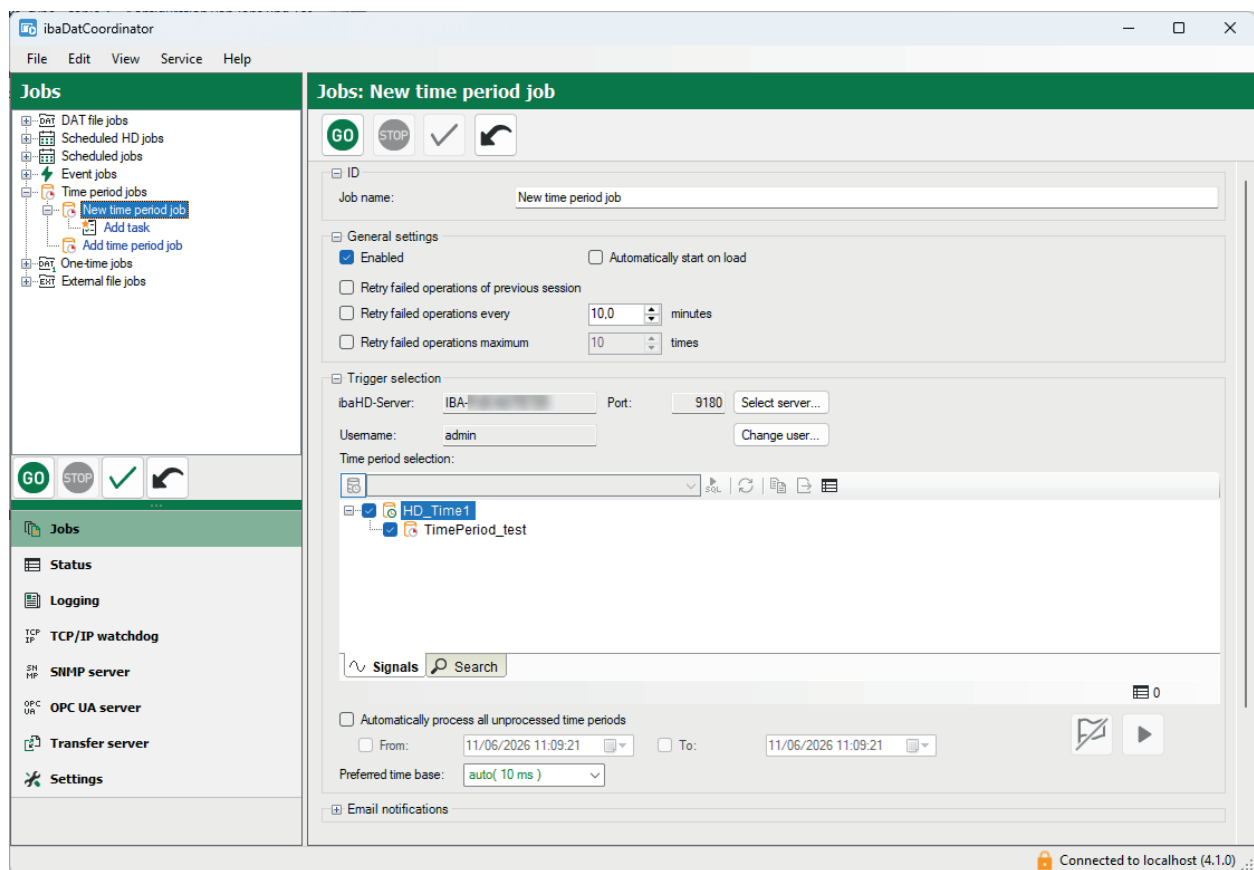
For testing purposes, you can generate an HDQ file by clicking on the <Generate test file> button. In order for such a file to be created, any occurrence of one of the specified events is taken, or, if no event occurrence is found, the current time.

Once you have completed these settings, you can add the desired tasks to the job. The tasks then process the measurement data contained in the HDQ file.

5.2.4 Time period job – Settings

For time period jobs, configure the connection to the *ibaHD-Server*, the time period store, and the query period in addition to the default settings.

For information on the default settings, see [↗ Configuring a job, page 59](#).



General settings

Enabled

The job will only be executed if this option is enabled.

Automatically start on load

When enabled, the job starts automatically as soon as the XML configuration file has been loaded. This is the case if you load the configuration manually via the *File – Open* menu, or if the *ibaDatCoordinator* client or service starts.

Retry failed operations of previous session

Enable this option if you want to ensure that operations that could not be completed in the previous session are repeated.

Retry failed operations every ... minutes

Set the time after which failed tasks marked as failed with time periods will be performed again. *ibaDatCoordinator* only performs these tasks.

Retry failed operations maximum ... times

Here you can define how many times the attempt will be made to reprocess time periods. If the maximum number of attempts has been performed without success, the corresponding task is added to a file list of "constantly failed tasks".

If you disable this option, *ibaDatCoordinator* will not stop trying to process tasks marked as failed.

Trigger selection

In the *Trigger selection* area, make the settings for the connection to the *ibaHD-Server* and time period store.

To open the server selection dialog, click the <Select server...> button. Select the desired *ibaHD-Server* from the table and close the dialog by clicking <OK>.

If user management is enabled for the selected *ibaHD-Server*, you must then enter a user name and the correct password.

For more information on the *ibaHD-Server* connection and user management, see ➤ *Establishing ibaHD-Server connection, page 157*.

Under *Time period selection*, select the desired time periods to be processed. For more information on how to use the time period table and its filter options, see ➤ *Properties of the time period table, page 75*.

Automatically process all unprocessed time periods

By default, *ibaDatCoordinator* automatically processes new period periods finished by *ibaPDA*. If this option is activated, existing time periods (starting from the configured time) can also be processed.

If the "To" option is not set, *ibaDatCoordinator* will continue to process the new time periods as usual. After clicking <GO>, the corresponding time periods are processed and the option is reset to be inactive.

Preferred time base

A time period period is usually processed with *ibaAnalyzer*. Therefore, the preferred time base used to query time-based data must be set with *ibaAnalyzer*.

Control elements

	Removes the markers from the time periods that they have already been processed. To use it, it is necessary to set the <i>Process existing time periods</i> option and set the time periods accordingly. Only time periods within the configured time range are re-set. After the marker has been removed, the time periods can be processed again.
	While the job is running, individual time periods can be selected from the table for immediate processing. The time periods are always processed immediately, regardless of whether they have already been marked as processed. If the flag was not already set, it will be set.

5.2.4.1 Properties of the time period table

The time period table offers advanced configuration options in the time period selection area. To open the time period table, click on .

Time-period selection:

 select_timeperiods

Name	Start time	End time	Sin1	Sin2	sin4
trig3_613	11/20/23 9:44:02 AM	11/20/23 9:44:06 AM	0.9929371	0.8247603	0.9830312
trig3_612	11/20/23 9:43:52 AM	11/20/23 9:43:57 AM	-0.536811	-0.6600611	-0.4471461
trig3_611	11/20/23 9:43:41 AM	11/20/23 9:43:46 AM	-0.3036958	0.4467436	-0.7337569
trig3_610	11/20/23 9:43:31 AM	11/20/23 9:43:36 AM	0.9260436	-0.2039042	0.8574792
trig3_609	11/20/23 9:43:21 AM	11/20/23 9:43:26 AM	-0.8831706	-0.05240971	0.2542355
trig3_608	11/20/23 9:43:10 AM	11/20/23 9:43:15 AM	0.2058745	0.3052602	-0.9996533
trig3_607	11/20/23 9:43:00 AM	11/20/23 9:43:05 AM	0.6193109	-0.5379385	0.3047927
trig3_606	11/20/23 9:42:50 AM	11/20/23 9:42:55 AM	-0.9996161	0.7350684	0.8292065
trig3_605	11/20/23 9:42:39 AM	11/20/23 9:42:45 AM	0.6618499	-0.8836234	-0.7685033

500 [more...](#)

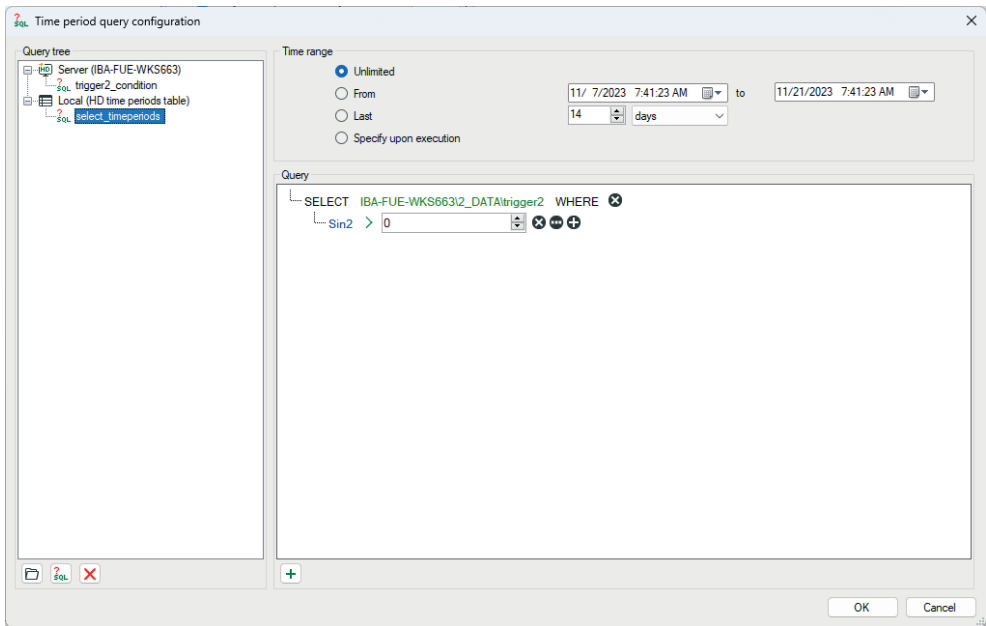
The table shows the latest time periods from the defined time range. You can sort and filter the table and display calculated columns. The filter has no effect on the processing.

Controls for the time period table

	Shows the latest time periods
	Opens the editor to create SQL queries
	Executes the SQL query selected in the selection list
	Copies the time periods marked in the table to the clipboard. Only possible in pause mode.
	Exports the time periods marked in the table to an Excel or text file. Only possible in pause mode.
	Shows and hides the time period table.

By default, all time periods in the selected time period storage are processed. By using SQL type queries, specific time periods can be processed.

To open the query editor, click on  **SQL**. You can add local queries that are stored in the job configuration, or store queries on the *ibaHD-Server* so that all connected clients can use the same server queries.



Select a query from the list. To execute the query, click on . The table then displays the time periods that meet this condition. When the job starts, only the time periods queried here are processed.

Time-period selection:

Name	Start time	End time	Sin2	sin4
trig2_2167	11/20/23 9:43:57 AM	11/20/23 9:44:03 AM	0.000214545	0.6488556
trig2_2165	11/20/23 9:43:43 AM	11/20/23 9:43:49 AM	0.008685051	0.7967904
trig2_2163	11/20/23 9:43:29 AM	11/20/23 9:43:35 AM	0.00825568	0.9059212
trig2_2161	11/20/23 9:43:15 AM	11/20/23 9:43:21 AM	0.007826308	0.9746464
trig2_2159	11/20/23 9:43:01 AM	11/20/23 9:43:06 AM	0.007396935	0.9999009
trig2_2157	11/20/23 9:42:47 AM	11/20/23 9:42:52 AM	0.00696756	0.9805583
trig2_2155	11/20/23 9:42:33 AM	11/20/23 9:42:38 AM	0.006538185	0.9174812
trig2_2153	11/20/23 9:42:19 AM	11/20/23 9:42:24 AM	0.006108807	0.8134829
trig2_2151	11/20/23 9:42:04 AM	11/20/23 9:42:10 AM	0.00567943	0.6732021

500 [more...](#)

Other documentation

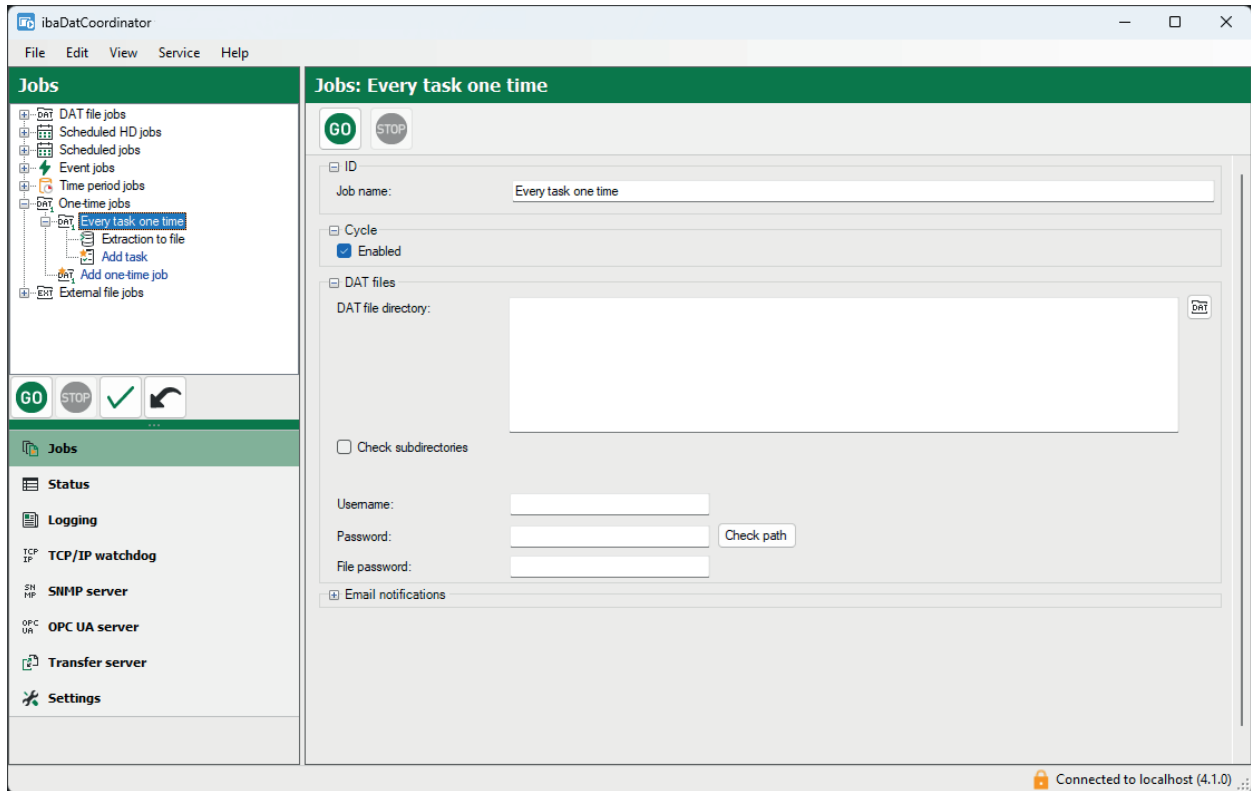


A detailed description of how to configure time period queries can be found in the *ibaHD-Server* documentation.

5.2.5 One-time job – settings

The settings available within one-time jobs are largely the same as those for DAT file jobs, see 61.

For information on the default settings, see ↗ *Configuring a job, page 59*.



Cycle

Enabled

The job will only be executed if this option is enabled.

DAT files

For one-time processing, you can select individual .dat files or directories containing .dat files. You can drag and drop the desired files or folders from Windows Explorer into the *DAT file directory* field, or click . A file browser will open, allowing you to select one or more files or directories.

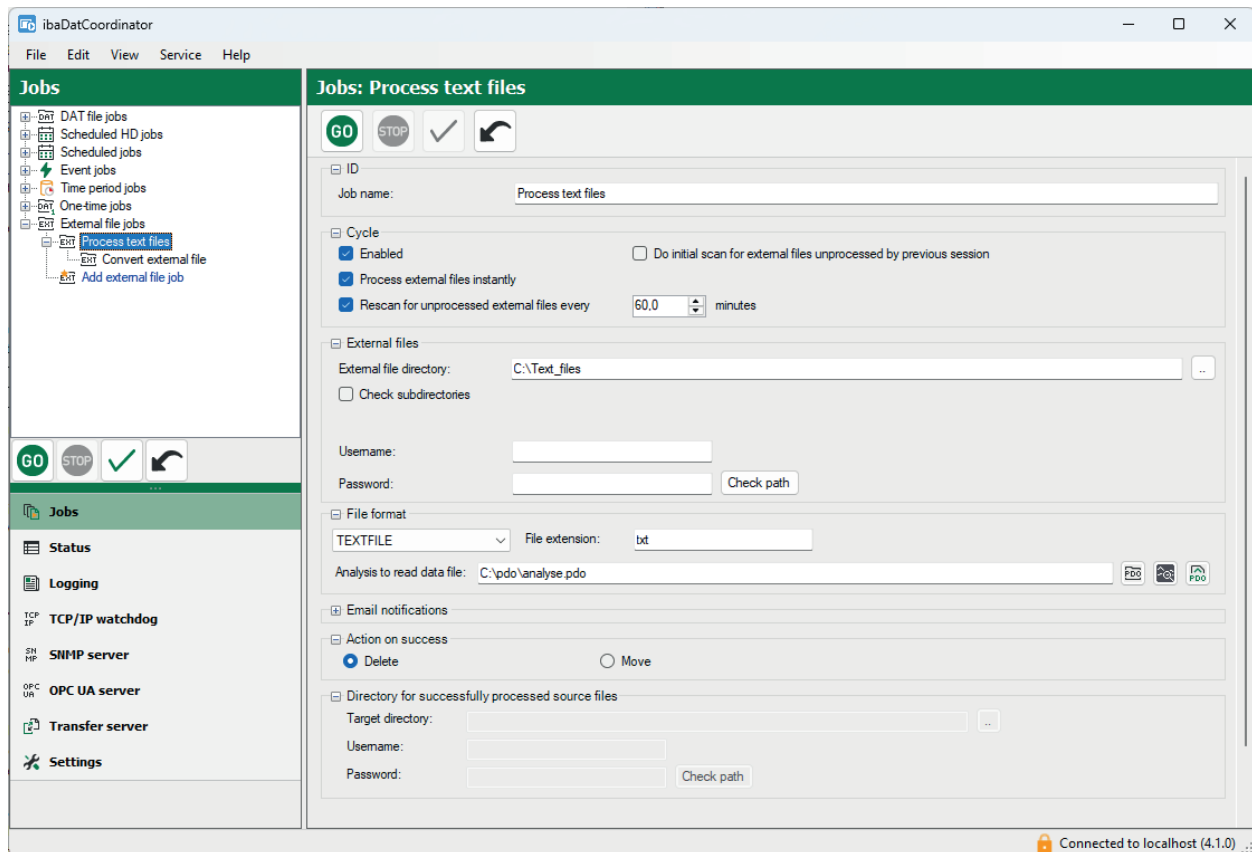
If the files are located on a network drive, enter your username and password to access them, if necessary.

To check the login data for the source directory, click <Check path>, optionally taking subdirectories into account.

5.2.6 External file job – settings

For external file jobs, in addition to the default settings, configure the cycle, the directory to be monitored, and the file format to be converted.

For information on the default settings, see [↗ Configuring a job, page 59](#).



Cycle

Enabled

The job will only be executed if this option is enabled.

Process External file files instantly

When activated, *ibaDatCoordinator* immediately begins processing the closed external file. If you disable this option, new, unprocessed files will only be processed when the directory is scanned again (see field *Rescan for unprocessed files every... minutes*).

Do initial scan for external files unprocessed by previous session

When enabled, the file directory is scanned for unprocessed files as soon as *ibaDatCoordinator* starts. If you disable this option, new, unprocessed files will only be processed when the directory is scanned again (see field *Rescan for unprocessed files every... minutes*).

If the processing of unprocessed files has no priority, you should disable this option, as this might overload the CPU when the process is started.

Rescan for unprocessed external files every... minutes

This option allows you to specify the interval at which *ibaDatCoordinator* regularly scans the directory for unprocessed files.

If you disable this option, only external files that are picked up by the *Process external files immediately* option will be processed.

Tip

Disable this option if you only want to process the current files, e.g., to generate a report or publish calculated values.

Activate this option if you want to process all DAT files (possibly with a delay).

External files

Specify the source directory where the external files to be processed are located. You can enter the name of the directory or select the directory using the browser button. If a network drive is to be accessed, you can enter the network path in UNC notation with user name and password here, for example. To test whether *ibaDatCoordinator* can access the specified directory, click on the <Check path> button.

File Format

Select the file format to be processed. Only files with the corresponding format (and *File extension*) are processed. The following formats are possible:

- TEXTFILE
- DAS
- COMTRADE
- PARQUET

Several file extensions are possible for text files (*.txt or *.csv). You can enter these manually in the *File extension* field. For text formats, an analysis file is also required for correct reading. All other formats are automatically interpreted correctly by *ibaAnalyzer*.

Note

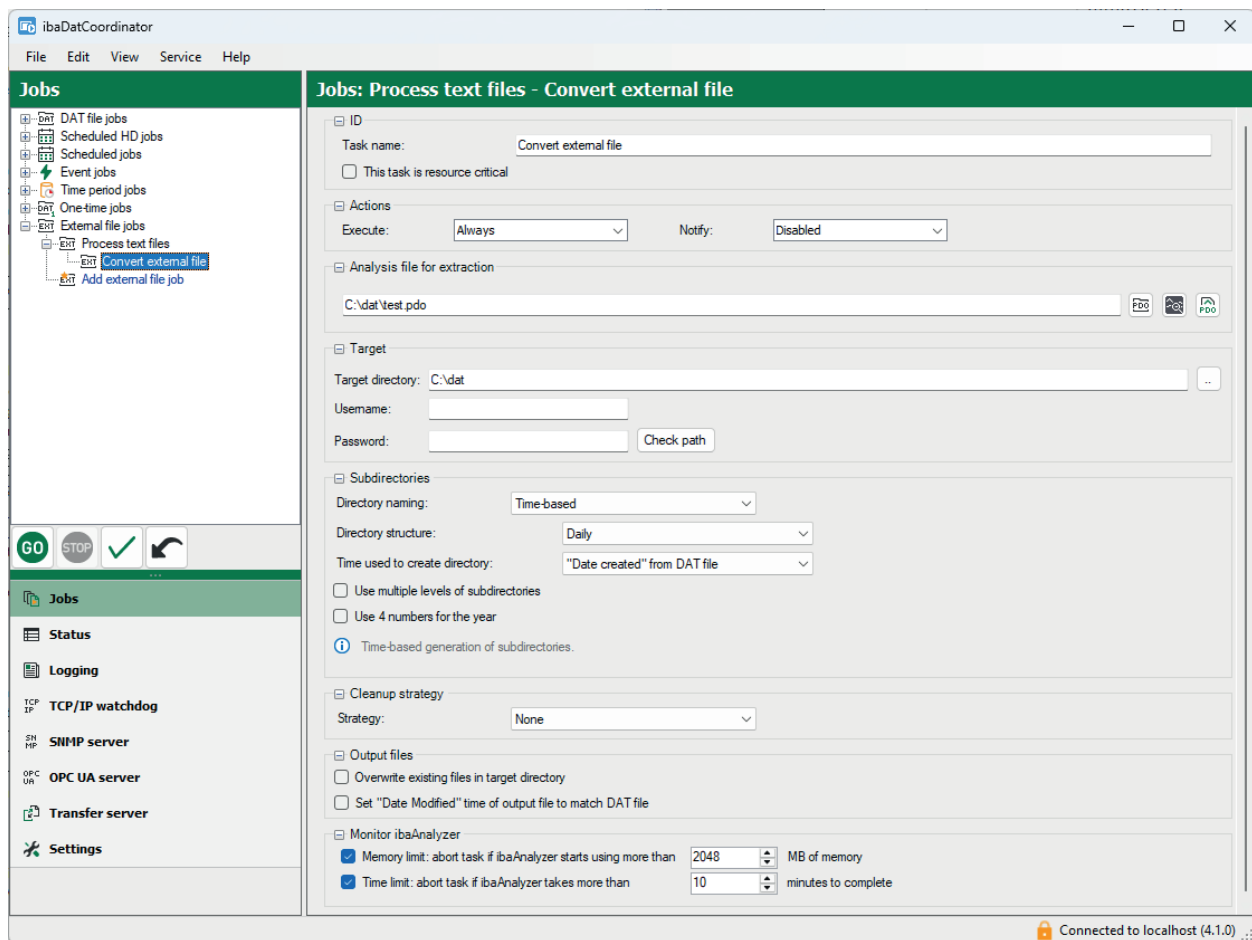
In the "reading" PDO file, the configuration for reading the text file must also be saved in the analysis.

Since, unlike DAT files, no information can be stored in the files as to whether the file has already been processed by *ibaDatCoordinator*, a different mechanism is used. For successfully processed files, you can specify in the *Action on success* section whether the original file should be deleted or moved to another folder. You can configure the path to the *Move* option.

Another special feature of the external file job is that there is only one possible task for converting to DAT format, which is added automatically.

5.2.6.1 Convert external file task – settings

A "Convert external file" task is automatically created for each job. You cannot create any further tasks in the job, as the job is intended solely for converting files.



Analysis file for extraction

In a PDO file, you can define the desired extraction profiles, including selecting the signals and sampling rate. If you do not specify a PDO file, all signals will be written to the DAT file in full resolution.

5.3 Overview of task types

Below you will find an overview of the available task types in *ibaDatCoordinator*.

Symbol	Task	Description
	Copy/move/delete	Copies or moves data files to another directory, or deletes them
	Transfer	Transfers data files to another <i>ibaDatCoordinator</i> instance
	Upload	Uploads data files to various data storage servers

Symbol	Task	Description
	Report	Generates a report based on an analysis with the <i>ibaAnalyzer</i> report generator
	Condition	Forms a branch during task processing using definable criteria based on the data in the data file/HD query
	Cleanup	Deletes files with a configurable file name extension from a directory
	Pause	Interrupts processing for a configurable period of time
	Split	Splits a data file into two or more sub-files based on a condition in an analysis file
	Script	Executes any scripts
	HD offline event	Creates an event based on a data file or an HD query together with rules defined in an analysis file and writes it to an HD event store
	HD import	Imports data files into an HD store in a <i>ibaHD-Server</i>
	Extract to file	Automatically extracts data using the <i>ibaAnalyzer</i> data extractor into one of the supported file formats (DAT file, ASCII, COMTRADE, TDMS, PARQUET, MATLAB) Additional license required, see ↗ Licenses , page 9.
	Extract to DB	Automatically extracts data using the <i>ibaAnalyzer</i> data extractor into one of the supported databases (MSSQL, Oracle, PostgreSQL, MySQL/MariaDB, IBM DB2, SQLite) Additional license required, see ↗ Licenses , page 9.
	Publish to DB	Extracts only calculated columns and info field columns using the <i>ibaAnalyzer</i> data extractor to one of the supported databases (MSSQL, ORACLE, PostgreSQL, MySQL/MariaDB, IBM DB2, SQLite) Additional license required, see ↗ Licenses , page 9.
	Kafka publish	Publishes values calculated on the basis of an analysis file to an Apache Kafka cluster Additional license required, see ↗ Licenses , page 9.
OPC UA	OPC UA publish	Publishes values calculated on the basis of an analysis file using the integrated OPC UA server Additional license required, see ↗ Licenses , page 9.
SN MP	SNMP publish	Publishes values calculated on the basis of an analysis file using the integrated SNMP server Additional license required, see ↗ Licenses , page 9.

Symbol	Task	Description
	Time period update	Writes or updates values of existing period periods Additional license required, see ↗ Licenses , page 9.
	Update	Writes subsequently defined data from a database to a data file Additional license required, see ↗ Licenses , page 9.
	S7 writer	Writes values from data files or analyses to an S7 data block Additional license required, see ↗ Licenses , page 9.
	Merge	Merges several DAT files into one DAT file Additional license required, see ↗ Licenses , page 9.
	DB sync	Synchronizes database and file system Additional license required, see ↗ Licenses , page 9.

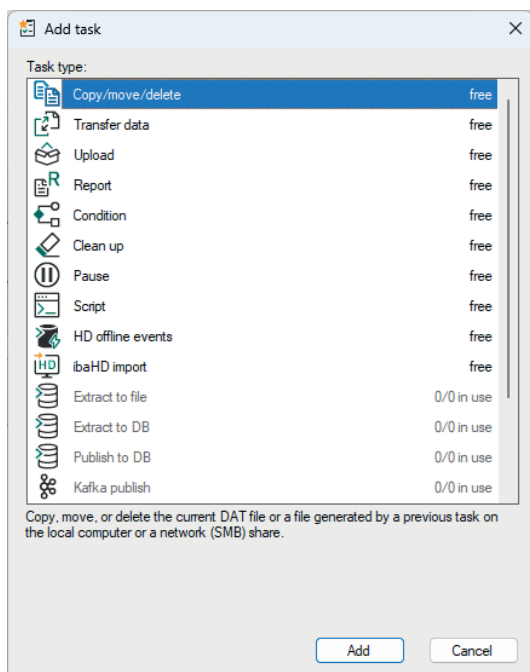
5.4 Configuring a task

To add tasks to a job, follow these steps.

Prerequisite: The job has been stopped.

1. Decide which task type you need, see [↗ Overview of task types](#), page 80.
2. In the job tree, click the “Add task” link below the job, or select the command from the job's context menu.

→ The selection dialog opens and displays all task types permitted for that job type.



All license-free task types and those for which you have licenses are active. All other task types are grayed out. The right-hand column shows the respective license status. When you select a task type, a description of the task will also appear at the bottom of the window.

For further information on task types and plug-ins requiring a license, see [🔗 Licenses](#), page 9.

3. Select the desired task type and confirm with <Add>.

→ The new task is added at the last position under the job node.

4. Adjust the order of the tasks as needed.

The order of the tasks is important because it corresponds to the processing sequence from top to bottom. Pay particular attention to the correct order if the execution of a task depends on the result of a previous task.

5. Select the added task node in the tree.

→ The settings options for the task appear in the display and configuration area.

6. Configure the basic settings for the task, see [🔗 General task settings](#), page 83.

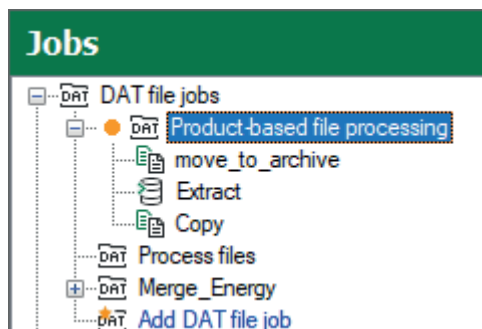
All tasks share the settings areas ID and Actions. The target settings relevant for file processing in some tasks are also described in the basic settings.

7. Configure the task-specific settings.

8. If necessary, repeat the steps for additional tasks until all tasks for the job have been created.

9. To apply the changes to the configuration, click in the parent job or to the left above the navigation pane on .

You can identify jobs with unsaved configuration changes by the orange dot in the job tree:



5.4.1 General task settings

All tasks in *ibaDatCoordinator* share the settings areas *ID* and *Actions*. Below you will find a detailed description of the corresponding fields. In addition, the settings options in the *Target* and *Subdirectories* area are described, which are also relevant for various task types.

ID

Task name

Enter a descriptive name. By default, the name of the task type is entered, e.g., "Copy/Move/Delete".

This task is resource critical

With this option, you can mark a task as resource critical.

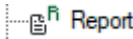
In the *ibaDatCoordinator* settings, you can then limit the number of resource-critical tasks to be processed simultaneously in order to prevent the system from freezing due to overload, see [➤ Settings – Processing, page 44](#).

Tasks with a high demand on resources include report or extract tasks that are processed in *ibaAnalyzer* with very extensive and complex analysis rules and/or high data volumes.

If you enable this option, the associated task will only be executed if, at the time of execution, the number of active tasks with high resource requirements does not exceed the set maximum.

Actions

Execute

Option	Symbol	Meaning
Disabled		The task is disabled and will not be executed.
Always		The task is always executed. If the execution fails, it will be retried in the next cycle.
On success		The current task is executed if the previous task has been executed successfully or the previous condition is TRUE.
On failure		The current task is executed if the previous task failed or the previous condition is FALSE.
On 1st failure*		The task is only executed if the previous task failed for the first time or if the result of a previous condition is FALSE for the first time.

* Option is only available if an active previous task is defined.

Notify

Specify in which cases a notification should be sent. You can specify the general settings for recipients and senders, as well as whether notifications should be sent for the job, in the job settings, see [➤ Configuring a job, page 59](#).

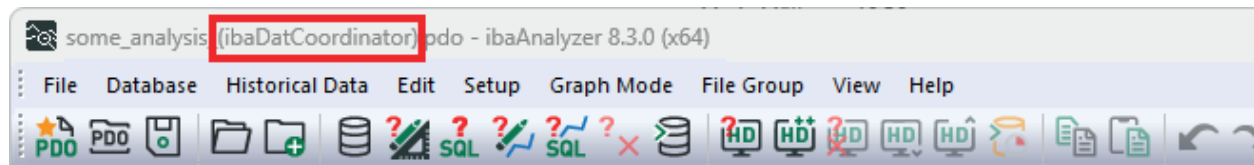
Option	Meaning
Disabled	No notification will be sent.
Always	Whenever a DAT file is processed, a notification is sent regardless of the task status.
On success	If the current task is executed successfully or the result of a condition is TRUE, a notification will be sent.
On failure	If the current task fails or the result of a condition is FALSE, a notification will be sent.
On 1st failure task	For a job with several tasks a notification is sent if the first task fails for the first time or the first condition is FALSE for the first time.
On 1st failure file	For a job with multiple tasks, a notification is sent if any of the tasks fails for the first time or if a condition evaluates to FALSE for the first time.

Task analysis file

For some tasks, you can select an analysis file for configuring the task. The following options are then available:

	Opens the file browser where you can select the analysis file.
	Opens a copy of the selected analysis file with <i>ibaAnalyzer</i> .
	Uploads the updated file to the server and applies all changes.

When you open an analysis file in *ibaAnalyzer*, *ibaDatCoordinator* creates a local copy of the PDO file with the suffix "_(*ibaDatCoordinator*)" in the file name:



Changes to the analysis file are initially saved in the local copy. As soon as you close the PDO file in *ibaAnalyzer*, *ibaDatCoordinator* detects whether changes have been made. A corresponding message appears.

To apply the changes to the server and the original file, click .

Note



The file with the suffix "_(*ibaDatCoordinator*)" is a local copy that only serves to ensure access to the original file while you are working on the analysis file in *ibaDatCoordinator*. The copy is overwritten each time you make changes to the PDO file in *ibaDatCoordinator*.

Target and Subdirectories

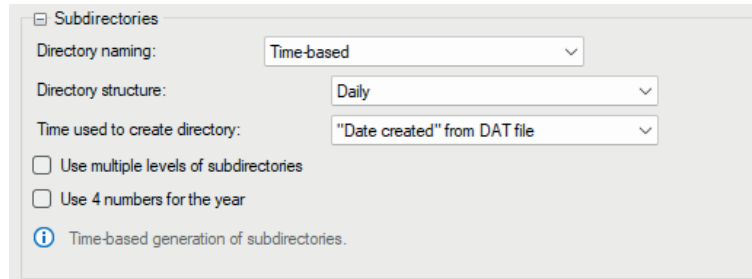
When you configure a copy/move/delete task or create a file in a report, extract, update, or split task, you must define a target directory. To access a network drive, you can also enter the login details of a user with write permissions, if necessary.

In *ibaDatCoordinator*, there are several options for defining the storage of the processed files in detail, such as the organization of subdirectories.

None

The files are stored under the defined path (target directory).

Time-based



Additional subdirectories will be created at the defined path. The names of the subdirectories are created on the basis of the date or time (YYMMDD). Each subdirectory contains the files which were created in the specified time period.

Example

For example, a subdirectory created for a specific day would be named 250821 for the 21/08/2025. The files for the following day would then be saved in the directory 250822.

If you want to save the files on a weekly basis, the system calendar (Windows) is consulted for determining the week. As the week number can be different depending on the country, more parameters have to be defined.

- First day of the week (e.g. Europe = Monday, USA = Sunday)
- Start time of the week (at which time the new week starts on the first day)
- The first week of the year (the first week around January 1st, the first whole week of the year or the week with the first four days of the year).

These settings should be adjusted in compliance with the regional rules to get a proper week counting.

Time stamp for the name of the directory

This setting determines how the directory name for storing the task result is formed. Choose from the options offered:

- "Date created" from DAT file
 - "Date modified" from DAT file
 - Current system time
 - "Start time" in DAT file
- To use the start time on the computer on which *ibaDatCoordinator* is running, *ibaAnalyzer* version 6.5.0 or later must be installed.

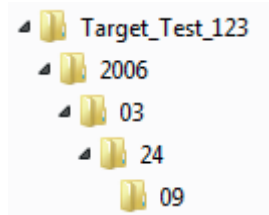
Use multiple levels of subdirectories

If you enable this option, all time classes are stored in cascading directories, beginning with the highest class, e.g. year – month – day – hour for hourly subdirectories or year – month – week for weekly subdirectories.

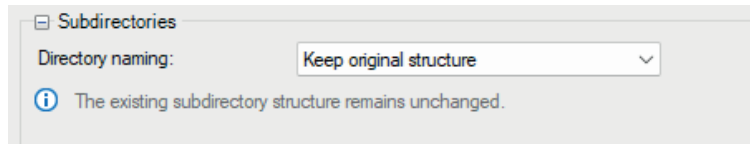
Use 4 numbers for the year

When enabled, the year in the folder name is formatted as YYYY; otherwise, it is two digits.

Subdirectories with several levels and a year with 4 numbers



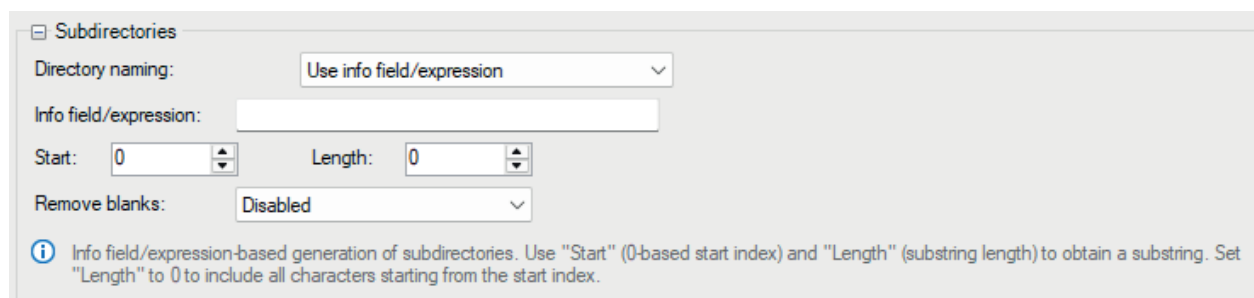
Keep original structure



The files are stored under the defined path. The structure of the directory corresponds to the structure of the source directory the .dat files are stored in.

Use info field/expression

Select this option to use the contents of one or more info fields in a DAT file for naming subdirectories. This allows you to customize and dynamically design directory names and store them separately according to customer name, lot number, material designation, or other criteria.



Single info field:

- Enter the name of the info field in the input field and optionally specify the part to be used by entering the start index and length (number of characters). If you want to use the entire content of the info field, leave the start index and length set to 0.
- Optionally, choose whether to remove spaces at the end or all spaces from the string in the info field.
- For structured info fields, make sure to separate the string sections with a period when entering the info field name.

Instead of a single info field, you can also enter an expression with text functions from the expression builder.

Examples of text functions

ToText(...) converts a numeric value into text.

Concat(...) appends multiple texts, thereby linking multiple info fields into a single text.

You can find an example of how this is used in ➤ *Example: Using info fields for directory/file names, page 154*

Other documentation



Further information on the expression builder can be found in the *ibaAnalyzer* documentation, Part 3 "Expression builder".

Cleanup strategy

You can define a cleanup strategy for all tasks that generate an output file, such as copying files, extracting to a file, and creating a report as a file. With every execution of the task, the cleanup conditions are checked and, if required, the oldest files in the target directory are deleted.

Note



The cleanup strategy only takes DAT files into account. For cleaning up other file types, e.g. PDF, please configure a cleanup task.

Note



In order to prevent overlapping cleanup actions, disable the task-specific clean-up strategies if you want to use the global cleanup (see chapter ➤ *Settings, page 44*).

None

If you choose this option, data are written into the target directory until there is no free disk space left.

Limit subdirectories

Maximum number of subdirectories to be created for this task. When the maximum is reached, the oldest folder will be removed first.

This is only relevant if a subdirectory structure has been selected.

Limit disk space usage

Size of the hard disk space in megabytes (MB) that should be available for the respective directory. When the limit is reached, the oldest target files are deleted or overwritten.

Minimal free disk space

Minimum amount of free hard disk space in megabytes (MB). If there is the risk that the limit is exceeded, the oldest target files are deleted or overwritten. Note that 1 GB is equal to 1024 MB.

Note



The option *Minimal free disk space* may only be used once on a hard disk. If this option is used by multiple jobs or tasks, the cleanup functions will compete with each other and undesirable side effects may occur, such as deletion of the wrong data.

If you start *ibaDatCoordinator* for the first time, the storage space that is already occupied will be calculated. Whenever a file is written to the target directory, *ibaDatCoordinator* adds the size of the file to the value that has already been calculated. If the remaining free disk space falls below the limit, the oldest files will be deleted first.

Note

Due to the way *ibaDatCoordinator* calculates the hard disk space, iba recommends defining the storage limit 10% under the desired hard disk capacity.

Output files**Overwrite existing files in target directory**

When a file is written to the specified directory, the name of the DAT file is taken and a task-specific file extension is appended. If this name already exists in the target directory, an index is added to the file name.

The index will be incremented by one each following time. If this option is enabled, *ibaDatCoordinator* does not add an index, but overwrites the existing file.

Example

The file name 45124527.dat already exists, and the same DAT file is copied to the destination directory again. If the option is disabled, a new file named 45124527_00.dat will be created. If this option is enabled, the existing file will be overwritten.

Note

Duplicate or multiple files are created when a file-generating task (extract, copy/move/delete, report) is re-executed when the job is repeated with the same source file. A job is repeated if one of its tasks could not be executed and the job therefore ends with an error. When the job is repeated, tasks that were already successfully executed during the previous run are also processed again. In consequence, extraction, copy or report would create files with the same data and name.

When extracting to a database, use the *Remove extraction duplicates* option in the *ibaAnalyzer* extraction dialog if necessary to avoid redundant data.

Use infofield/expression for output file name

As with subdirectories, you can also use info fields and expressions to customize and dynamically generate the names of output files. For more information, see the description of the field *Use infofield/expression for directory name* (see above).

You can find an example of use here ➤ *Example: Using info fields for directory/file names*, page 154

Monitor ibaAnalyzer

These options are only available for tasks involving *ibaAnalyzer*, i.e., report, extract to file, extract to DB, publish to DB, condition, and split tasks.

Memory limit

When enabled, the memory used by *ibaAnalyzer* is monitored. If usage exceeds the set storage capacity, the task is canceled and the currently processed DAT file is marked as failed.

Time limit

When enabled, the processing time for this task is monitored. If the set duration is exceeded, the task is canceled and the currently processed DAT file is marked as failed.

5.4.2 Copy/move/delete task



The copy/move/delete task allows you to copy, move, or delete the current data file or the output file from a previous task.

Jobs: Process product-based files - Copy

ID

Task name:

☐ This task is resource critical

Actions

Execute:

Notify:

File operation

Source:

Operation:

☐ Create ZIP archive

Target

Target directory: ..

Username:

Password:

Subdirectories

Directory naming:

Cleanup strategy

Strategy:

Output files

☐ Overwrite existing files in target directory

☐ Use info field/expression for output file name

The task-specific settings are explained below. For all other settings, see [General task settings, page 83](#).

File operation

Source

Specify here whether the task should be executed on a DAT file or on the output file of a previous task. The output of a previous task can be, for example, a report or an extracted file.

Operation

Choose from the options “Copy,” “Move,” and “Delete.”

Create ZIP archive

Select this option if you want to convert the file to a ZIP archive.

5.4.3 Transfer task



The transfer task allows you to transfer files to a remote *ibaDatCoordinator* instance. The task can transfer either the DAT file currently being processed or the result of a previous task (e.g., a PDF file).

Note



To use the task, ensure that the *ibaDatCoordinator* transfer server is activated on the receiving system. See [Transfer server](#), page 42.

Jobs: Every task one time - Transfer

ID
 Task name:
☐ This task is resource critical

Actions
 Execute: Notify:

Source
 Source:

Target
 Server:
 Port: ☐ Limit bandwidth
 Certificate:

Options
 Remote path:
☐ Delete after transfer ☐ Create ZIP archive

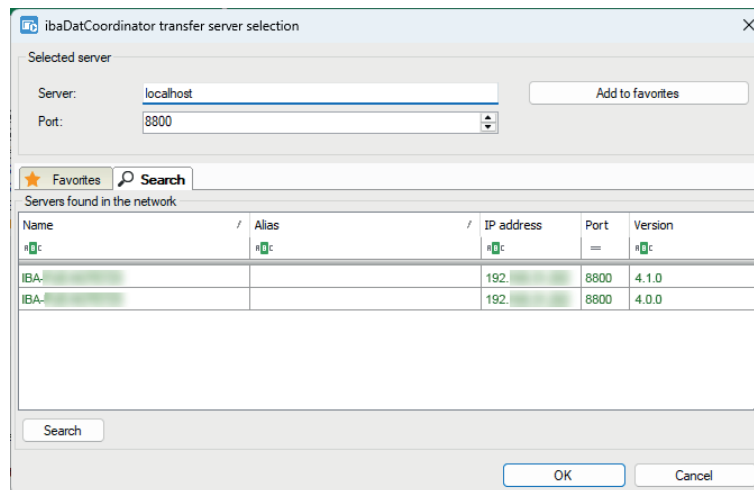
The task-specific settings are explained below. For all other settings, see [General task settings](#), page 83.

Target

Server

You can enter the server name manually in the *Server* field. Alternatively, open the *Transfer server selection* dialog by clicking the <Select server> button.

In the selection dialog, you can see the list of all transfer servers available in the network. The font color of the server indicates whether the server is running (green) or not (red).



Once you have selected the desired server, you can apply the settings in the client by double-clicking on the corresponding entry or clicking on <OK>. The server list is updated with the <Search> button.

Limit bandwidth

This option allows you to activate the maximum transfer speed. To do this, enter the desired bandwidth in the numeric field and choose between the units "mbps" or "kbps".

Certificate

Here you can select a certificate if encryption of the connection has been configured on the other side.

If you select the "Manage certificates" option, you will be taken directly to the *Settings* tab. There, you can also import or export existing certificates in the certificate management section. See [➤ Settings – Certificates](#), page 47.

Note



The same certificate must be used on both the transfer client (for the transfer task) and the transfer server. When exporting the certificate, the private key does not have to be exported. If you do not want encryption, select the "No certificate" option on both sides.

Options

Remote path

Here you can select a file path where the data will be stored on the transfer server.

Delete after transfer

Enable this option if you want the file to be deleted from the client after processing.

Create ZIP archive

When enabled, files are converted to a ZIP archive before being transferred.

5.4.4 Upload task



Similar to the copy/move/delete task, the upload task allows you to upload files to a remote server. The task can transfer either the DAT file currently being processed or the result of a previous task (e.g., a PDF file).

The task-specific settings are explained below. For all other settings, see [↗ General task settings, page 83](#).

Connection

Protocol

Here you can select the transmission protocol. You can test the respective connection settings using the <?> button. The connection settings differ depending on the protocol.

Connection settings FTP

Jobs: Process files with roll force transmission - Upload

ID

Task name:

☐ This task is resource critical

Actions

Execute:

Notify:

Source

Source:

Connection

Protocol:

Server:

Port:

☐ Anonymous

Username:

Password:

FTP mode:

Encryption:

Certificate file:

☐ Accept any TLS certificate

Option

Remote path:

☐ Create ZIP archive

Subdirectories

Directory naming:

Directory structure:

Time used to create directory:

☐ Use multiple levels of subdirectories

☐ Use 4 numbers for the year

i Time-based generation of subdirectories.

Output files

☐ Overwrite existing files in target directory

☐ Use info field/expression for output file name

For the connection via FTP, the server and the port (default setting: 21) are required. For authentication, you can enter your user name and password or alternatively use anonymous login. Then both the user name and password are "anonymous".

In the *Encryption* field, you can specify whether the simple unsecured variant (no encryption) or one of the encryption methods should be used to call up FTPS. When you select an encryption method, you can specify a certificate to be used to establish the connection. Alternatively, any certificate can be accepted by the server.

Under *FTP Mode*, you can switch between active and passive mode.

Issue 4.1

94



Connection settings SFTP/SCP

Connection settings SFTP/SCP dialog box. Fields include: Protocol (SCP), Server (s3.amazonaws.com), Port (22), Username (account), Password (masked), Private key (empty), and a checkbox for 'Accept any SSH host key'.

For the connection via SFTP or SCP, the server and the port (default: 22) are required. Authentication can be done with a user name and password.

Connection settings Amazon S3 Bucket

Connection settings Amazon S3 Bucket dialog box. Fields include: Protocol (Amazon S3), Region (Europe (Frankfurt)), URL (empty), Force path style (unchecked), Access key ID (account), Secret key (masked), and Checksum algorithm (CRC-32).

In addition to the region for the connection to Amazon S3, the *Access key ID* and the *Secret key* of an AIM user with the appropriate authorization are required. Further information can be found in the AWS documentation (Amazon Web Services).

Azure Data Lake/Azure Blob Storage

Connection settings Azure Data Lake/Azure Blob Storage dialog box. Fields include: Protocol (Azure Data Lake), Account name (account), Authorization scheme (Shared key), and Shared key (masked).

To connect to Azure Data Lake, you must specify the *Account name* and *Shared key* or *Blob SAS token* for the data lake. Further information can be found in the Azure documentation.

Remote path and Target

In the *Remote path* field, specify the destination folder on the target system. Subdirectories can be created automatically in relation to this directory (see chapter [General task settings](#), page 83)

Option dialog box. Fields include: Remote path (/), Create ZIP archive (unchecked), Subdirectories section with Directory naming (Time-based), Directory structure (Daily), Time used to create directory ('Date created' from DAT file), Use multiple levels of subdirectories (unchecked), Use 4 numbers for the year (unchecked), and an information icon for 'Time-based generation of subdirectories.'

With the option *Create ZIP archive* it is possible to pack the DAT file into a ZIP archive before uploading.

5.4.5 Report task



This task allows you to create reports automatically. You can print the report using the default printer or save it as a file in one of the formats listed below. The report is generated using an analysis file (PDO file) which was created with *ibaAnalyzer*.

Jobs: Process product-based files - Report

ID
Task name:

☐ This task is resource critical

Actions
Execute:
Notify:

Report analysis file

Target

☒ File
 ☐ Print

Target directory:
..

Username:
Password:

File type:
⌵

☐ Create subdirectories for image files

Subdirectories
Directory naming:

Directory structure:
⌵

Time used to create directory:
⌵

☐ Use multiple levels of subdirectories
☐ Use 4 numbers for the year
☒ Time-based generation of subdirectories.

Cleanup strategy
Strategy:

Output files

☐ Overwrite existing files in target directory
☐ Set "Date Modified" time of output file to match DAT file
☐ Use info field/expression for output file name

Monitor ibaAnalyzer

☒ Memory limit: abort task if ibaAnalyzer starts using more than MB of memory
☒ Time limit: abort task if ibaAnalyzer takes more than minutes to complete

The task-specific settings are explained below. For all other settings, see [General task settings](#), page 83.

Report analysis file

See [Task analysis file](#), page 85.

Target

Select here whether the report should be output to the default printer (*Print* option) or written to a *File*.

The following formats are available for file output:

File type and extension	Remark
Bitmap (*.bmp)	-
CSV (*.csv)	Delimiter-separated text
Enhanced Metafile (*.emf)	-
Web page (*.htm, *.html)	HTML v3.2-compatible no overlapping objects allowed Turned RTF not supported
JPEG (*.jpg, *.jpeg)	-
Single file web page (*.mht, *.mhtml)	HTML v3.2-compatible no overlapping objects allowed Turned RTF not supported
PDF (*.pdf)	-
Rich Text (*.rtf)	Turned RTF and images are not supported
TIFF (*.tif, *.tiff)	-
Text (*.txt)	Table export only
Microsoft Excel (*.xls)	Excel version-independent (native export) RTF text embedded as picture Overlapping objects are not fully supported
XML (*.xml)	-

Output files – Date of the output file

If you activate the option *Set "Date Modified" time of output file to match DAT file*, the (modification) date of the original DAT file is adopted as the date of the output file.

Note

If *ibaDatCoordinator* is installed as a service, the software runs under the system account. If you want to print something, the default printer should be available under the local system account.

If printing is to be done on a network printer, the *ibaDatCoordinator* service may need to be run under a suitable domain user who has access permissions to the network printer.

Other documentation



Information on creating PDO files and defining reports can be found in the *ibaAnalyzer* documentation.

5.4.6 Condition task



Use the condition task if you want to control the execution of a subsequent task using an expression. The result of the task can be TRUE or FALSE.

You can create the condition directly from the signals in the DAT file, or in an analysis file (PDO) with logical signal definitions and sub-expressions.

It may happen that the result of the expression is neither TRUE nor FALSE. In this case, the following rules apply:

- If the expression is not precisely defined or contains errors, the result is evaluated as FALSE.
- For analog signals, the result is evaluated as TRUE if it is greater than 0.5. Otherwise, the expression returns FALSE.
- If an expression is not constant, an average value is determined.

Note



The condition only affects the task that directly follows it in the tree structure of the job view, immediately below the condition.

Furthermore, the task that is to be executed or not executed depending on the condition must be set to "On success" or "On failure" in the execution options. Once the dependency has been established, this is indicated by an indented task icon in the tree.

Jobs: Process product-based files - Condition

ID

Task name:

☐ This task is resource critical

Actions

Execute:

Notify:

Expression definition

Optional analysis:

Expression:

X-type:

Test

DAT file:

DAT file password (optional):

Monitor ibaAnalyzer

☒ Memory limit: abort task if ibaAnalyzer starts using more than MB of memory

☒ Time limit: abort task if ibaAnalyzer takes more than minutes to complete

The task-specific settings are explained below. For all other settings, see [General task settings, page 83](#).

Expression definition

Optional analysis

Optionally, enter the full path and file name of an analysis file from which a Expression or calculation result is to be used as a condition. *ibaDatCoordinator* then opens the data file and loads the analysis file specified here, in which a suitable expression is defined.

Expression

Enter the name of the (finished) expression.

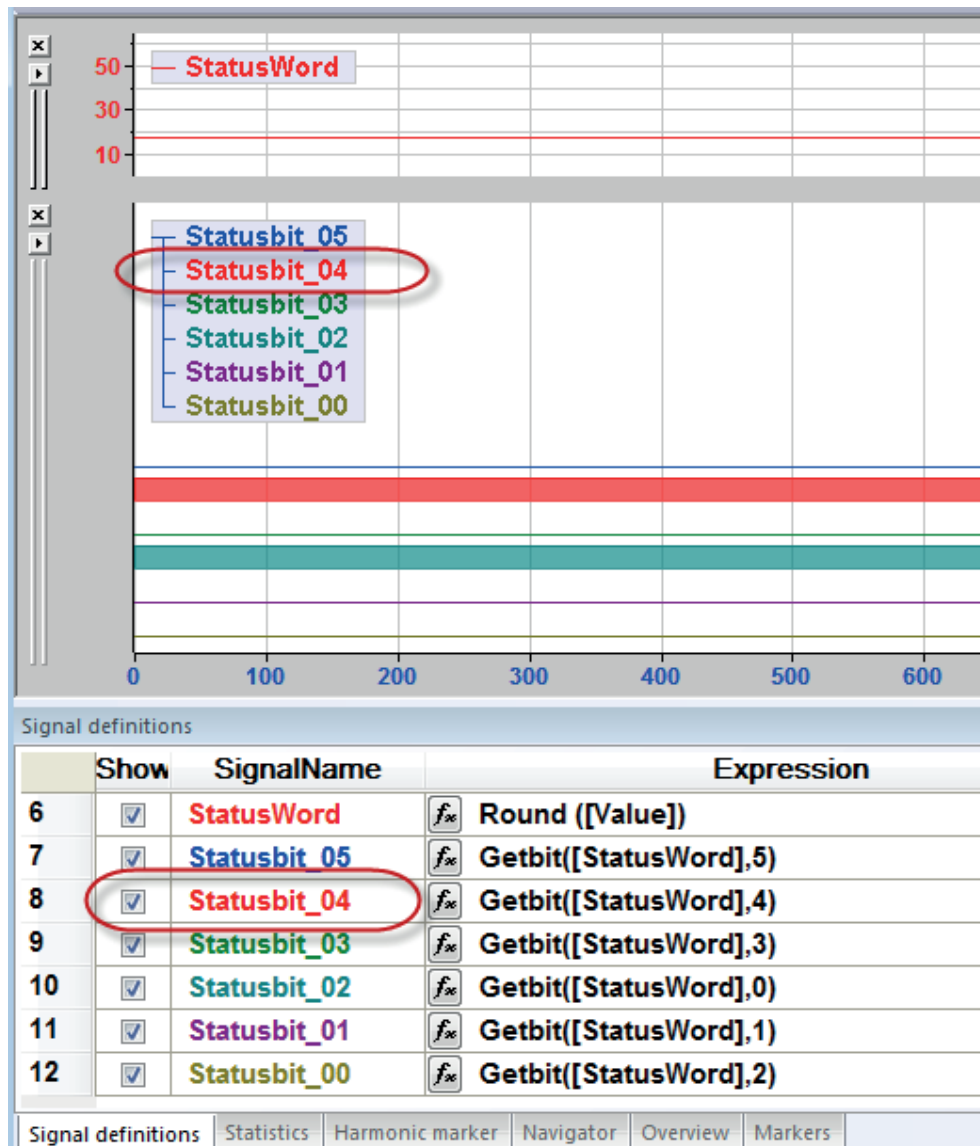
If you want to configure the condition without an analysis file, simply enter a signal contained in the data file in the *Expression* field. You can also enter a formula or an expression, in which the signals from the data file are used. In this case, the command syntax of *ibaAnalyzer* must be taken into account.

X-type

As the expressions are calculated by *ibaAnalyzer* and *ibaAnalyzer* can calculate the expressions in different ways, you need to define the type of the X-axis. The X-axis type specifies how *ibaAnalyzer* calculates an expression. *ibaAnalyzer* can interpret an expression as time, length, frequency, or inverse length.

Condition using an analysis file

With the condition, subsequent tasks are only to be processed if a certain status bit is set in an analysis (e.g. `GetStatusbit.pdo`).



Make the following settings in *ibaDatCoordinator*:

The 'Expression Definition' dialog box is shown. It has three main fields:

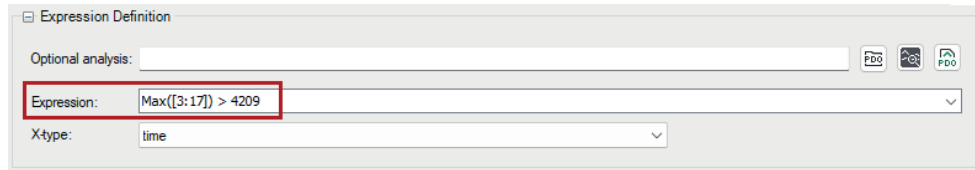
- Optional analysis:** D:\pdo_files\GetStatusbit.pdo
- Expression:** [Statusbit_04]
- Xtype:** time

Since "Statusbit_04" is already a binary signal and can therefore only be TRUE or FALSE, you only need to enter the name when under *Expression*.

Condition without analysis file

The subsequent tasks should only be performed if the maximum value of signal [3:17] in the data file is greater than 4209.

Make the following settings in *ibaDatCoordinator*:



The screenshot shows the 'Expression Definition' dialog box. It has a title bar with a minus, maximize, and close button. Below the title bar is a section labeled 'Optional analysis:' with a text input field and three icons: 'FDO', 'FDO', and 'FDO'. Below this is a section labeled 'Expression:' with a text input field containing 'Max([3:17]) > 4209'. Below the 'Expression' field is a section labeled 'X-type:' with a dropdown menu showing 'time'.

After the expression has been created, you can test it with a data file that you enter or select in the *Test* section, entry field *DAT file*. To do this, click on <?>.

5.4.7 Cleanup task



With a cleanup task, you can limit the data amount of any directory, by deleting the oldest files. The cleanup task operates independently of the cleanup strategies used by the individual file-generating tasks (Extract, Copy/Move/Delete, Report, Update).

As any file-type can be deleted, as long as it can be identified clearly by a file-extension, the cleanup task is suitable for cleaning up any files. For example, you can automatically remove temporary files that are created during script tasks on a regular basis to free up storage space.

Like other tasks, the cleanup task can always be executed depending on a preceding task or a condition within the course of jobs. The cleanup task is particularly suitable for scheduled jobs, e.g., to clean up directories on a weekly basis.

To configure a cleanup task, you must specify a cleanup directory that will be monitored. To access a network drive, you can also enter the login data of a write-authorized user, if required.

Note



To avoid overlaps in cleanup actions, disable task-specific cleanup strategies if you want to use global cleanup, see [➤ Settings – Global cleanup](#), page 55.

Tip



If you use cleanup strategies, look at all cleanup strategies that refer to the storage medium. The cleanup strategy that has been configured in *ibaPDA* might be among these strategies.

Reduce the risk that a process "falls behind" and cannot store its data.

Calculate the required storage capacity for the output files of the different processes and tasks for being able to judge which data sets are especially fast-growing.

Jobs: Process product-based files - Cleanup

ID

Task name:

☐ This task is resource critical

Actions

Execute:

Notify:

Target

Cleanup directory: ..

Username:

Password:
Check path

File extension:

Cleanup strategy

Strategy:

Directory limit:

The task-specific settings are explained below. For all other settings, see [General task settings, page 83](#).

Clean up

Cleanup directory

To configure a cleanup task, you must specify a directory to be monitored. To access a network drive, you can also enter the login details of a user with write permissions, if necessary.

File extension

By specifying the file extension, you can limit the cleanup task to certain file types. You must enter at least one extension. The default setting is *.dat.

You can replace the entry with another one or add more, e.g. *.dat, *.pdf, *.jpg, etc. If you specify multiple extensions, separate them with a comma.

The criterion for deleting files is solely the date of the file.

Cleanup strategy

Strategy

Select which of the following strategies should be used to monitor and clean up the specified directory:

- Limit subdirectories

When the number of allowed subdirectories is reached, the oldest subdirectories containing files with one of the specified extensions are deleted. If all files in a directory are deleted, the subdirectory is also deleted. If there are still files with extensions other than those specified in the subdirectory, they will not be deleted and the subdirectory will remain intact.

- Limit disk space usage

If the total size of all files in the directory exceeds the storage limit set here, the oldest files will be deleted until the limit is no longer exceeded. The date of the files is the only criterion. Only files with the specified extensions are taken into account.

- Minimal free disk space

If the total (!) free space on the hard drive falls below the set limit, the oldest files will be deleted.

This strategy should be used carefully, as the hard disk can be filled by many processes. However, the cleanup task can only delete files in the specified directory. In extreme cases, even deleting files in the directory cannot prevent the free space from falling below the limit.

5.4.8 Pause task



The pause task allows you to pause a job for a specified period of time.

Jobs: Process product-based files - wait_30_sec

ID

Task name:

☐ This task is resource critical

Actions

Execute:

Notify:

Wait time

Duration: seconds

Pause until the specified amount of time has elapsed since

☐ start of the pause task
 ☒ last time the DAT file was modified

The task-specific settings are explained below. For all other settings, see [General task settings, page 83](#).

Wait time

There are two different ways to define how the pause task is performed:

... (since the) start of the pause task

The task is always executed and suspends the execution of the job for the entire specified time period in the *Duration* field.

... (since the) last time the DAT file was modified

ibaDatCoordinator checks the modification date of the DAT file.

If the change occurred longer ago than the time period defined in the *Duration* field, the job continues without interruption.

If the change falls within the defined *Duration*, the job is paused for a period corresponding to the remaining duration between the defined time span and the difference between the current time and the time of the change.

5.4.9 Split task



The split task allows you to split a DAT file into several smaller DAT files. The files contain the same signals as the original DAT file, but with smaller time ranges.

If the original file is password-protected, the generated partial files will be protected with the same password.

Jobs: Process files with roll force transmission - Split

ID

Task name:

☐ This task is resource critical

Actions

Execute:

Notify:

Expression definition

Optional analysis:

Expression:

Split from:

Test

DAT file:

DAT file password (optional):

Target

Target directory:

Username:

Password:

Subdirectories

Directory naming:

Directory structure:

Time used to create directory:

☐ Use multiple levels of subdirectories

☐ Use 4 numbers for the year

Time-based generation of subdirectories.

Cleanup strategy

Strategy:

Output files

☐ Overwrite existing files in target directory

☐ Set "Date Modified" time of output file to match DAT file

☐ Use info field/expression for output file name

Monitor ibaAnalyzer

☒ Memory limit: abort task if ibaAnalyzer starts using more than MB of memory

☒ Time limit: abort task if ibaAnalyzer takes more than minutes to complete

The task-specific settings are explained below. For all other settings, see ➤ *Configuring a task*, page 82.

Expression definition

Here you can define how the output file should be split into DAT files with smaller time ranges using an expression.

Optional analysis

Specifying an analysis file is optional. If the expression for splitting the DAT file is sufficient, it will not be necessary to specify an analysis. If you want to use an analysis, enter the path name and file name of the .pdo file in this field or select the file using the browse button. Click on the *ibaAnalyzer* button next to the field to check whether *ibaAnalyzer* can open the PDO file.

Expression

Enter an expression here that can be used to split the DAT file. A digital signal is normally used here. You can also enter a formula or an expression, in which the signals from the data file are used. In this case, the command syntax of *ibaAnalyzer* must be taken into account.

Split from

Here you can decide how the signal specified in the "Expression" field is used to split the DAT files. There are two options to choose from:

- Rising to rising edge:

The first part of the DAT file is split from the beginning of the time range until the first time the expressions is changed from FALSE to TRUE. Furthermore, the file is split from each transition from FALSE/TRUE to the next transition from FALSE/TRUE. The last part starts from the last FALSE/TRUE transition up until the end of the time range.

- Rising to falling edge:

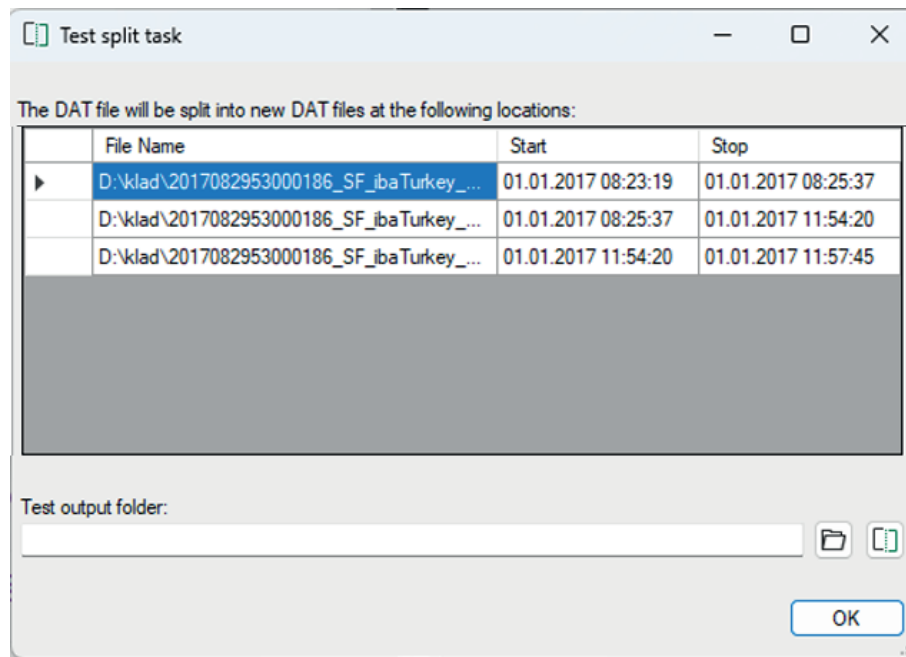
Only the parts of the time range of the DAT file in which the expression is recognized as TRUE are saved to new files. A separate DAT file is generated for each range in which the expression is TRUE.

Note that when opening and attaching the generated files in *ibaAnalyzer* and selecting the *Rising to rising edge* option, the entire original file is represented without losing any data. When selecting the *Rising to falling edge* option, there will be gaps in the signals if the expression was FALSE. This data will not be represented.

Test

In the *Test* section you can test the split task using a DAT file. Enter the path name and file name of the test file in the field or select the file using the browse button. Once you have entered a valid path, open the test dialog by clicking the <?> button.

However, testing the split task is not supported when the service is running on a remote system.



The potential part files are listed in the table. Depending on the complexity of the expression, the optional analysis and the volume of data, this can take a while. You can cancel the process using the <Stop> button.

In the *Test output folder* field, enter a path to a directory in which the resulting .dat files will be generated or select a directory using the browse button. To start splitting the DAT file according to the analysis specification, click the <Split> button. Instead of the <Split> button there will again be a <Stop> button, which you can use to cancel the process. Click <OK> to leave the dialog.

5.4.10 Script task



The script task allows you to run a script that you have created. Scripts that typically run under the Windows command prompt are supported, e.g., batch files, VBS (Visual Basic Script), or JVS (Java Script).

Note



For security reasons, the script task is disabled by default. To use the script task, enable the corresponding option in the *ibaDatCoordinator* settings, see [Settings – Script task](#), page 46.

Jobs: Process product-based files - Script

ID

Task name:

☐ This task is resource critical

Actions

Execute:
Notify:

Script

```
1  Enter script here...
```

Execute

Abort script if it takes more than minutes to complete

Arguments:

%f: "entire file path" e.g.: "C:\dats\file1.dat"

%g: file name with file extension e.g.: file1.dat

%h: file name without extension e.g.: file1

☒ Use DAT file as file in arguments

☐ Use the output of a previous task as file in arguments

Test file:

Test script on ☐ client side ☒ server side

The task-specific settings are explained below. For all other settings, see [General task settings, page 83](#).

Script

You can create a script with *ibaDatCoordinator* or open an existing script using the browser button. If the code is known, *ibaDatCoordinator* highlights the syntax. To use the script in *ibaDatCoordinator*, you must save it in an accessible location.

Execute

You can use iba-specific arguments in the script by entering them in the argument line. The parameters (%f, %g, or %h) in the argument line refer to the last processed .DAT file or the last output file from a previous task; the file used depends on the selection in *ibaDatCoordinator*. The following arguments are available:

- %f: entire file path and file name
- %g: entire file name without path
- %h: prefix of the filename without path and extension

To use multiple arguments, enter them one after the other, separated by a space, in the argument line.

Example: Batch script

If you want to include the arguments %f and %g in your script, enter the following in the argument line: %f %g

You can then pass the arguments in the script:

```
set "Para1=%1"  
set "Para2=%2"
```

You can test the script by selecting a DAT file using the browser button or entering the path. To start the test, click on .

The test run of the script can be performed either on the server or client side. The execution on the client side may be helpful for debugging, as an interactive effect with the desktop is possible.

Note

Please understand that iba AG's support team can only provide limited assistance with problems relating to user-generated scripts due to the individual program code.

5.4.11 HD offline event task



With an HD offline event task, you can define one or more events based on existing HD recordings or data files. Each time the task is executed, the event is entered into an existing HD event store in real time.

By configuration of multiple tasks you can generate multiple events.

The task can be configured with every kind of job.

How often the occurrence of an event is recorded depends on the following factors:

- Whether and how often the HD offline event task is executed.
- Whether the event is triggered once or depending on a trigger signal.

A special feature of offline events is that additional information can be stored in the HD store with the event:

- Calculated values (e.g., KPIs)
- Name of the processed data file

Calculations of (statistical) key values for a period, a process section, or a product can thus be performed solely on the *ibaHD-Server* without database extraction, and the results can be stored there.

Note



To be able to use the offline events feature to its full extent, you need at least the following versions of the interacting software products:

- *ibaHD-Server*: v2.4.0 or higher
 - *ibaPDA*: v7.2.0 or higher
 - *ibaPDA*: v2.3.0 or higher
 - *ibaAnalyzer*: v7.1.7 or higher
-

5.4.11.1 HD offline event task – settings

The task-specific settings are explained below. For all other settings, see [General task settings, page 83](#).

Data source

In the data source settings you determine the sources which supply measured values and text signals to be used in the message text of the event.

Analysis file

Enter or browse for the analysis file which includes expressions for calculated, aggregated values or text signals in the logical signal definitions and/or signal table. This analysis file is then applied to the data file (for DAT file jobs) or HD query (for scheduled HD jobs/event jobs).

The calculated figures will be available for use in numeric and text fields of the message under the node "expressions" in the signal tree, additional to the signal tree of an optionally selected data file (see below).

Note



The calculated figures should only return one (1) value per processed data file or HD query. For figures which return multiple values, e. g. a measured signal, the values of all samples will be averaged and the average will be written in the numeric field.

Expressions from the "Computed columns" and "Info columns" of the extractor or report generator dialogs cannot be used.

DAT file (optional)

If you want to use signal values out of the data file in the message text, additionally or instead of just expressions from the analysis file (*.pdo), then you should select an appropriate data file here.

The data file serves as a configuration help for your convenience when looking for the needed signals and using them as placeholders. When the job is running in operation the actual values will be retrieved from the currently loaded data file.

For DAT file jobs you should select one of the original data files as a template.

In case of scheduled HD jobs which rather use HD queries instead of data files for the tasks, you should create a data file from an HD query and select it as optional data file.

If the data file in question is protected by a password, enter the password in the corresponding field for *ibaDatCoordinator* to be able to access the file.

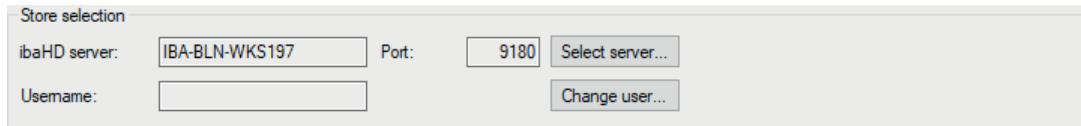
Event configuration

The configuration of offline events follows the same principle like the event configuration in the HD data store configuration of *ibaPDA*.

After the connection to an *ibaHD-Server* has been established you can define the events in the *Configuration* tab and *Client options* tab.

See ↗ *HD offline event task – Configuration tab, page 114* and ↗ *HD offline event task – Client options tab, page 120*.

Store selection



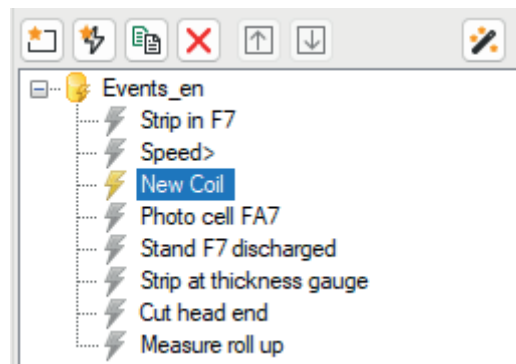
The 'Store selection' dialog box contains the following fields and buttons:

- Store selection** (title)
- ibaHD server:** Text field containing 'IBA-BLN-WKS197'
- Port:** Text field containing '9180'
- Select server...** button
- Username:** Text field (empty)
- Change user...** button

First, select the *ibaHD-Server* whose events you want to use and where you want to store the offline events. Click on <Select server...> and choose the server you want to use from the listing of available *ibaHD-Servers*.

If user management is activated on the relevant server, select a user name with which *ibaDatCoordinator* can access and write to the *ibaHD-Server* by clicking on <Change user...>.

When the connection to the *ibaHD-Server* has been established, the event stores and all defined events on this server will be displayed in the signal tree window below.



The event tree shows all available events that are defined on the connected *ibaHD-Server*. The event "Coil finished" is the defined event in the current task.

It is possible to write data for any event using an HD offline event task. The event symbols ("flash") are designed in different colors for visual differentiation:

- Yellow: This is an event written by the currently open task.
- Grey: The event is available on the *ibaHD-Server*, but is not written by the current task. This can be an event that is written by other tasks of the same *ibaDatCoordinator* configuration on the same computer or an event that is written by another system, e.g. in the *ibaPDA* data recording.

Consequently, offline events, defined by *ibaDatCoordinator* will be displayed in the *ibaPDA* HD data store configuration with a grey icon.

Adding, managing and removing events

In the upper pane (event tree) you can add an event by using the buttons. If required you can create folders for a better organization of events. You can also change the order of events in the tree and copy or delete events here.

	Adds a new folder to the tree
	Adds a new event to the tree
	Copies the selected nodes in the tree
	Deletes the selected nodes in the tree
	Moves the selected node inside the hierarchy level up to the top
	Moves the selected node inside the hierarchy level down to the bottom
	Opens the event wizard

Table 3: Buttons for event management

If you have selected a data file in the data source section, its signal tree will be shown in the lower pane. Then, you can drag & drop signals or expressions from the data file into the event tree pane above, if you want to generate one or more offline events out of it. If you do so, the HD event wizard opens automatically.

Other documentation

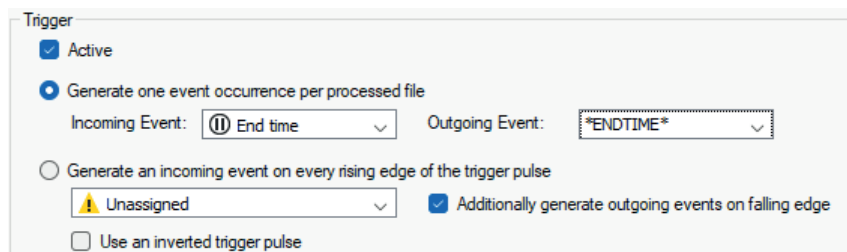


A detailed description of the event wizard can be found in the documentation for *ibaHD-Server*.

5.4.11.2 HD offline event task – Configuration tab

In the *Configuration* tab, you can configure the settings for the events for the HD offline event task.

Trigger



The screenshot shows the 'Trigger' configuration window. It has a title bar 'Trigger'. Inside, there is a checked checkbox 'Active'. Below it, there are two radio buttons. The first is selected: 'Generate one event occurrence per processed file'. This option has two dropdown menus: 'Incoming Event:' with 'End time' selected, and 'Outgoing Event:' with '*ENDTIME*' selected. The second radio button is 'Generate an incoming event on every rising edge of the trigger pulse'. This option has a dropdown menu 'Unassigned' and a checked checkbox 'Additionally generate outgoing events on falling edge'. At the bottom, there is an unchecked checkbox 'Use an inverted trigger pulse'.

Active

When enabled, the relevant event is written with the current task. When activated, the event icon turns yellow. The event is also automatically marked as active when you make changes to the event configuration. New events defined in the current task are automatically active.

Note

You can also modify the configuration of an existing event that was not originally written by the current task. The changes are saved on the *ibaHD-Server* and applied to all clients that use the event. If an *ibaPDA* client is currently writing the relevant event, the changes will only take effect after restarting the recording in the *ibaPDA* client. If you change event names, number fields, or text fields, the configuration will not be accepted after a simple restart, and you will be prompted to make the appropriate adjustments in the *ibaPDA* data recording configuration.

Therefore, the urgent recommendation:

Do not change the configuration of events written by other clients. This can result in incorrectly configured events or data loss!

Generate one event occurrence per processed file

Select this option if you only want to enter the event once per task processing, i.e., per data file or HD query.

Event time

This allows you to determine the time stamp of the generated event. The selection list contains two predefined times, "Start time" and "End time," which refer to the start and end times of the data file or the HD query.

You also have the option of selecting an expression or an info field from the PDO file instead, which is used to determine the time. You can even enter an expression manually in the field. In any case, the expression must show the number of seconds since the start of the data file/HD query as a floating point value.

Please note that all signals are calculated at the specified time.

Recommendation: Use constant values when a specific value is to be written.

Generate one event per processed file

Choose this option if you want the event to be written multiple times depending on a trigger signal with reference to the loaded data file or HD query. In this case you also have to select an appropriate trigger pulse signal.

The trigger signal can be part of the measurement file, the HD query, or the analysis file. The signal can be a virtual signal, computed by *ibaPDA* or any other digital signal, e. g. from a PLC.

Non-constant numerical values are averaged over the duration for which the trigger signal is TRUE.

The event will be written with every falling edge of the trigger signal at the corresponding instant of time.

Example Limit violation

Configure a virtual signal in *ibaPDA* by an expression which compares a measured value and a limit value. If the measured value is higher than the limit, the pulse signal is TRUE, else it is FALSE. Make sure that the pulse signal is recorded in the data file.

If the limit has been exceeded several times over the time range of the data file or HD query, multiple rising edges are available in the data recording.

In the event configuration of the HD offline event task, select the virtual trigger signal from the signal tree of the data file. The next time the task is executed, the limit violations will be entered as events in the HD recording.

If you perform an HD query later with *ibaAnalyzer* you will find the event occurrences at the correct instants of time.

General**Name**

Here you enter the name of the event. The event is displayed under this name in the left tree.

Priority

Here, you can select a priority for the event. The three priority groups *High*, *Normal* and *Low* are predefined in the drop-down list.

However, you can also add your own priority classes to the list by simply writing the desired plain text (e.g. "error", "warning", "information" into the field. After accepting the configuration, this priority class is also available in the drop-down list. There is no need to assign a priority to an event.

Comment 1 or comment 2

Just like for an analog or digital signal in the I/O Manager of the *ibaPDA*, you can enter every additional information in the fields *Comment 1* and *Comment 2*.

URI

Here you can specify a URI that opens in the default browser as soon as the *ibaPDA* user clicks on the corresponding field in the event table.

Tip

You can also set or reset the property *Active* for a folder. Hereby, all events which are in the folder are activated or deactivated.

Message

Here you can enter a message which will be put out while triggering the event.

The entered text can contain placeholders for numbers and texts which refer to analog/digital signals and text signals of the I/O manager. When configuring offline events with *ibaDatCoordinator* the placeholders can be used for numeric values or texts from a data file or HD query and from a PDO file. While triggering, these placeholders in the text of the message are filled dynamically with the current value of the signal or technostring.

How to define the placeholders is described under "Numeric fields" and "Text fields".

Numeric fields

In the table, you can define up to eight placeholders for numbers which can be used in the message.

Column Name

Here, you enter an arbitrary name for the numeric field.

Column Channel

Name	Channel	Format
Product number	[5:0] ProductNo	1.00
Speed	[3:13] 109 F7-speed for tens	1.00
119 Strip thckn. beh. F7	3:13: 109 F7-speed for tension reel	
	3:14: 110 add. setpoint F6	
	3:15: 111 Delta-width M3	
	3:16: 112 Temp. behind F7	

Here, you select the signal whose current value has to be entered into the message while triggering the event.

Column Format

Name	Channel	Format
Product number	[5:0] ProductNo	1.00

Clicking on the <...> button opens the window for entering the number format.

Define the format which will be used to display the current value in the message here.

Formatting

Minimum number of digits before decimal point: 1

Number of digits after decimal point: 2

Format positive numbers: add nothing

Example: 1.00
-1.00

OK Cancel

You enter a placeholder for a numeric field in the message text by placing the cursor on the right place in the message and by clicking on the icon  at the desired numeric field. The placeholder is automatically entered at the cursor position (marked by square brackets and red bold printed text).

Message

Product no. **[Product number]** in Stand F7; Speed **[Speed]** Thickness

[119 Strip thckn. beh. F7]

Name	Channel	Format
Product number	[5:0] ProductNo	1.00
Speed	[3:13] 109 F7-spe	1.00
119 Strip thckn. beh. F7	[3:23] 119 Strip t	1.00

Text fields

In the table, you can define up to eight placeholders for texts which can be used in the message.

Column Name

Here, you enter a random name for the text field.

Column Channel

Here, you select the text signal whose current value has to be entered into the message at the triggering of the event.

Name	Channel
Client	Client ID

Unassigned Client ID

You enter a placeholder for a numeric field in the message text by placing the cursor on the right place in the message and by clicking on the icon  at the desired numeric field. The placeholder is automatically entered at the cursor position (marked by square brackets and red bold printed text).

Message

Product no. **[Product number]** in stand F7; Speed **[Speed]**; Thickness **[Thickness]**; Client **[Client]**

Name	Channel	Format
Product number	^ [3:17] 113 Hot coil nu	1.00
Speed	^ [3:13] 109 F7-speed f	1.00
Thickness	^ [3:23] 119 Strip thick	1.00

Name	Channel
Client	Client ID

Note



The signals for channels in the table for number or text fields can be selected via the drop-down menu and also via drag & drop from the signal tree displayed in the left-hand area of the configuration window. Thereby only the correct signal types in the correct tables are accepted (analog or digital signals for number fields, text signals for text fields).

Note



The names of the number and text fields for an event have to be unique!

Note



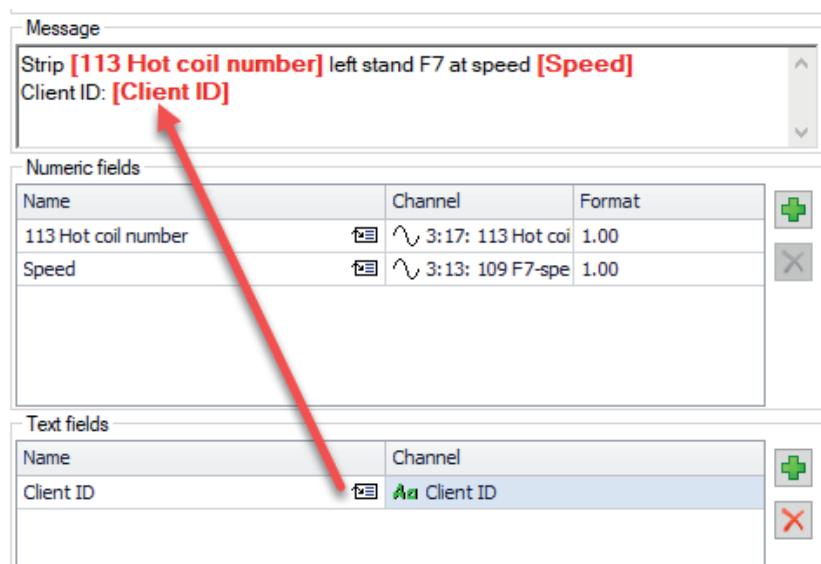
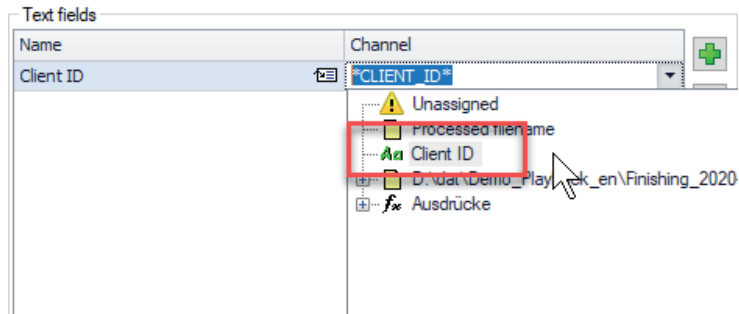
All used expressions need to be timebased! Lengthbased signals can be converted to timebased signals using the function *ConvertBase ([lengthbased_expression],1,0)*.

Additional text elements for HD offline events

When configuring HD offline events, additional elements are available in the "Channel" column.

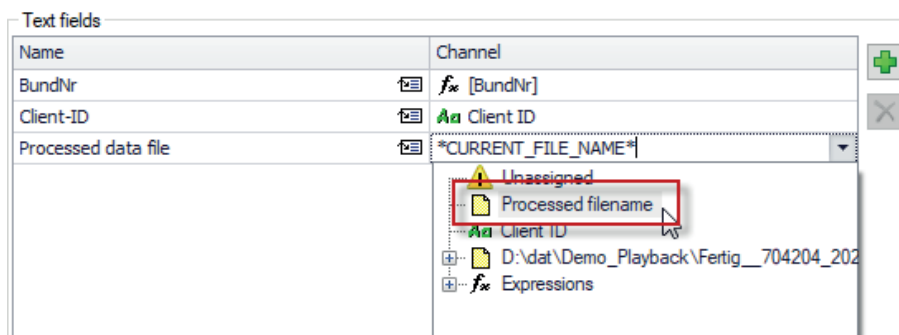
Client ID

HD offline events can be generated from various sources and stored in the same event store. To make it easier to identify the origin of an event, i.e., its creator, in the event display later on, you can insert the client ID in the message text. The client ID is available as a text signal in the signal tree of the "Channel" column.



Name of the processed data file

The name of the processed data file is another piece of additional information that is available by default and can be inserted into the message as a text field. This will give you a reference to the data file that was used for the respective event.



Use of the data file name in the message:

Message

Strip **[113 Hot coil number]** left stand F7 at speed **[Speed]**
 Client ID: **[Client ID]**
 Data file: **[Processed data file]**

Numeric fields

Name	Channel	Format
113 Hot coil number	3:17: 113 Hot coi	1.00
Speed	3:13: 109 F7-spe	1.00

Text fields

Name	Channel
Client ID	Client ID
Processed data file	Processed filename

5.4.11.3 HD offline event task – Client options tab

By configuring the client options, you can determine how events are displayed on the *ibaPDA/ibaHD* client computers in the event display and which commands are sent to the trend displays when an event occurs.

Other documentation



You will find a detailed description of the *Client options* tab settings in the *ibaHD-Server* documentation.

5.4.12 HD import task



With the HD import task you can import DAT files into an HD store of an *ibaHD-Server*. To prevent the network from being overloaded by this task, you can limit the transfer rate of the data when importing. If the DAT files contain info fields, text channels can be created from them.

Jobs: New time period job - HD import

ID

Task name:

☐ This task is resource critical

Actions

Execute: Always

Notify:

Disabled

Destination

⚠ Importing data from different sources into the same HD store should be avoided. Switching signal trees will increase store sizes and eventually lead to performance problems.

ibaHD server:

[Redacted]

Port:

9180

Select server...

Username:

admin

Change user...

ibaHD store selection:

☐ HD_Time1

Data cap

☐ Limit throughput to

1.5

MB/s

Info fields

☐ Import info fields as text channel

Module number:

8190

Module name:

DAT info

Channel No. /	Info Field	Channel Name

+
×

The task-specific settings are explained below. For all other settings, see ➤ *General task settings, page 83*.

Target

Here you can enter the IP address or name of the *ibaHD-Server* and the time-based HD store as the destination for the DAT file directly, or select it from the list. The table lists all *ibaHD-Servers* detected in the network. Click the <Search> button to update the list.

For further information on *ibaHD-Server* connection and user management, see ➤ *Establishing ibaHD-Server connection, page 157*.

Data cap

You have the option of limiting the transfer rate. This may be important if data collection is also active at the same time in the target HD store. Enable the *Limit throughput* option if necessary and enter an upper limit for the transfer rate in MB/s.

Info fields

Enter the info fields which are to be imported as a text channel. Enter a module number and a module name.

5.4.13 Extract to file task



The extract to file task allows you to extract data from a DAT file using an analysis file and store it in another file. You can create the analysis file for this in *ibaAnalyzer*. To use the task type, you need a corresponding license *ibaDatCoordinator-File-Extract*, see [🔗 Licenses](#), page 9.

Other documentation



Further information on data extraction with *ibaAnalyzer* can be found in the *ibaAnalyzer* documentation.

Jobs: Process product-based files - Extract to file

ID

Task name:

☐ This task is resource critical

Actions

Execute:

Notify:

Analysis file for extraction

Target

☐ Extract to database
☒ Extract to file

Target directory:

Username:

Password:

File type:

Subdirectories

Directory naming:

Directory structure:

Time used to create directory:

☐ Use multiple levels of subdirectories

☐ Use 4 numbers for the year

Time-based generation of subdirectories.

Cleanup strategy

Strategy:

Output files

☐ Overwrite existing files in target directory

☐ Set "Date Modified" time of output file to match DAT file

☐ Use info field/expression for output file name

Monitor ibaAnalyzer

☒ Memory limit: abort task if ibaAnalyzer starts using more than MB of memory

☒ Time limit: abort task if ibaAnalyzer takes more than minutes to complete

The task-specific settings are explained below. For all other settings, see [General task settings, page 83](#).

Analysis file for extraction

See [Task analysis file, page 85](#).

Target

Extract to database/Extract to file

The “Extract to file” option is selected by default.

File type

Select here the required type for the output file.

File type	Remark
DAT file (<i>ibaPDA</i> binary format)	iba DAT file (*.dat)
TXT files	ASCII file, tab-separated format
COMTRADE	ASCII-file specifically for measurement purposes in energy technology (*.dat, *.cfg)
TDMS	National Instruments file format (*.tdms, *.tdms_index)
Apache Parquet (*.parquet)	Apache Parquet format (*.parquet)
MATLAB	MATLAB format (*.mat)
CSV file	CSV format (*.csv)

5.4.14 Extract to DB task and publish to DB task



Using the extract to DB task and the publish to DB task, you can use an analysis file to extract data from a DAT file and store it in a database. Unlike the extract to DB task, the publish to DB task allows you to write only calculated columns and info field columns to the database. You can create the analysis file for this in *ibaAnalyzer*.

To use the task types, you need a corresponding license *ibaDatCoordinator-DB*. Alternatively, you can use the *ibaDatCoordinator-Publish* license for the publish task, see [Licenses, page 9](#).

Other documentation



Further information on data extraction with *ibaAnalyzer* can be found in the *ibaAnalyzer* documentation.

The task-specific settings are explained below. For all other settings, see [General task settings, page 83](#).

Jobs: Process product-based files - Extract to DB

ID
 Task name:
☐ This task is resource critical

Actions
 Execute: Notify:

Analysis file for extraction
   

Target
☒ Extract to database ☐ Extract to file
☐ Extract only "Computed Columns" and "Info Columns" (requires ibaDatCoordinator-Publish license)

Monitor ibaAnalyzer
☒ Memory limit: abort task if ibaAnalyzer starts using more than MB of memory
☒ Time limit: abort task if ibaAnalyzer takes more than minutes to complete

Analysis file for extraction

See [Task analysis file](#), page 85.

Target

Extract to database/Extract to file

The "Extract to database" option is selected by default.

To extract data into a database, all you need is a PDO file containing all the relevant settings for the extraction. The additional option *Extract only "Computed Columns" and "Info Columns"* is enabled for the publish to DB task. In this case, *ibaAnalyzer* ignores the profile settings in the PDO file and extracts only the calculated columns and info columns to the database, but no channel or segment data. So only the "file table" is filled.

5.4.15 Kafka publish task



The Kafka publish task allows you to send calculated values to any Kafka broker instance.

Note



The Kafka broker is not included in the installation and needs to be provided by the user. You will also need an *ibaDatCoordinator-Publish* license, see [Licenses](#), page 9.

Each task writes to only one topic. If values are to be sent to multiple topics, you must configure multiple tasks.

The task-specific settings are explained in [Kafka publish task – Connection tab](#), page 126 and [Kafka publish task – Target tab](#), page 131.

For all other settings, see [General task settings](#), page 83.

5.4.15.1 Kafka publish task – Connection tab

Depending on the security settings on the broker, the connection settings can be different. If Type "Kafka" is selected, the following settings are available:

The screenshot shows the 'Connection' tab of a configuration window. It contains the following fields and controls:

- Identifier:** A text input field.
- Type:** A dropdown menu set to 'Kafka'. To its right is a button labeled 'Test connection and populate topic list'.
- Cluster address:** A text input field.
- Cluster connection security:** A dropdown menu set to 'PLAINTEXT'.
- Schema registry address:** A checkbox labeled 'Schema registry address:' followed by a text input field.
- Schema registry connection security:** A dropdown menu set to 'HTTP'.
- Message timeout:** A spinner box set to '3' with a unit 's'.
- Acknowledgment mode:** A dropdown menu set to 'None'.
- Additional parameters:** A table with two columns: 'Key' and 'Value'. To the right of the table are three icons: a plus sign (+), a refresh/circular arrow icon, and a save/disk icon.

Identifier

The *Identifier* is a text-based value that can be included in the datasets written in the cluster. It may be useful for distinguishing between several Kafka publish tasks or *ibaDatCoordinator* instances.

For further settings for the "Kafka" type, see [Kafka type, page 126](#).

For a description of the settings for the "Event Hub" type, see [Type Event Hub, page 131](#).

5.4.15.1.1 Kafka type

The following settings are required in the Kafka publish task for the "Kafka" type.

Cluster address

Enter the host name and port of one of the brokers in the Kafka cluster here. In case you want to address multiple brokers at the same time, separate the single entries by commas (e.g. 'hostname1:9092, hostname2:9092, hostname3:9092').

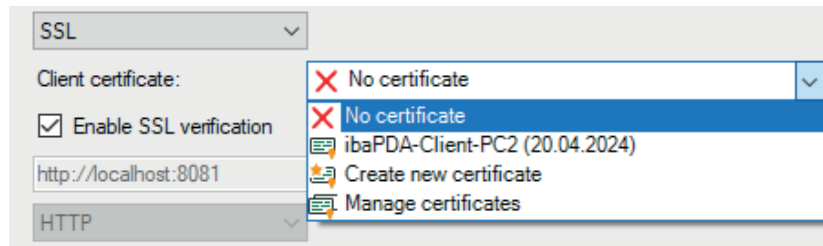
Cluster connection security

Different encryption and authentication methods are supported. Depending on the selected option, further entries are required.

■ PLAINTEXT

Data is exchanged in plain text, no further entries

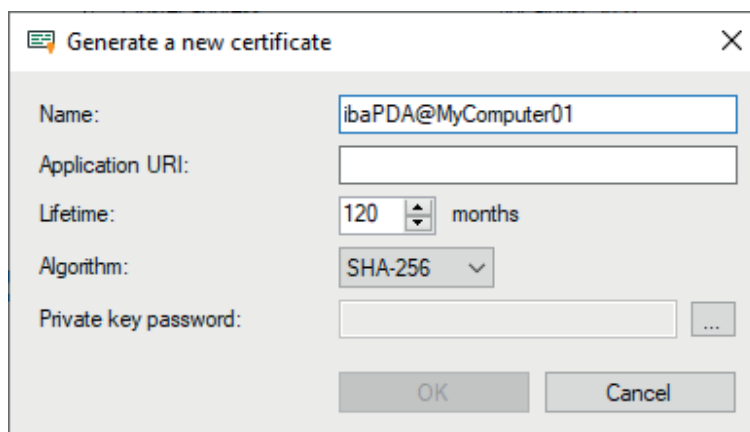
■ SSL



The SSL communication requires certificates. You can store and manage certificates in the central certificate store in *ibaDatCoordinator*. Detailed information about the certificate store can be found in the *ibaPDA* manual.

The certificates that can be used are available for selection in the *Client certificate* drop-down menu. In addition, more options can be selected.

- *No certificate*: No certificate is used. However, this usually causes the validation to fail.
- *Manage certificates*: Opens the central certificate store, where certificates can be managed centrally in *ibaDatCoordinator*.
- *Create new certificate*: *ibaDatCoordinator* creates a new self-signed certificate. Enter the necessary settings in the opening dialog. When the certificate is successfully created, the new certificate is selected.



Enter a name for the certificate. You can change the default name.

Entering an Application URI is optional. Set the lifetime and select the algorithm. Available for selection are SHA-256, SHA-384 and SHA-512.

You have to assign a password in order to create a certificate. To enter the password for the private key, click the <...> button. For security reasons, you must enter the password twice in the following dialog. The password can be assigned arbitrarily, it does not have to meet any other requirements.

Enable SSL verification

If you enable the SSL verification, the certificate of the Kafka server is verified automatically. Among the certificates in *ibaDatCoordinator*, the issuer certificate with which the Kafka server certificate was signed must be available.

Note**Using chained SSL certificates**

In *ibaDatCoordinator*, you can also import chained certificates. When importing (e.g. PEM file or PFX file), the chained certificates are splitted into single certificates. After the import of a chained certificate, there are therefore several individual entries under the certificates.

If the Kafka server certificate was signed using a chain of issuer certificates and you want to use the SSL verification at the same time, the configuration of the keystore in the Kafka server is important: If the keystore also contains the complete chain of issuer certificates in addition to the Kafka server certificate, then only the certificate of the root CA must be available in *ibaDatCoordinator*. If the keystore contains only the Kafka server certificate, then in *ibaDatCoordinator* the complete chain of issuer certificates must be available, so that the verification is possible.

■ SASL/PLAINTEXT

First of all, select the SASL authentication mechanism:

- PLAIN (all data is exchanged in plain text)
- SCRAM-SHA-256
- SCRAM-SHA-512

Enter the username and the password in the corresponding fields for logging in to the Kafka cluster.

■ SASL/SSL

As with *SASL/PLAINTEXT*, select the SASL mechanism and enter the username and password. The certificates that can be used are available for selection in the *Client certificate* drop-down menu.

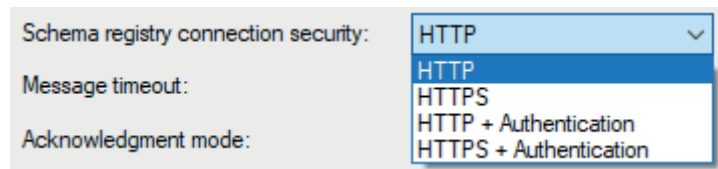
For the activation and use of the SSL verification, the instructions from the section on cluster connection security with *SSL*.

Schema registry address

Enable this option if a schema registry is used in the Kafka cluster. Enter the host name and port of the schema registry. Depending on the connection security used, the address must start with "http://" or "https://". By default, this option is disabled.

Schema registry connection security

ibaDatCoordinator supports different methods for connection security. Depending on the selected option, further entries are required.



The screenshot shows a configuration form with three labels: 'Schema registry connection security:', 'Message timeout:', and 'Acknowledgment mode:'. The 'Schema registry connection security:' dropdown menu is open, showing four options: 'HTTP', 'HTTPS', 'HTTP + Authentication', and 'HTTPS + Authentication'. The 'HTTP' option is currently selected.

■ HTTP

No further entry required

■ HTTPS



The screenshot shows the configuration form with 'Schema registry connection security:' set to 'HTTPS'. Below it, 'Client certificate:' is set to 'No certificate' (indicated by a red X icon). The 'Enable SSL verification' checkbox is checked.

The HTTPS communication requires certificates. The certificates that can be used are available for selection in the *Client certificate* drop-down menu.

The selection of a certificate is done as described above in the description for cluster connection security with *SSL*.

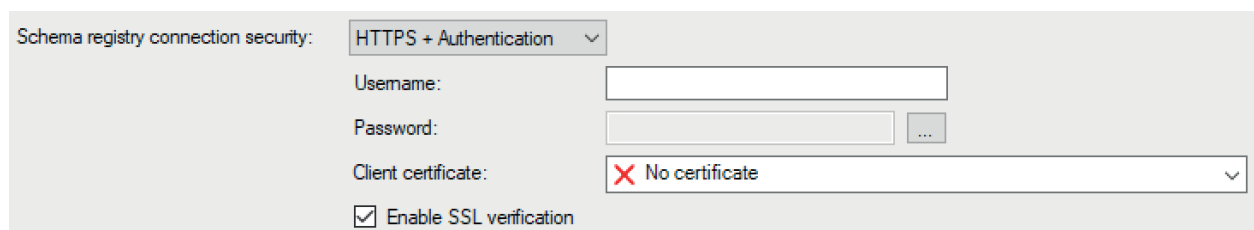
■ HTTP + Authentication



The screenshot shows the configuration form with 'Schema registry connection security:' set to 'HTTP + Authentication'. Below it, there are input fields for 'Username:' and 'Password:'. The 'Password:' field has a small '...' button next to it.

This method requires the input of username and password.

■ HTTPS + Authentication



The screenshot shows the configuration form with 'Schema registry connection security:' set to 'HTTPS + Authentication'. Below it, there are input fields for 'Username:' and 'Password:'. The 'Password:' field has a small '...' button next to it. Below these fields, 'Client certificate:' is set to 'No certificate' (indicated by a red X icon). The 'Enable SSL verification' checkbox is checked.

Like HTTPS, additionally the input of username and password is required.

Message timeout

Time in seconds that *ibaDatCoordinator* waits for a response from the Kafka cluster. The value corresponds to the parameter *message.timeout.ms* in the library *librdkafka*.

Acknowledgment mode

- *None*: Data messages are sent as a continuous stream, but without confirmation of whether the Leader received them or not. This is the fastest mode, but it is not guaranteed that the broker has acquired the data.
- *Leader*: *ibaDatCoordinator* waits until the topic leader confirms the acquisition of the sent data. If the leader malfunctions after sending the confirmation and before the followers have replicated the dataset, the data is lost.
- *All*: *ibaDatCoordinator* does not send new data until the topic leader and the followers have confirmed the respective recording of the data.

The value corresponds to the parameter *acks* in the library *librdkafka*.

Additional parameters

You can configure additional parameters of the *librdkafka* library.

Additional parameters:

Key	Value






Parameters that have already been created are listed in the list with the name and value. Next to the list, there are buttons with the following features:

	Add parameters Enter the name and a value in the fields <i>Parameter</i> and <i>Value</i> .
	Delete selected parameters
	Import parameters You can import parameters as CSV files by selecting the CSV file in the file browser.
	Export parameters You can export to a CSV file. Enter a file name and select a folder.

Note



Configured parameters in the list always have priority and overrule other settings.

The documentation of the parameters available in the *librdkafka* library can be found at:

<https://github.com/edenhill/librdkafka/blob/master/CONFIGURATION.md>

5.4.15.1.2 Type Event Hub

The following settings are required for the **Event Hub** type:

The screenshot shows the 'Connection' tab of a configuration window. It includes fields for 'Identifier', 'Type' (set to 'Event Hub'), 'Connection string', and 'Message timeout' (set to 3 seconds). There is a checkbox for 'Enable SSL verification' and a button labeled 'Test connection and populate topic list'.

Connection string

Enter the connection string from the properties of the Event Hub in the Microsoft Azure Portal here. The string is made up of the end point, key name, key and entity pad.

<Test connection and populate topic list>

You can use the button to test the connection to the Event Hub.

Enable SSL verification

The current CA root certificate "DigiCert Global Root G2" is required for SSL verification, as Azure services use it to sign certificates. This certificate is automatically stored in the certificate store when *ibaDatCoordinator* is installed.

Message timeout

Time in seconds, *ibaDatCoordinator* is waiting for a response from the Event Hub.

5.4.15.2 Kafka publish task – Target tab

In the *Target* tab you will find the configuration for the sent messages and data.

The screenshot shows the 'Target' tab of a configuration window. It includes fields for 'Optional analysis', 'Example DAT file' (C:\dat\ProduktNr_048160.dat), 'DAT file password (optional)', 'Topic' (ibaDatCoordinator_Test), 'Key' (MyKey), 'Signal reference' (\$signalname), 'Metadata' (Name, Timestamp), 'Timestamp' (Start time), 'Data format' (JSON (grouped)), 'UTC offset' (Ignore), and 'Digital values format' (true / false). There is a 'Copy from parent job' button. Below these fields is a tree view showing a file structure with folders like '0. Hydr. Anstellung', '1. Schere / RSF / S1-S6', '2. Gerüste 1-7 u Walzkräfte', '3. IBA-Logic', '4. Schere', and '5. Virtuell'. To the right is a table with columns 'ibaAnalyzer expression (ID)', 'Name', 'Data type', and 'Test value'. The table contains one row with the value '[0:0]' in the first column and '000 F1 Weg ...' in the second column, with 'DOUBLE' in the third column. There are also buttons for adding, deleting, and moving items.

Optional analysis

Here you can specify analysis file (PDO file) to use expressions, defined in it. Enter the path and file name in this field or select the file using the browser button. The following features are available:

- Click on the **ibaAnalyzer** button to check whether **ibaAnalyzer** can open the PDO file.
-  creates a copy of analysis file with name “*_ (ibaDatCoordinator).pdo”, where “*” is an original analysis name, and launches *ibaAnalyzer* with specified file loaded. You can edit this file locally.
-  Uploads the modified analysis file if changes have been made to the local copy.

Example DAT file

You can open a DAT file to use the data it contains for testing expressions by clicking <?>. This file does not affect the task execution and is used only for testing. The evaluated values appear for each signal in the value table in the *Test value* column.

Topic

Here you select the Kafka topic to which the messages are sent. The topic list is populated by pressing <Test connection> button in the *Connection* tab.

Key

Messages are sent to the Kafka broker as key-value-pairs. You can configure the key for the messages using placeholders, which can be used, for example, to identify signals or the *ibaDatCoordinator* instance on the broker.

Key:	My key
Signal reference:	\$signalname
Metadata:	The following placeholders are available:
Data format:	\$identifier: Identifier defined in data storage configuration
Digital values format:	\$signalname: Signal name
	\$unit: Signal unit
	\$comment1: Signal comment 1
	\$comment2: Signal comment 2
ibaAnalyzer Expression	
▶ [0:0]	When using a grouped data format, signal related placeholders will be replaced by empty text

Signal reference

Signal reference is used as signal identifier within the messages. Different placeholders are available and should be chosen to uniquely identify different signals by the values.

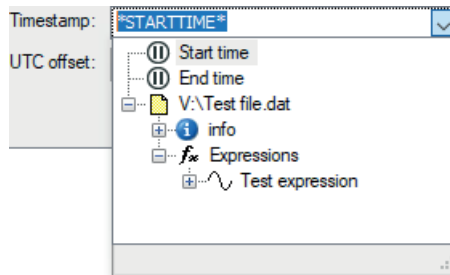
Metadata

You can select which signal metadata you want to attach to the messages sent to the Kafka broker. When using “AVRO (per signal)” metadata are always included.

Timestamp

“Start time” and “End time” are related to processed file and taken from its metadata.

If “Timestamp” is selected as metadata, the timestamp can be computed from the processed DAT file. Besides the usual “Start time” and “End time” options, the timestamp can also be evaluated from an expression. If you select an expression, its averaged value is added to start time in seconds.



The Timestamp format in JSON messages is “YYYY.MM.DDThh:mm:ss:ffffffZ” . The “T” divides date and time, the fractional part of the second has always 6 digits, even if they are all zeroes, and the “Z” marks the time as UTC time.

Data files can have a time UTC offset . Choose from three options how to handle it:

- Ignore: The local time is taken and reported as UTC time.
- Convert timestamp to UTC time: The UTC offset is taken into account and the time stamp is converted to UTC time.
- Append to the timestamp string (JSON formats only): For the JSON formats, the UTC offset can be appended to the timestamp string instead of the letter “Z”. Example: “2022.08.20T12:30:00:000000+02:00”

Data format

JSON (per signal). One message for every computed value is sent. Each message has the following structure.

```
{
  "Signal": "My beauty signal",
  "Value": 1:23,
  [optional metadata fields]
}
```

JSON (grouped). All computed values are sent within one single message with the following structure:

```
{
  "My signal 1": 1:23,
  [optional metadata fields of signal 1]

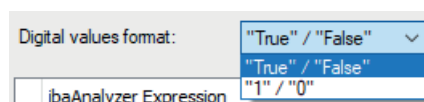
  "My signal 2": 4:56,
  [optional metadata fields of signal 2]
}
```

AVRO. One message for every computed value is sent. Each message is serialized using the following avro schema:

```
{
  "namespace": "de.iba",
  "type": "record",
  "name": "DatCoordinatorRecord",
  "fields": [
    {"name": "Signal", "type": "string"},
    {"name": "ID", "type": ["null", "string"]},
    {"name": "Name", "type": ["null", "string"]},
    {"name": "Unit", "type": ["null", "string"]},
    {"name": "Comment1", "type": ["null", "string"]},
    {"name": "Comment2", "type": ["null", "string"]},
    {"name": "Timestamp", "type": [
      "null",
      {"type": "long", "logicalType": "timestamp-micros"}
    ]},
    {"name": "Identifier", "type": ["null", "string"]},
    {"name": "ValueType", "type": {
      "type": "enum",
      "name": "ValueTypeEnum",
      "symbols": ["BOOLEAN", "DOUBLE", "STRING"]
    }},
    {"name": "BooleanValue", "type": ["null", "boolean"]},
    {"name": "DoubleValue", "type": ["null", "double"]},
    {"name": "StringValue", "type": ["null", "string"]}
  ]
}
```

Digital values format

Since JSON is a text format, digital (Boolean) values are transformed to a text and can be converted to "True"/"False" or "0"/"1" alternatively.



Value table

Here you can specify any valid *ibaAnalyzer* expression (e.g., predefined expressions from the PDO file). You can also select signals or modules from the signal tree and add them to the value table using drag and drop. The signal tree is displayed as soon as you load an analysis file in the *Optional analysis* field or a DAT file in the *Example DAT File* field. The selected expression will be evaluated when the job runs and the result will be published on the server. When a PDO or test file is loaded, you can simply select expressions from the *ibaAnalyzer* file tree.

Note

If the result of the expression is a non-constant signal, the signal will be averaged first before being used. Text signals evaluate to the first text value.

The *Name* is used to identify the signal on the server (it is inserted into the tag names).

In the *Data type* column, select the data type corresponding to the signal type: Analog (by default), Text (string) string or Digital (or Boolean).

You cannot edit the *Test value* column. It is used to show the values of the evaluated expressions against a test DAT file. You can use the buttons on the right side of the table to add, copy, delete, or move expressions up and down.

5.4.16 OPC UA publish task



The OPC UA publish task allows you to publish calculated values via the integrated OPC UA server.

Note

The OPC UA server must be enabled for you to perform this task, see [↗ OPC UA server](#), page 38. You will also need an *ibaDatCoordinator-Publish* license, see [↗ Licenses](#), page 9.

In this task, you can evaluate expressions and publish the results via OPC UA. The number of expressions is not limited. If the result of the expression is a non-constant signal, the signal will be averaged first before being used.

Jobs: Every task one time - OPC UA publish

ID

Task name:

☐ This task is resource critical

Actions

Execute: Notify:

Optional analysis:

Example DAT file:

Optional analysis icons:

DAT icon:

ibaAnalyzer expression (ID)	Variable name	Data type	Test value
[2:0]	Product_Coil_ID	Text	
[2:1]	length_product	Text	
[2:2]	date_string_ym...	Text	
[2:0]	Verbunden	Digital	
[3:0]	length_gauging_...	Analog	
[3:1]	length_gauging_...	Analog	
[3:2]	length_gauging_...	Analog	
[3:3]	length_gauging_...	Analog	
[3:4]	length_gauging_...	Analog	

Monitor ibaAnalyzer

☒ Memory limit: abort task if ibaAnalyzer starts using more than MB of memory

☒ Time limit: abort task if ibaAnalyzer takes more than minutes to complete

The task-specific settings are explained below. For all other settings, see [↗ General task settings, page 83](#).

Optional analysis

Here you can specify an analysis file (PDO file) to use the expressions defined in it. Enter the path and file name in this field or select the file using the browser button. Click on the *ibaAnalyzer* button to check whether *ibaAnalyzer* can open the PDO file.

Example DAT file

You can open a DAT file to use the data it contains for testing expressions by clicking `<?>`. This file does not affect the task execution and is used only for testing. The evaluated values appear for each signal in the value table in the *Test value* column.

Value table

Here you can specify any valid *ibaAnalyzer* expression (e.g., predefined expressions from the PDO file). You can also select signals or modules from the signal tree and add them to the value table using drag and drop. The signal tree is displayed as soon as you load an analysis file in the *Optional analysis* field or a DAT file in the *Example DAT File* field. The selected expression will be evaluated when the job runs and the result will be published on the server. When a PDO or test file is loaded, you can simply select expressions from the *ibaAnalyzer* file tree.

Note



If the result of the expression is a non-constant signal, the signal will be averaged first before being used. Text signals evaluate to the first text value.

The *Variable name* is used to identify the signal on the server (will be included in the tag name).

In the *Data type* column, select the data type corresponding to the signal type: Analog (by default), Text (String) or Digital (or Boolean).

You cannot edit the *Test value* column. It is used to show the values of the evaluated expressions against a test DAT file. Use the buttons on the right side of the table to add, copy, delete, or move expressions up and down in the table.

5.4.17 SNMP publish task

SNMP

The SNMP publish task allows you to publish calculated values via the integrated SNMP server.

Note



The SNMP server must be enabled for you to perform this task, see [↗ SNMP server, page 32](#). You will also need an *ibaDatCoordinator-Publish* license, see [↗ Licenses, page 9](#).

In this task, you can evaluate expressions and publish the results via SNMP. The number of expressions is not limited. If the result of the expression is a non-constant signal, the signal will be averaged first before being used.

Jobs: Every task one time - SNMP publish

ID
Task name:

☒ This task is resource critical

Actions
Execute:
Notify:

Optional analysis:

Example DAT file

C:\dat\ProduktNr_048160.dat

- 0. Hydr. Anstellung
- 1. Schere / RSF / S1-S6
- 2. Gerüste 1-7 u Walzkräfte
- 3. IBA-Logic
- 4. Schere
- 5. Virtuell

ibaAnalyzer expression (ID)	Variable name	Data type	Test value
[0:0]	000 F1 Weg A...	Analog	
[3:12]	108 Zusatzsoll...	Analog	
▶ [5:0]	neueProdukt...	Digital	

Monitor ibaAnalyzer
☒ Memory limit: abort task if ibaAnalyzer starts using more than MB of memory

☒ Time limit: abort task if ibaAnalyzer takes more than minutes to complete

The task-specific settings are explained below. For all other settings, see [General task settings, page 83](#).

Optional analysis

Here you can specify an analysis file (PDO file) to use the expressions defined in it. Enter the path name and file name of the PDO file in this field or select the file using the browser button. Click on the *ibaAnalyzer* button to check whether *ibaAnalyzer* can open the PDO file.

Example DAT file

You can open a DAT file to use the data it contains for testing expressions by clicking <?>. This file does not affect the task execution and is used only for testing. The evaluated values appear for each signal in the value table in the *Test value* column.

Value table

Here you can specify any valid *ibaAnalyzer* expression (e.g., predefined expressions from the PDO file). You can also select signals or modules from the signal tree and add them to the value table using drag and drop. The signal tree is displayed as soon as you load an analysis file in the *Optional analysis* field or a DAT file in the *Example DAT File* field. The selected expression will be evaluated when the job runs and the result will be published on the server. When a PDO or test file is loaded, you can simply select expressions from the *ibaAnalyzer* file tree.

Note



If the result of the expression is a non-constant signal, the signal will be averaged first before being used. Text signals evaluate to the first text value.

The *Variable name* is used to identify the signal on the server (will be included in the tag name).

In the *Data type* column, select the data type corresponding to the signal type: Analog (by default), Text (String) or Digital (or Boolean).

You cannot edit the *Test value* column. It is used to show the values of the evaluated expressions against a test DAT file. Use the buttons on the right side of the table to add, copy, delete, or move expressions up and down in the table.

5.4.18 Time period update task



You can use the time period update task to write or update values from info fields in existing time periods. Creating new fields is not possible with this task. You can add the task to any time period job.

Jobs: New time period job - Time period update

☐ ID

Task name:

☐ This task is resource critical

☐ Actions

Execute: Notify:

☐ Definition

Analysis:   

☐ Monitor ibaAnalyzer

☒ Memory limit: abort task if ibaAnalyzer starts using more than MB of memory

☒ Time limit: abort task if ibaAnalyzer takes more than minutes to complete

You configure the values to be updated in *ibaAnalyzer*. In *ibaDatCoordinator*, you then assign the PDO file containing the configuration of the info fields to be updated in the *Analysis* field.

For all other settings, see [General task settings, page 83](#).

Note



To be able to use this task, you need *ibaAnalyzer* v8.3.0 or higher.

5.4.19 Update task



With an update task, you can add info fields to a DAT-file and rename the DAT file on the basis of the data available in a database. To store the processed data file, you must define a target directory.

To use the update task, you need the *ibaDatCoordinator-Update Data Task* license, see [🔗 Licenses](#), page 9.

Jobs: Process product-based files - Update

ID

Task name:

☐ This task is resource critical

Actions

Execute:

Notify:

Database info

Database provider:

Table name:

Target

Target directory:

Username:

Password:

Subdirectories

Directory naming:

Cleanup strategy

Strategy:

Output files

☐ Overwrite existing files in target directory

☐ Set "Date Modified" time of output file to match DAT file

The task-specific settings are explained below. For all other settings, see [🔗 General task settings](#), page 83.

Database info

In this section you can configure the connection to the database.

Database provider

If database connections have already been configured in the *Settings* navigation menu, you can select a database connection from the list here. See [🔗 Settings – Databases](#), page 53.

If no database connection has been configured yet, select *Create database connection* from the drop-down list. You will be taken to the dialog for configuring a database connection.

Other documentation



A detailed description of the database connection configuration can be found in the *ibaAnalyzer* documentation, Part 4 "Database Interface."

To edit an existing connection, select the connection from the list and then click the <Edit> button.

You can delete a selected connection using the <Remove> button.

Table name

Enter the name of the database table containing the data for the update task here.

Information about the database table

The database table must contain the following columns:

- ID_REF (string data type)
The original name of the DAT file without path and extension
- ID_NEW (string data type)
The new name of the DAT file without path and extension.
- CREATED (time/date)
Date and time can only be used for debugging.
- READY_TO_PROCESS (Boolean / short integer)
This bit notifies *ibaDatCoordinator* if a file can be processed or not. If the cell contains a 0 (FALSE), *ibaDatCoordinator* does not process the file. If the cell contains a 1 (TRUE), *ibaDatCoordinator* processes the file.
- PROCESSED (Boolean / short integer)
The status of this field is changed by *ibaDatCoordinator* when the file is being processed.
Status = 1 (TRUE) on successful processing

All other columns are added to the DAT file as info fields. If an info field of the string data type contains one or more "newline" characters (line break), these are replaced by blanks before being written into the DAT file.

If an info field contains the value NULL, this info field is not generated.

5.4.20 S7 writer task

S7

With the S7 writer task you can extract or calculate data from an iba .data file (DAT file) and write this data to data blocks (DBs) within an S7 PLC. In this task, you can define *ibaAnalyzer* expressions with S7 operands. When executing the task, these expressions are evaluated and sent to the S7. To use the S7 Writer task, you need the ibaDatCoordinator S7 Writer license; see

To use the S7 writer task, you need the *ibaDatCoordinator S7 Writer* license; see [↗ Licenses](#), page 9.

The following requirements must be met in order to use the task:

- S7 PLC for writing the data
- Connection to the S7 PLC via TCP/IP

Jobs: Every task one time - S7 writer

ID
 Task name:
☐ This task is resource critical

Actions
 Execute: Notify:

Connection type: Timeout (s):
 Address: Rack: Slot:
 Optional analysis:
 Example DAT file:

DAT file password (optional):

DAT file
 C:\dat\ProduktNr_048160.dat
 0. Hydr. Anstellung
 1. Schere / RSF / S1-S6
 2. Gerüste 1-7 u Walzkräfte
 3. IBA-Logic
 4. Schere
 5. Virtuell

	ibaAnalyzer expre...	DB	Address	Bit	S7 dat...	Test value
1	[0:0]	1	0	0	REAL	

☐ Ignore expressions that cannot be evaluated

Monitor ibaAnalyzer
☒ Memory limit: abort task if ibaAnalyzer starts using more than MB of memory
☒ Time limit: abort task if ibaAnalyzer takes more than minutes to complete

The task-specific settings are explained below. For all other settings, see [↗ General task settings](#), page 83.

Configuration

S7 connection parameters

Here you can configure the connection to the S7. The *Connection type* can be PG, OP or "other". You can select a connection type that is available on your S7. Enter the IP address of the S7 CPU into the field *Address* and enter the corresponding *Rack* and *Slot* number.

Optional analysis

Optionally, you can specify an *ibaAnalyzer* analysis file (PDO file). When processing a DAT file, this analysis is first loaded before the expressions written to the S7 are evaluated.

This allows you to define more complex expressions in the analysis and refer to these expressions in the expressions for the S7 signals. This enables simpler expressions for the S7 signals which reference expressions in the analysis directly.

Enter the path and file name in the field or select the file using the browser button. Click on the *ibaAnalyzer* button to check whether *ibaAnalyzer* can open the PDO file.

Define S7 signals

In the table you can define the S7 signals. You can add any number of signals by entering new data records in the last row. You can also select signals or modules from the signal tree and add them to the value table using drag and drop. The signal tree is displayed as soon as you load an analysis file in the *Optional analysis* field or a DAT file in the *Example DAT File* field. The table contains the following columns:

- *ibaAnalyzer expression*: Enter a valid *ibaAnalyzer* expression here. This expression is evaluated when executing the task and the result is written to the S7. Ideally, you should enter an expression that evaluates to a constant value. However, if the result of the expression is a fluctuating signal, the signal is averaged before it is processed.
- *DB*: The number of the data block to which the data will be written.
- *Address*: The address within the data block to which the data will be written.
- *Bit*: When it is a digital signal, this is the bit number to which the data will be written.
- *S7 data type*: Type of data to be written. Use BOOL for digital signals.
- *Test value*: This displays the values of the expressions when they are evaluated using a test DAT file.

You can copy and paste text into the table, e.g. from *ibaAnalyzer*. You can delete rows by highlighting a row and then pressing the <Delete> key.

Ignore expressions that cannot be evaluated.

If disabled, the task will not be executed if any of the expressions cannot be evaluated. If you enable this option, signals whose expressions cannot be evaluated will not be written. The other signals with valid expressions are written. If there are no expressions that can be evaluated, the task is canceled.

Test ibaAnalyzer expressions

You can test the *ibaAnalyzer* expressions in the S7 signal table using a test DAT file. Enter the path for the test file into the text field or select a file using the browse button. If the specified DAT file exists and can be accessed, you can click <?>. Otherwise, the button will be grayed out.

After performing the test, the *Test value* column will be populated with the valid expressions. If the expressions are invalid, the column remains empty. A window will appear with one of the following messages:

Error message	Possible cause/remedy
Analysis file not found	The specified analysis file (PDO file) does not exist or is not accessible. The task will fail when it is executed. Remember that you do not have to specify an analysis. You can therefore leave the text field empty, so this error will not occur.
Class not registered	<i>ibaAnalyzer</i> is not installed or a very old version is installed (older than 5.0.0). The task will fail when it is executed.
Your current <i>ibaAnalyzer</i> version is x.y.z. Please update to at least version 6.5.0. We recommend version 6.7.0 or higher. (x.y.z stands for your current <i>ibaAnalyzer</i> version)	This message is displayed if the <i>ibaAnalyzer</i> version is older than 6.7.0. If the version is older than 6.5.0, the execution of the task will fail. If the version is greater than or equal to 6.5.0 but less than 6.7.0, the task will be executed successfully, but with a possible loss of accuracy. (Only floating point numbers with simple precision are used.)
Expression for operand X could not be evaluated (X is an S7 operand)	The <i>ibaAnalyzer</i> expression is invalid or the signals used by the expression do not contain any data or are empty. The task may fail depending on whether the option "Ignore expressions which cannot be evaluated" is enabled. This message is repeated for every operand for which the corresponding <i>ibaAnalyzer</i> expression could not be evaluated.
No valid fields are specified to write to the S7.	None of the operands has an evaluable <i>ibaAnalyzer</i> expression (or operands are not defined at all). The task will fail when it is executed.

Tip



If you have specified both an analysis file and a test file and both files exist and are accessible, the button next to the text box also loads the test file together with the analysis when you start *ibaAnalyzer*.

In addition, the following error messages can appear:

Error message	Possible cause/remedy
Error 0xFFFF50004 (AGL40_TIME-OUT) when calling function AGLPLC-Connect	No connection can be established to the S7. Check the specified IP address. You can also check if the S7 is reachable by pinging it.

Error message	Possible cause/remedy
Error 0xFFFF5001C (AGL40_PLC_NOT_FOUND) when calling function AGLPLCConnect	A connection can be established with the IP address, but this does not belong to an S7 CPU. Check the specified IP address. Also check that the rack and slot numbers are correct.
Writing operand X returned error Error 0xFFFF5000A (AGL40_NO_DATA_ERROR) when calling function AGL_WriteMixEx	The operand is not available in the S7. Check that both the DB number and the address are correct.

5.4.21 Merge task



The merge task allows you to merge multiple DAT files into a single file. A common expression is used for this purpose, which must return the same value for all files involved. This allows multiple DAT files to be opened simultaneously and combined into a new file.

A license for *ibaDatCoordinator-File-Extract* or *ibaAnalyzer-File-Extract* is required to use this task, see [🔗 Licenses](#), page 9.

The merge task is triggered by a new DAT file in a configured directory folder. The configured expression is applied to this file, and *ibaDatCoordinator* searches other configured directories for matching accompanying files where the expression yields the same value. These files are then automatically merged.

Jobs: Every task one time - Merge

ID
 Task name:
☐ This task is resource critical

Actions
 Execute: Notify:

Merge settings
 Analysis file:
 Expression to merge:

Text expression settings
 Start: Length: ☐ Remove blanks at the end ☐ Remove all blanks

Directories

Directory	Username	Directory pas...	File password
▶ The main directory: C:\dat\merge_1	...		
C:\dat\merge_2	...		
*	...		

☐ Scan subdirectories Pause before merge: seconds
☐ Fail task if not all companion files are found

Test
 Example DAT file:
 DAT file password (optional):

The task-specific settings are explained below. For all other settings, see 83.

Tip

Here is an example of how to use the merge task: ➤ *Example: Using the merge task for multistation, page 150*

Merge settings**Analysis file**

Enter the analysis file (PDO) that contains the configuration for the extraction here. Enter the path name and file name of the PDO file or select the file using the browser button. For creating this analysis file, note that all files are opened in parallel in the order configured in the table “Directories”.

Expression to merge

This expression is evaluated for all files to find the right companion files. You can also define the expression in the analysis file.

Text expression settings

If the expression is a text channel there are options to evaluate only parts of the string. Enter at least the *Start* and *Length* of the string.

Directories

Select the directories from which the files to be merged are taken from. The main directory in the first line corresponds to the directory from the job configuration and is read-only.

The order in the table corresponds to the order in which the files are opened by *ibaAnalyzer* during the processing. You can use the controls on the right to add or delete directories or move lines. You can also move the lines using drag & drop. You can also create new lines by editing the empty line at the very bottom.

If the directories or files are password-protected, also enter the user name and password.

Scan subdirectories

If activated, the system also searches for suitable companion files in the subdirectories of the folders.

Pause before merge: ... seconds

When the job is triggered, the specified time should be waited before searching the suitable companion files.

Fail task if not all companion files are found

If activated, the task fails if one or more files are missing. If you disable this option, the merge will be performed without the missing files and a warning will be issued.

Control elements

	Checks whether all configured directories are accessible for <i>ibaDatCoordinator</i> .
	Removes the markers from the DAT files that they have already been processed. After the marking has been deleted, <i>ibaDatCoordinator</i> treats these files as newly created files.

Test

You can check the expression in advance using a test file. To do this, select a DAT file and click the <?> button to execute the expression and display the result. If the DAT file is password-protected, you will need to enter the corresponding password.

5.4.22 DB sync task



The DB Sync task is used to synchronize the database and file system and is only available if the "DBSync" plugin is installed. You will find the installation file on the "iba Software & Manuals" data medium in the following directory:

```
...\01_iba_Software\ibaDatCoordinator\02_Plugins\DBSyncPlugin
```

The task type is designed for use in a scheduled job.

When extracting values from a data file into a database using *ibaAnalyzer*, an entry is created for each data file in the index table (ibaFile table). In addition to the user-defined values for info fields and evaluated columns, this record contains a number of other bits of information for each data file, such as the file name with full file path.

The file name is used by *ibaAnalyzer* or *ibaDaVIS* as a reference for the drill-down into the high-resolution raw data.

A file reference may become invalid if the original file is moved to another folder or completely removed from the system, e.g., by a cleanup task.

Each time it is executed or synchronized, the plugin checks the validity of the file references in the database index table by verifying the existence of the corresponding data file. The result of this check of the validity of the reference can be registered in an additional column of the index table. The status "1" for a record of this column will indicate that the file reference is valid and "0" indicates that the file could not be found at the referenced file path.

An option in the plugin allows you to delete entries for which no data file could be found.

For information on the general settings of the task, see [General task settings, page 83](#).

For a description of how to configure the task, see [Configuring a DB sync task, page 147](#).

5.4.22.1 Configuring a DB sync task

Proceed as follows to configure the DB sync task in *ibaDatCoordinator*.

Prerequisite: The plug-in is installed.

You will find the installation file on the "iba Software & Manuals" data medium in the following directory:

...\01_iba_Software\ibaDatCoordinator\02_Plugins\DBSyncPlugin

1. To run the task regularly, e.g., every 8 hours or once a day, create a scheduled job.

See [🔗 Configuring a job, page 59 and 65](#).

2. Add a DB sync task to the job.
3. Configure the database connection in the *Database information* section.

The screenshot shows a 'Database info' dialog box with the following fields and options:

- Database provider:** A dropdown menu set to 'Sql-server'.
- Computer:** Two radio buttons: 'Local machine' (unselected) and 'Database server' (selected).
- Database name:** A text box containing 'ibaDatMgr'.
- Authentication:** Two radio buttons: 'Use Windows NT authentication' (selected) and 'Specify authentication info' (unselected).
- Username:** An empty text box.
- Password:** An empty text box.
- Table name:** A text box containing 'deFile'.
- Database server:** A text box containing 'IBA-BLN-NOTE345\SQLEXPRESS'.
- Test database connection:** A button located to the right of the authentication options.

Note



In order to use the synchronization results in *ibaDaVIS*, the database connection must refer to the same database and index table that is configured in *ibaDaVIS*.

If you have configured different views in *ibaDaVIS* that refer to different index tables, you must create a DB sync task for each index table.

- **Database provider:** Choice between SQL Server, Oracle, DB-2, or ODBC (Access, MySQL)
- **Computer:** Local machine or a database server
- **Database name:** Database name or DSN (ODBC connection)
- **Authentication:** By operating system or username and password
- **Table name:** Name of the index table (ibaFile table)

Click the <Test database connection> button to check the connection to the database.

4. Configure the parameters for the task.

- **Delete entries for which no file was found:**

If disabled, "1" is entered in an additional column called `FILE_REFERENCE_VALID` for valid references and "0" for references for which no file could be found.

If you activate this option, entries marked with "0" in the `FILE_REFERENCE_VALID` column will be deleted from the index table.

- **Replace part of the reference file path:**

If disabled, the path entered in the database is used when accessing the data file.

If you enable this option, the first part of the file name (path) in the database column `_filename` will be replaced for access to the data file. The value "1" is entered in the `FILE_REFERENCE_VALID` column if the file exists in the specified target directory (*To*). The value "0" is entered in the `FILE_REFERENCE_VALID` column if the file is located in the original directory (*From*) and not in the target directory.

- **From:** Part of the original path name that is to be replaced.
- **To:** New part of the path name where the file is actually located.
- **UNC credentials:** If *ibaDatCoordinator* needs to access a network drive, enter the login details of an authorized user here.

Note



The `FILE_REFERENCE_VALID` column is automatically generated by *ibaDatCoordinator* if it does not already exist in the index table.

In *ibaDaVIS*, you can see the column in the configuration dialog of the data source if you have activated the display of the iba columns.

During synchronization, all processes are logged by the plug-in.

For more information about logging, see Chapter [Logging](#), page 29

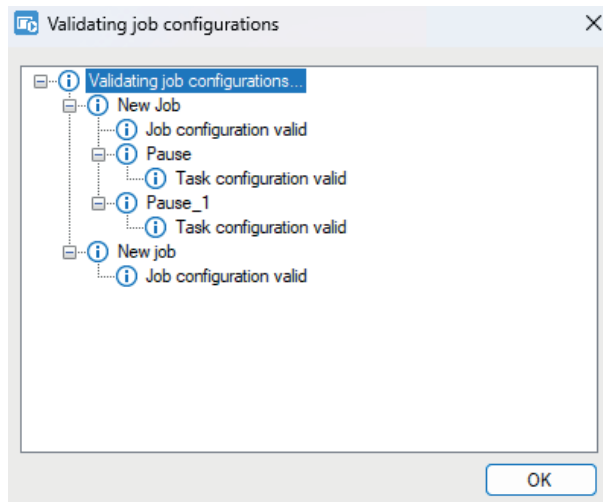
5.5 Starting and stopping jobs

Follow these steps to start and stop configured jobs in *ibaDatCoordinator*.

You can either start and stop all jobs at once or just individual jobs. To do this, use either the controls above the navigation menu (all jobs) or at the top of the configuration area for the respective job.

1. To start job processing, click .

→ *ibaDatCoordinator* validates the saved job configuration and displays the result in a window.



2. Click <OK> to confirm.

If there are no errors, the window closes automatically.

→ The view automatically switches to the status screen and displays the processing status of the jobs and their associated tasks, see [Job status](#), page 26.

3. To stop the job or all jobs and pause processing, click .

6 Application examples and FAQ

Here you will find useful application examples and answers to frequently asked questions about the *ibaDatCoordinator*.

6.1 Example: Using the merge task for multistation

The merge task in *ibaDatCoordinator* supports multistation operation by creating a single consolidated data file per event, even when the recording is distributed across multiple computers. The merged file provides a quick and comprehensive overview of the event and enables automated post-processing based on predefined analysis rules. Automated reporting workflows are also built on this concept.

Other documentation



Further information on *ibaPDA-Multistation* can be found in the corresponding documentation in the download area of the iba website.

The following example shows the configuration for automatically merging data files from multiple *ibaPDA* systems in multistation operation using *ibaDatCoordinator*. The measurement data from the *ibaPDA* systems is recorded synchronously, meaning that the data files have identical values in the "starttime" info field. For easier analysis, the data files should now be merged.

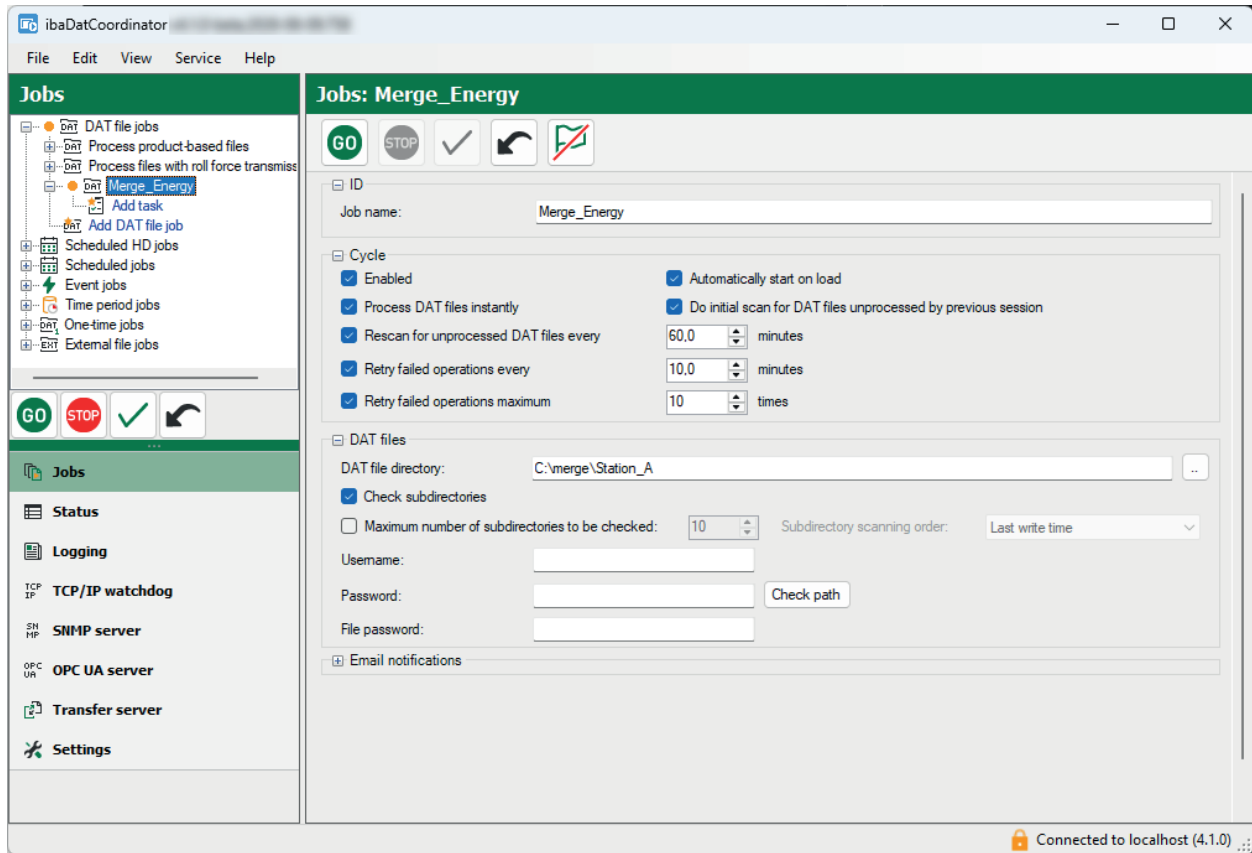
The data used in the example can be found on the "iba Software & Manuals" data medium [04_Libraries_and_Examples\110_ibaDatCoordinator-Multifile\02_Merge_Energy](#).

Prerequisite: The data files from the *ibaPDA* systems are available.

ibaPDA system A	ibaPDA system B
C:\merge\Station_A\StationA_2018-09-03_14.07.57_trigger.dat info clk: 5E-05 typ: real frames: 0000060000 starttime: 03.09.2018 14:07:57.834200 \$PDA_UtcOffset: +02:00 \$PDA_TimeSyncMode: 0 (None) starttrigger: 0000000000 start_event_type: \$PDA_MultiStation: standalone totalSignalCount: 2 version: ibaPDA 6.39.9 \$DATCOOR_status: processed ibaFiles: V7 2.1.2 (update) \$DATCOOR_times_tried: 1 filename: StationA_2018-09-03_14.07.57_trigger.dat 3. ibaMS4xAI-380VAC 4. ibaMS3xAI-1A/100A	C:\merge\Station_B\StationB_2018-09-03_14.07.57_trigger.dat info clk: 5E-05 typ: real frames: 0000060000 starttime: 03.09.2018 14:07:57.834200 \$PDA_UtcOffset: +02:00 \$PDA_TimeSyncMode: 0 (None) starttrigger: 0000000000 start_event_type: \$PDA_MultiStation: standalone totalSignalCount: 2 version: ibaPDA 6.39.9 \$DATCOOR_Merged: yes ibaFiles: V7 2.1.2 (update) filename: StationB_2018-09-03_14.07.57_trigger.dat 3. ibaMS4xAI-380VAC 4. ibaMS3xAI-1A/100A

Create job

To merge the files, you can use a DAT file job, for example.



1. Enter a job name (in the example, "Merge_Energy").
2. Select the DAT file directory.

The directory corresponds to the main directory of the merge task and, in the example, corresponds to the directory of an *ibaPDA* system (station A).

Create merge task

The screenshot shows the 'ibaDatCoordinator' application window. On the left is a 'Jobs' tree view with categories like 'DAT file jobs', 'Scheduled HD jobs', 'Event jobs', 'Time period jobs', 'One-time jobs', and 'External file jobs'. The 'Merge_Energy' job is selected. The main panel is titled 'Jobs: Merge_Energy - Merge'. It contains the following sections:

- ID:** Task name: Merge. ☐ This task is resource critical.
- Actions:** Execute: Always. Notify: Disabled.
- Merge settings:** Analysis file: C:\merge\Merge.pdo. Expression to merge: InfoFieldText(0, "starttime"). Text expression settings: Start: 0, Length: 0. ☐ Remove blanks at the end. ☐ Remove all blanks.
- Directories:** A table with columns: Directory, Username, Directory pa..., File password.

Directory	Username	Directory pa...	File password
The main directory: C:\merge\Station_A	...		
C:\merge\Station_B	...		
*	...		
- Test:** Example DAT file: . DAT file password (optional): . Copy from parent job.
- Target:** Target directory: C:\merge\merged. Username: . Password: . Check path.
- Subdirectories:** Directory naming: None.

At the bottom right, it says 'Connected to localhost (4.1.0)'.

1. Assign a name to the task (in the example, "Merge files based on start time").
2. Select the analysis file (here [Merge.pdo](#)).
3. Add additional directories in which the data files of other *ibaPDA* systems in the multistation network are stored and which are to be merged with the data files in the main directory.
4. Enter an expression for merging.

This expression defines the criteria according to which the files are to be merged. The expression in the example "InfoFieldText(0, "starttime")" merges the DAT files based on the info field "starttime".

Other documentation

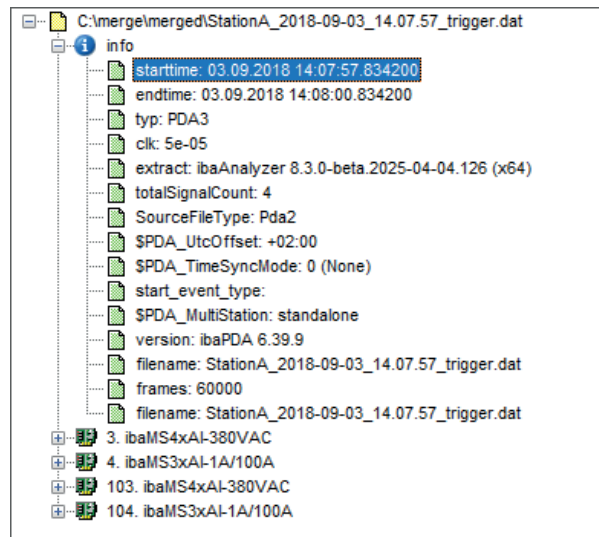


Detailed information on the use of expressions can be found in the *ibaAnalyzer* documentation, part 3 "Expression builder".

- Specify a target directory where the resulting data files should be saved (in the example, `C:\merge\merged`).

Execute job

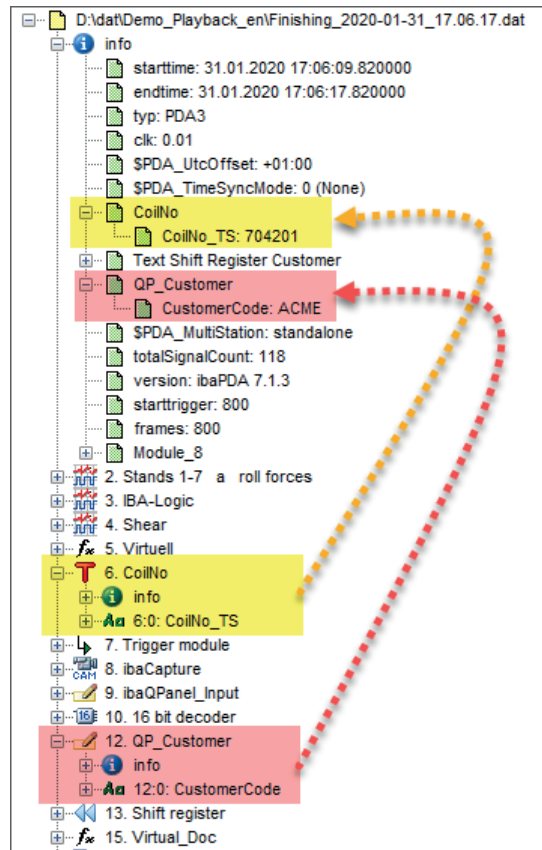
When you start the job, the files from the *ibaPDA* systems are merged and saved in the specified target directory. The following image shows the result of such a DAT file.



The image shows that the module numbers from the data files of *ibaPDA* System B were automatically changed from 3 to 103 and from 4 to 104, respectively.

6.2 Example: Using info fields for directory/file names

In a report task, both the names of the subdirectories and the names of the output files are to be formed depending on certain info fields.



The information for the info fields comes from text signals:

- "CoilNo_TS" from a text creator module
- "CustomerCode" from a Qpanel input

Note



If the contents of text signals are also to be available as info fields in the data file, the relevant text signals must be selected in the *ibaPDA* data store configuration, node *Files – Info fields*. Since an info field can only contain one value, the value of the text signal is written to the info field either at the beginning or at the end of the data file.

The contents of the `QP_Customer.CustomerCode` info field should be used for the subdirectory name.

Two info fields from the data file should be used for the file names of the output files:

- Customer code (`QP_Customer.CustomerCode`)
- Coil number (`Coil.No.CoilNo_TS`)

In the final file name, the two pieces of information should be separated by an underscore.

Sample: `CustomerCode_CoilNo.pdf`

Configuration

Enter the following information in *ibaDatCoordinator*:

Example of subdirectories:

☒ Use infofield/expression for subdirectory:

Start: Length:

Example for output files:

☒ Use infofield/expression for output file name:

Start: Length:

```
ConcatText ( InfoFieldText ( 0, "QP_Customer.CustomerCode" ), "_", InfoField ( 0, "CoilNo.CoilNo_TS" ) )
```

Tip



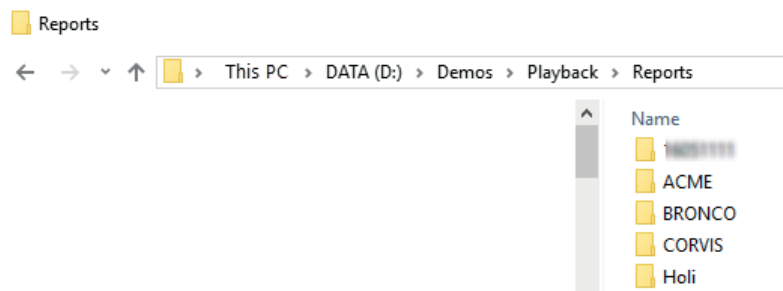
Use the expression builder in *ibaAnalyzer* to formulate a correct expression.

1. Open *ibaAnalyzer* and load a appropriate data file.
2. Open the expression builder either via the signal table (add signal) or the logical expressions (use a new, empty expression).
3. Enter the desired expression and click <OK>.
4. The final expression is displayed in the new signal line or in the new logical signal respectively.
5. Mark the entire expression and copy it to the clipboard.
6. Return to *ibaDatCoordinator* and paste the expression from the clipboard into the input field.

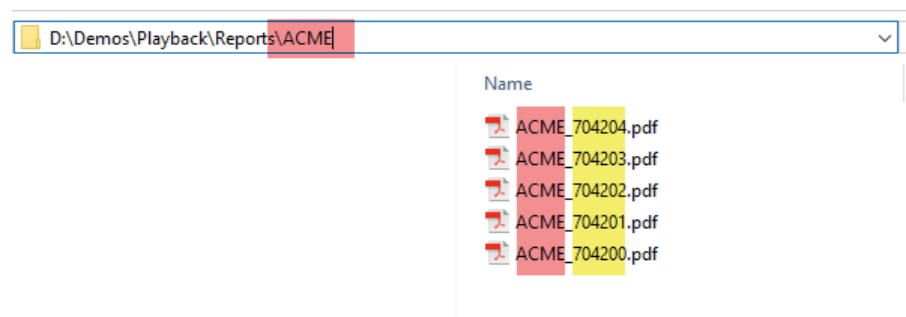
Result

Result in the file system after the job has been executed:

Subdirectories by customer ID in the target directory for reports



Report output files with file names from info fields



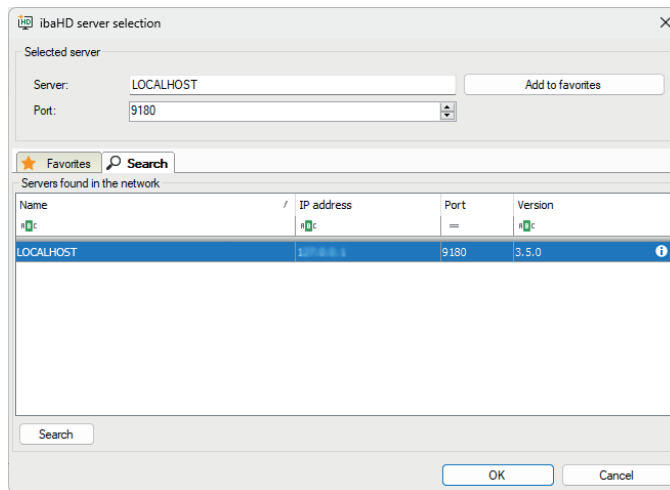
6.3 Establishing ibaHD-Server connection

The job types scheduled job, event job, and time period job, as well as the HD import task, require a connection to an *ibaHD-Server*.

In the dialog area for the *ibaHD-Server* connection, the currently connected *ibaHD-Server* and port, the user currently logged on to the *ibaHD-Server* and the available folders are always displayed.

To establish a connection to an *ibaHD-Server*, proceed as follows.

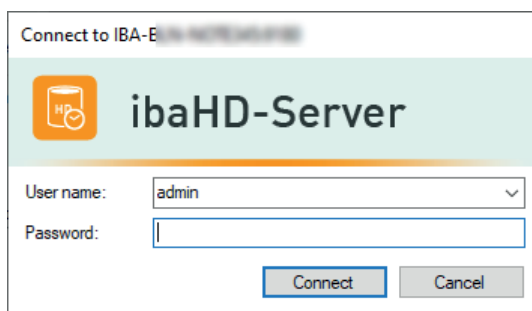
1. Click on the <Select server> button to open the selection dialog for the *ibaHD-Server*.



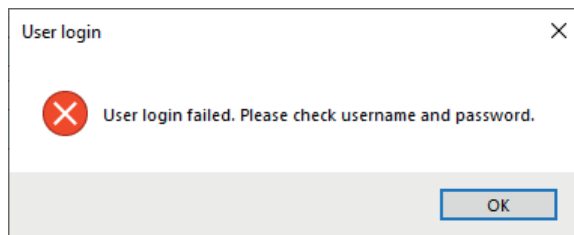
→ The table lists all *ibaHD-Servers* that are currently active in the network. If the desired *ibaHD-Server* is not displayed, try the <Search> button.

If the *ibaHD-Server* is not displayed even after the search, enter the IP address or the computer name of the *ibaHD-Server* in the address field above and click <OK>.

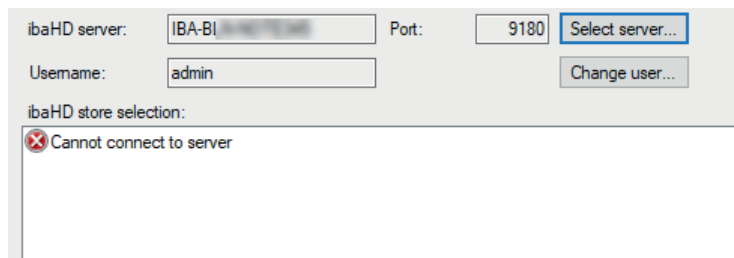
2. When the desired *ibaHD-Server* appears in the list, select it and close the dialog box by clicking <OK>.
3. If user management is enabled on the relevant *ibaHD-Server*, you must then enter or select a user name and enter the correct password.



→ If the password is not correct, an error message will appear.



→ If the user data is not entered correctly or the connection to the *ibaHD-Server* cannot be established for other reasons, another error message appears in the window for the shelves.



Check the configured users in the *ibaHD Manager* and, if necessary, create a suitable user or assign a new password.

6.4 Configuring service for other users

The following steps describe how to run the *ibaDatCoordinator* service under a different user account. Thereby, the interaction between *ibaDatCoordinator* and *ibaAnalyzer* is taken into account.

The names used in the examples are merely placeholders. Adapt the naming convention to your internal guidelines.

Other documentation



A detailed description of the following configuration steps can be found in the IT security guide in the download area of the iba website or in the iba help portal at <https://docs.iba-ag.com>.

Prerequisite: The *ibaDatCoordinator* and *ibaAnalyzer* software products are installed.

1. Create a managed service account on the server side.
2. Change the service account on the client side.
3. Set the necessary permissions for accessing the *ibaDatCoordinator* installation directory.
4. Set the DCOM permissions for communication between *ibaDatCoordinator* and *ibaAnalyzer*.
5. Add read and write permissions for the SNMP server in the registry.

7 Troubleshooting

In the following you will find help on possible errors when using *ibaDatCoordinator*. If you have any further questions, please contact the iba support.

7.1 Troubleshooting

Error indication	Possible cause	Corrective action
No connection to the server (GO/STOP buttons not available, red header bar in the navigation and display section)	Service is not running Port is already being used in the network Port blocked in firewall	Start service Change port number (Settings – Service) Release port in firewall or set permitted port (Settings – Service)
No editing of the configuration possible (orange header bar in the navigation and display section)	Edit mode inactive Server is configured by another client	Start editing
<i>ibaDatCoordinator</i> service cannot be started from client	Port is already being used in the network Port blocked in firewall Service disabled in Windows service administration	Change port number (Settings – Service) Release port in firewall or set permitted port (Settings – Service) Activate service in Windows service administration
<i>ibaDatCoordinator</i> client cannot be started	Client is already running in another session	Finish process in another session or "kill" it using the task manager.
Analyses with <i>ibaAnalyzer</i> are defective or do not work	The executing <i>ibaAnalyzer</i> instance does not "know" about important settings, macros or filters, because it is executed under another user account.	Transfer <i>ibaAnalyzer</i> settings from the client user account to the system account (Settings – <i>ibaAnalyzer</i> – <Transfer <i>ibaAnalyzer</i> settings>)
<i>ibaDatCoordinator</i> does not find any DAT files	Wrong directory path selected User account under which the <i>ibaDatCoordinator</i> service is running has no access authorization for the drive/directory	Correct the path Enter login data of an authorized user for the data file directory.

Error indication	Possible cause	Corrective action
DB sync task fails and/or the column FILE_REFERENCE_VALID has not been created in the index table of the database	The service is not authorized to write to the database	If the service is running on a system account, this account should be authorized to write on the database or Disable Windows authentication and enter the credentials of an authorized user in the settings of the database connection

8 Glossary

Below you will find an explanation of the most important iba terms.

Time period

Time periods mark a range of time inside a time-based HD store. The time period data is generated by *ibaPDA* and stored as a table in the *ibaHD-Server*. Time period data is useful for storing and labeling production data for analysis purposes or for marking data belonging to work shifts.

The start and end times and a unique identifier are stored for each time period. In addition to these standard parameters, calculated values or metadata from the production system can also be stored to enrich the information. You can then use the data for filtering or for any type of statistical process analysis.

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.

Product Security Incident Response Team (PSIRT) at iba

Current advisories: www.iba-ag.com/en/security

Email: psirt@iba-ag.com

Contact

Headquarters

iba AG
Gebhardtstrasse 10–20
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