

ibaDatCoordinator

Tool for automated data management

Manual

Issue 3.1

Measurement Systems for Industry and Energy

www.iba-ag.com

Manufacturer

iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

Contacts

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

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

© iba AG 2024, All rights reserved.

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

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

Version	Date	Revision	Author	Version SW
3.1	02-2024	Installer, time period job, merge task	TS, st	3.1

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	6
1.1	Target group and previous knowledge	6
1.2	Notations	6
1.3	Used symbols.....	7
2	Basic principles	8
2.1	Functions and applications	8
2.2	Licenses	10
2.3	System requirements.....	12
2.4	Installation	13
2.4.1	Installation via command line.....	15
2.5	Starting the program	16
2.5.1	ibaDatCoordinator client	16
2.5.2	ibaDatCoordinator status	18
2.6	ibaDatCoordinator status application.....	19
2.7	Features of the client-server installation	21
3	User interface and configuration	22
3.1	The menu bar	23
3.2	Navigation menu	24
3.2.1	Configuring jobs and tasks.....	24
3.2.2	Status.....	25
3.2.3	Logging	27
3.2.4	Data Transfer Server	29
3.3	Monitoring.....	32
3.3.1	TCP/IP Watchdog	32
3.3.1.1	Watchdog message.....	33
3.3.2	SNMP server	34
3.3.2.1	Configuration tab.....	34
3.3.2.2	Objects tab	36
3.3.2.3	System-OIDs 1.3.6.1.2.1.1.....	38
3.3.2.4	Diagnostics tab	39
3.3.3	OPC UA Server	40

3.3.3.1	OPC UA server - Configuration.....	41
3.3.3.2	OPC UA server - Tags	43
3.3.3.3	Diagnostics tab	44
3.4	Settings	44
3.4.1	dat file processing.....	45
3.4.2	ibaAnalyzer	46
3.4.3	Password protection	47
3.4.4	Global cleanup	48
3.4.5	Certificates.....	49
3.4.5.1	Central certificate store	50
3.4.5.2	Manage certificates	51
3.4.5.3	Use certificates	54
3.4.5.4	Save and protect certificates	54
3.4.6	Databases	55
4	Processing data - ibaDatCoordinator Jobs.....	56
4.1	General settings.....	57
4.2	dat-file triggered job.....	59
4.3	Scheduled job	64
4.4	Event jobs	70
4.5	Time period job	72
4.5.1	Properties of the time period table	75
4.6	One time job.....	77
4.7	External File Triggered Jobs	78
5	ibaDatCoordinator - Performing tasks	82
5.1	General settings for the task.....	83
5.2	Report task	89
5.3	Extraction task	92
5.4	Script task	94
5.5	Copy task	95
5.6	Upload task.....	96
5.7	Condition task.....	98
5.8	Update data task.....	101
5.9	Pause task.....	104

5.10	Cleanup task	105
5.11	Synchronization task.....	107
5.12	Splitter task.....	110
5.13	HD Offline event task.....	113
5.13.1	Task specific settings.....	113
5.13.2	Configuration tab.....	116
5.13.3	Client options tab	122
5.14	Kafka Publish task	122
5.14.1	Connection	122
5.14.1.1	Kafka type	123
5.14.1.2	Type Event Hub.....	128
5.14.2	Target.....	129
5.15	SNMP Publish task	132
5.16	OPC UA Publish task	134
5.17	Transfer Data task	135
5.18	Merge task.....	137
5.19	ibaHD import task.....	140
5.20	S7 writer task.....	141
6	Appendix	144
6.1	ibaHD server connection and user management	144
6.2	Configuration of queries.....	145
6.3	Using infofields for directory names/filenames.....	150
6.4	Configure ibaDatCoordinator service for other user	152
6.4.1	Server side	152
6.4.1.1	Create a managed service account.....	152
6.4.2	Client side	153
6.4.2.1	Changing the service account.....	153
6.4.2.2	Setting permissions.....	156
6.5	Troubleshooting.....	167
7	Support and contact.....	168

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 the 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 Basic principles

ibaDatCoordinator is the central component for automatic processing of measurement data and is hence vital for the use of the iba system in productive systems. All iba measurement files generated by *ibaPDA* or other systems (e.g. *ibaQDR* or *ibaLogic*) as well as data from HD stores of *ibaHD-Server* can be processed automatically.

2.1 Functions and applications

The main task of *ibaDatCoordinator* is the post-processing of measurement data, which can be available in different forms. In addition to processing DAT files and data from *ibaHD-Server*, external formats can also be processed (as a fee-based add-on). Depending on the data source, a suitable "job" must be created.

Available jobs:

.dat-file triggered job	Automatic processing of DAT files in one or more directories
Scheduled job	Processes data that is queried from an <i>ibaHD-Server</i> at regular intervals
Event job	Processes data that are retrieved from an <i>ibaHD-Server</i> based on events (<i>ibaHD</i> event store).
Time period job	Processes data that are retrieved from time periods of an <i>ibaHD-Server</i>
One time job	
External file triggered job	Responds to various external formats (currently: CSV, DAS, COMTRADE, PARQUET) and converts the data to DAT format. Subject to license.

Details about the different job types can be found in chapter [↗ Processing data - *ibaDatCoordinator* Jobs](#), page 56.

While the job type determines the data source, any number of tasks can be created for each job (exception: the external file triggered job) to control the further processing of the data.

The following tasks are available:

Report task	Generates a report based on an analysis file with the report generator of <i>ibaAnalyzer</i> .	
Extract task	Extracts data into a database or into a new file (<i>iba</i> DAT file or external format).	Subject to license
Script task	Executes arbitrary scripts.	
Copy task	Copies data files to another location.	
Upload task	Uploads data files to different data storage servers.	
Condition	Implements a branching for task processing based on defined criteria in the data file/HD query.	

Pause task	Interrupts the processing for an adjustable time span.	
Cleanup task	Deletes files with an configurable filename extension from a folder.	
Splitter task	Splits one data file into two parts based on an expression or analysis file.	
HD offline event task	Generates an event, based on a data file or HD query together with rules defined in an analysis file and writes it into an HD event store.	
Kafka Publish task	Publishes values calculated on the basis of an analysis file to an Apache Kafka cluster.	Subject to license
SNMP Publish task	Publishes values calculated on the basis of an analysis file using the integrated SNMP server.	Subject to license
OPC UA Publish task	Publishes values calculated on the basis of an analysis file using the integrated OPC UA server.	Subject to license
Transfer data task	Transfers measurement files to another <i>ibaDatCoordinator</i> instance.	
Merge task	Merges several DAT files into one DAT file.	Subject to license
ibaHD import task	Imports data files into an HD data store.	
Update data task	Writes subsequently defined data from a database into a processed data file.	Subject to license

Available as plug-in:

S7 writer task	Writes values from a data file or analysis into an S7 data bloc.	Subject to license
Synchronization task	Synchronizes database and file system.	

Details about the different tasks can be found in chapter [🔗 *ibaDatCoordinator - Performing tasks*](#), page 82.

2.2 Licenses

ibaDatCoordinator is basically a freely installable and usable software product. A comprehensive basic functionality can be used free of charge, for more advanced functions the appropriate licenses per configured task or job have to be purchased.

The tasks for simple processing of DAT files (copy, cleanup, upload, report and splitter task), the general tasks (pause, condition, script and transfer data task) as well as the ibaHD tasks (ibaHD import and HD offline event task) can be used free of charge and allow the configuration of extensive automated processing workflows.

Automated extraction of data (time series and calculated values) in databases requires one purchased license per configured task (*ibaDatCoordinator-DB*). Extracting data in files and the merging of several DAT files into one DAT file require one purchased *ibaDatCoordinator-File-Extract* license per configured task.

The automated conversion of files from external formats into DAT files requires one purchased license (*ibaDatCoordinator Convert-CSV-to-dat / DAS-to-dat / COMTRADE-to-dat / Parquet-to-dat*) per configured job, depending on the external format used.

Publishing calculated values via one of the publish tasks (OPC UA, SNMP or Kafka) requires the purchased *ibaDatCoordinator-Publish* license per configured task. In case of an extraction in databases, this license is also sufficient, as long as only scalar values ('Info assignments' and 'Calculated columns' in the ibaAnalyzer data extractor) and no time series are extracted.

The special tasks for the subsequent update of measurement files (Data update task) and for writing into SIMATIC S7 data blocks (S7 Writer task) also require separate licenses but are not limited by counters per configured task (*ibaDatCoordinator-Update Data Task, ibaDatCoordinator S7 Writer*).

Attractively priced 10-task bundles are also available for some task types.

ibaDatCoordinator licenses

34.010550	ibaDatCoordinator	Tool for data management automation
34.010510	ibaDatCoordinator-DB	Automatically extract data to databases (per task)
34.010511	ibaDatCoordinator-File-Extract	Automatically extract data to DAT files or other file formats such as CSV, COMTRADE, Parquet, Matlab, TDMS (per task)
34.010521	ibaDatCoordinator-File-Extract-10	Bundle for automated file extract (10 tasks)
34.010512	ibaDatCoordinator Convert-CSV-to-dat	Automatically convert CSV files to DAT files (per job)
34.010513	ibaDatCoordinator Convert-DAS-to-dat	Automatically convert DAS files from Danieli FDA systems to DAT files (per job) limited to the 32-bit version of ibaAnalyzer
34.010514	ibaDatCoordinator Convert-COMTRADE-to-dat	Automatically convert COMTRADE files to DAT files (per job)

34.010516	ibaDatCoordinator Convert-PARQUET-to-dat	Automatically convert PARQUET files to DAT files (per job)
34.010515	ibaDatCoordinator-Publish	Publish computed values via OPC UA, SNMP, Apache Kafka, or SQL database (per task)
34.010525	ibaDatCoordinator-Publish-10	Bundle to publish computed values (10 tasks)
34.010552	ibaDatCoordinator-Update Data Task	Plugin for input function, subsequent entries in measurement files
34.010556	ibaDatCoordinator S7 Writer	Extracting data from a measurement file and writing it to data blocks (DB) of an S7 PLC

ibaAnalyzer licenses

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

2.3 System requirements

Hardware

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

Software

- Windows 2012 Server (x64), Windows 8.1 (x86/x64), 2012 Server R2 (x64), Windows 10 (x86/x64), Windows 11, Windows Server 2016, Windows Server 2019, Windows Server 2022
- .NET- Framework v4.8
- *ibaAnalyzer* v8.0.0 or higher

These specifications refer to *ibaDatCoordinator* v3.0.

You can open the version history via the *Help - version history* menu in *ibaDatCoordinator* client. All relevant requirements regarding the operating system and other iba components for the respective version of *ibaDatCoordinator* are listed in the "Version compatibility" section.

Compatibility to other iba products

ibaAnalyzer is the essential application for the function of *ibaDatCoordinator* as it processes many tasks in the background, such as extraction and reporting.

Note



Most of the functions of *ibaDatCoordinator* require *ibaAnalyzer* to be installed as well. We recommend to always use the latest available version of *ibaAnalyzer*.

In some cases, certain versions of other applications like *ibaPDA* or *ibaHD-Server* are also required.

Note



To use certain functions of *ibaDatCoordinator*, a connection to *ibaHD-Server* must be possible.

Due to these cross-dependencies iba recommends installing the most recent versions of the involved applications.

2.4 Installation

If you have a ZIP file of the new software version (for example after download), unzip it into a (temporary) folder of your choice.

You will find the file in the ...\\01_ibaSoftware\\ibaDatCoordinator directory of the “iba Software & Manuals” data medium.

Execute the “ibaDatCoordinatorSetup_vx.y.z.exe” file and follow the instructions in the installation wizard.

If an older version of the software is already installed, you can simply install the new one. The older version of the program is automatically removed after a prompt and confirmation. The settings and configurations chosen with the older version are preserved.

Note



Some options only appear during the first installation and not during an update. If you manually uninstall *ibaDatCoordinator* before installing the new version, the installation behaves like the first installation

For installation, the following steps are required:

1. On the welcome page, you can check the version history file before installing the software.
2. Read and accept the license agreement.
3. Select the installation folder (first installation only).
4. Select the components to be installed.

Select one of the three options:

- Client and Server Standard installation.

ibaDatCoordinator is installed and executed as a service. The configuration is done via the client application.

- Client only

Can be selected if no *ibaDatCoordinator* service is to be operated on the PC, but only the client is needed for remote configuration on another system.

- Standalone executable

In this case, *ibaDatCoordinator* does not run as a service, but locally in a separate process. This installation type is recommended for testing purposes. For productive use, iba recommends the installation as client/ server.

Optional:

- ibaManagementStudioAgent

If there is an *ibaManagementStudio* server available in the network you can manage this installation by *ibaManagementStudio*. Therefore, you have to enable this option in order to install the *ibaManagementStudio* agent. If this program is already installed on your computer, you can disable this option.

5. Update of “WIBU Codemeter Runtime” (only shown if required)

If you are using the software “WIBU Codemeter Runtime” for licensing, the installer verifies

the version of this software. If its version is outdated the installer offers to update this software. Please note, that other iba applications may use this software for licensing as well and be subject to stop and restart in the course of updating. If you don't want to update the software at this point, just uncheck this option.

6. Select the user account (first installation only):

- Local System Account
- Custom User

7. Additional tasks page

Select if Desktop Shortcut should be created during installation.

8. On the finish page, you can select to automatically start *ibaDatCoordinator* after closing the wizard.

Note

When installing 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 a high demand on resources are to be performed, it is recommended to perform optimization. This can be manually done later via the *ibaDatCoordinator* status application.

When installing as a service, the *ibaDatCoordinator* Status program is also installed. It is used for operation and diagnosis of the *ibaDatCoordinator* service. After installation, the *ibaDatCoordinator* status icon appears in the Windows taskbar.

The different symbols indicate the following states:



ibaDatCoordinator service running



ibaDatCoordinator service stopped

Other documentation

The installation program is based on the Inno Setup program. Further information can be found in the Inno Setup documentation:

<https://jrsoftware.org/ishelp/index.php?topic=setupcmdline>

2.4.1 Installation via command line

The installation of *ibaDatCoordinator* can also be started via a command line. This is helpful for central software administration or when using deployment systems.

The installation process can also run in the so-called "Silent Mode", meaning that there is no interaction with the user (clicking the <Next> button in the installation wizard).

You can control how the installation should take place and which components should be installed with the command line switches that are set after the call of the installation program.

Notation:

/HELP	This switch shows a message box containing all supported command line parameters.
/SILENT	This switch installs <i>ibaDatCoordinator</i> 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.
/SUPPRESS-MSGBOXES	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 button 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, zhd. Examples: /LANG=en /LANG=de

/COMPONENTS=	<p>Add the names of the components to be installed right after the equal sign in a comma-separated list. Only the components specified in the list will be installed.</p> <p>Possible options: ibadatcoordinator\server, ibadatcoordinator\client, ibadatcoordinator\standalone, ibamanagementstudio</p> <p>Examples:</p> <p>/COMPONENTS=ibadatcoordinator\client</p> <p>/COMPONENTS=ibadatcoordinator\server,ibamanagementstudioagent</p>
/PORT=	<p>With this switch, you select the TCP port over which <i>ibaDatCoordinator</i> service communicates with the client. The port can be set later in the <i>ibaDatCoordinator</i> Status application.</p> <p>This switch can only be used together with the installation of client/server.</p> <p>Default setting: 8800 or setting from previous installation</p>

Example:

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

2.5 Starting the program

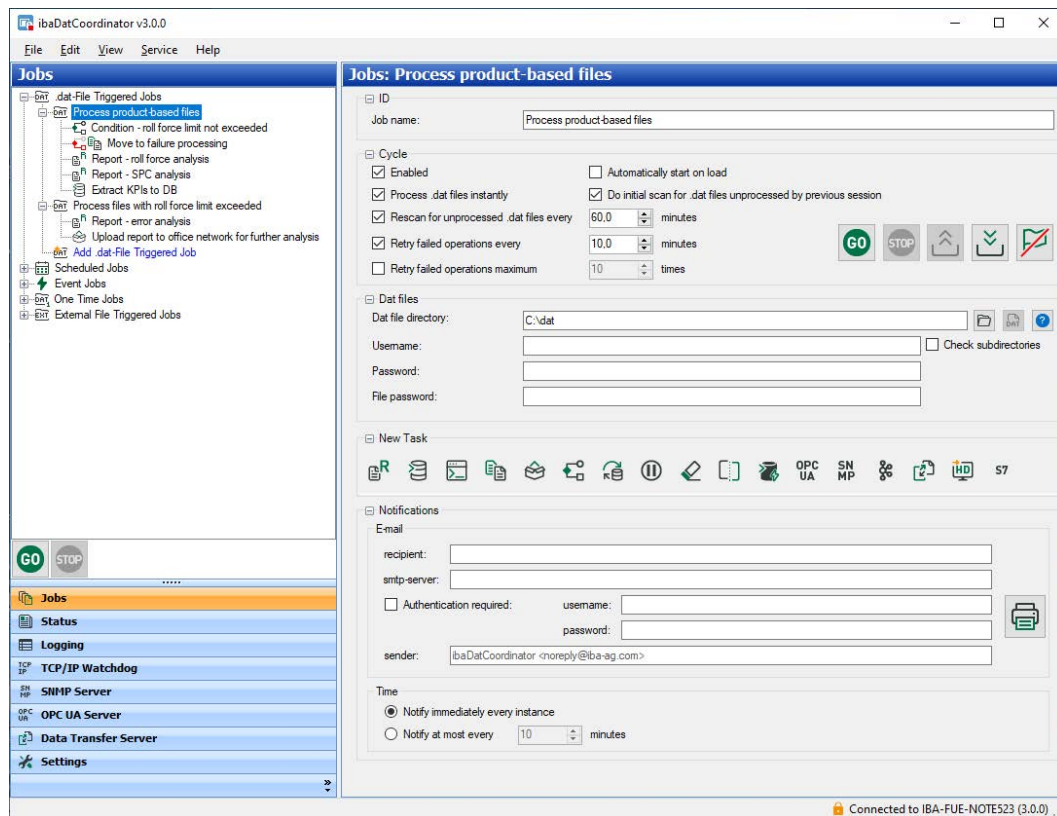
2.5.1 ibaDatCoordinator client

There are different ways to start the *ibaDatCoordinator* program or client:

The applications can be found in the start menu under:

- Windows start menu – *All programs – iba – ibaDatCoordinator*

If *ibaDatCoordinator* client is used, it automatically tries to connect to the last connected server. The currently connected server is displayed in the status bar of the program (bottom right).

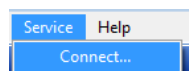


Note

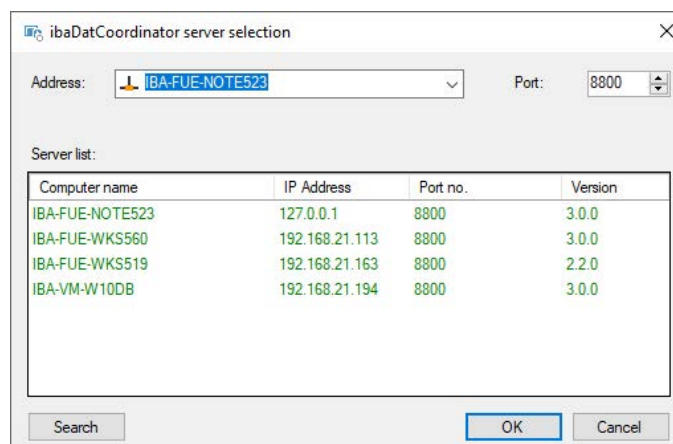


Normally, the communication is encrypted. Only if the *ibaDatCoordinator* service version does not support this, the configuration is transmitted unencrypted.

If you want to select a different server, open the *Service* menu and select *Connect*.



Select the server to be connected to the client in the dialog.



The selection of the *ibaDatCoordinator* server is done either by double-clicking the server in the list or by selecting the server and then clicking <OK>.

The <Search> button starts and stops the search for active *ibaDatCoordinator* servers on the computer and network. The search process is displayed by a blue progress bar in the dialog window.

If the server is not found in the network, it is possible to enter the IP address and port manually. The default port is 8800.

The field has a drop-down list that allows you to select a server from the 10 most recently connected servers.

If you confirm the dialog with <OK>, the client attempts to connect to the selected server by the specified port. If you leave the dialog using <Cancel>, the client remains connected to the previously connected server.

Note



When installing *ibaDatCoordinator* as a program, it is not necessary to connect to a server.

Further details on operation and configuration can be found in chapter [➤ User interface and configuration](#), page 22.

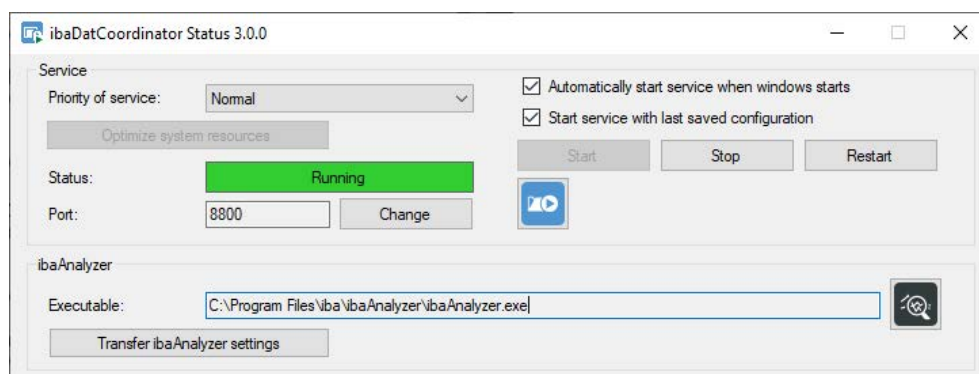
2.5.2 ibaDatCoordinator status

The *ibaDatCoordinator* status application is only available if the *ibaDatCoordinator* service is installed on the PC.

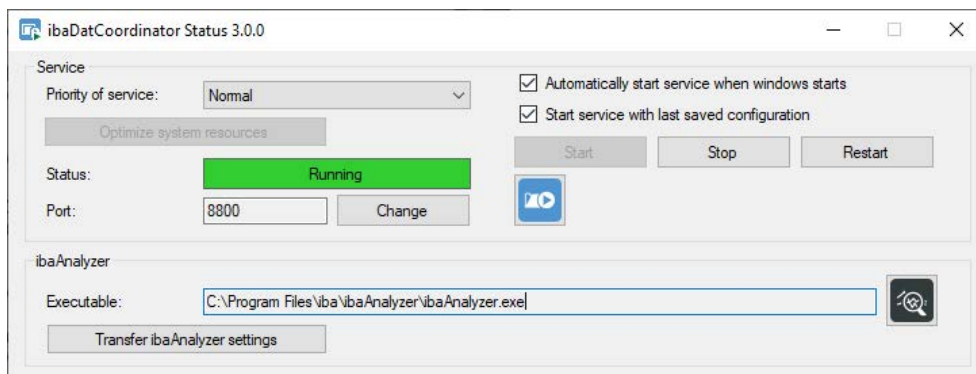
The application can be started in different ways:

- Double-click the *ibaDatCoordinator* icon in the taskbar
- Right-click the *ibaDatCoordinator* icon in the taskbar and select *restore* in the context menu.
- In the Windows "Start" menu: *All Programs – iba – ibaDatCoordinator – ibaDatCoordinator Server Status*

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



2.6 ibaDatCoordinator status application



The *ibaDatCoordinator* status application provides functions to monitor and configure the *ibaDatCoordinator* service.

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. If you lower the level, more CPU time is provided to other processes. Keep the setting "normal" if you use only one or a few jobs and there is enough time to complete the task.

Choose a higher priority if there is less time available and the execution of the job should not be interrupted by other processes. If the priority of service is changed, the *ibaDatCoordinator* service will need to be restarted.

Automatically start service when windows starts

When this option is enabled, the *ibaDatCoordinator* service starts automatically when Windows is starting.

Button <Optimize system resources>

If you click 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 done, the number of simultaneously available *ibaAnalyzer* instances is reduced to a maximum of 6.

When clicking the button, *ibaDatCoordinator* stores a back-up copy of the 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> buttons

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

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

Port

This is the port number through which the *ibaDatCoordinator* service communicates with the client. The port number can be adjusted via the <Change> button if necessary.


Default setting: 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

Clicking the *ibaDatCoordinator* symbol  opens the *ibaDatCoordinator* client which automatically connects to the correct server.

ibaAnalyzer

Executable

This displays the path to the *ibaAnalyzer.exe* application (read-only). When you click on the button with the *ibaAnalyzer* symbol, *ibaAnalyzer* is started. If no *ibaAnalyzer* installation was found, this is also displayed.

When you click the button with the *ibaAnalyzer* icon, *ibaAnalyzer* is started so that you can verify your setting.

Button <Transfer ibaAnalyzer settings>

Settings of *ibaAnalyzer*, like default settings, macros and filters are usually stored user-related. 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 then not work.

When clicking the button, *ibaAnalyzer* settings are transferred from the local user account to the system account so that they are available for the execution of analyses by *ibaDatCoordinator* and/or *ibaAnalyzer*.

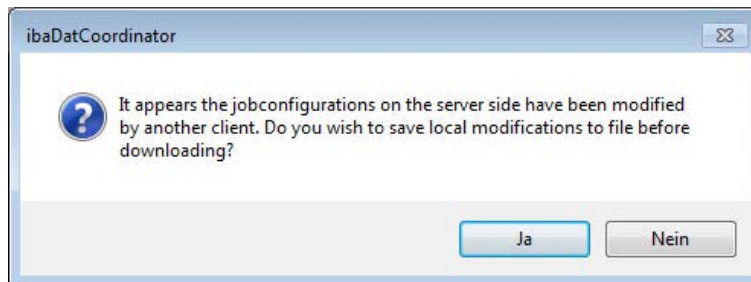
This applies in particular to the printer settings. It may be necessary to run the service under a different user in order to use network printers.

2.7 Features of the client-server installation

Multiple client connections

It is possible for multiple clients to connect to the same *ibaDatCoordinator* server. This means that multiple clients can modify the jobs and tasks of *ibaDatCoordinator* on the server.

Clients can detect whether another client has made changes. If this is the case, a dialog appears where the client is able to save local changes before the changes of the other client are applied to the current client.



To prevent multiple clients from modifying the server configuration, the password functionality can be enabled in the *Settings* dialog of the navigation menu.

Executing the service on a remote system

If the service is executed on a remote system, analysis files, scripts, and source and target directories should be 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.

Clicking the browse button where you can specify such paths opens a dialog 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 shared network location), a copy of the analysis file is downloaded to the client system each time. If changes have been made, the file can be uploaded to the server manually.

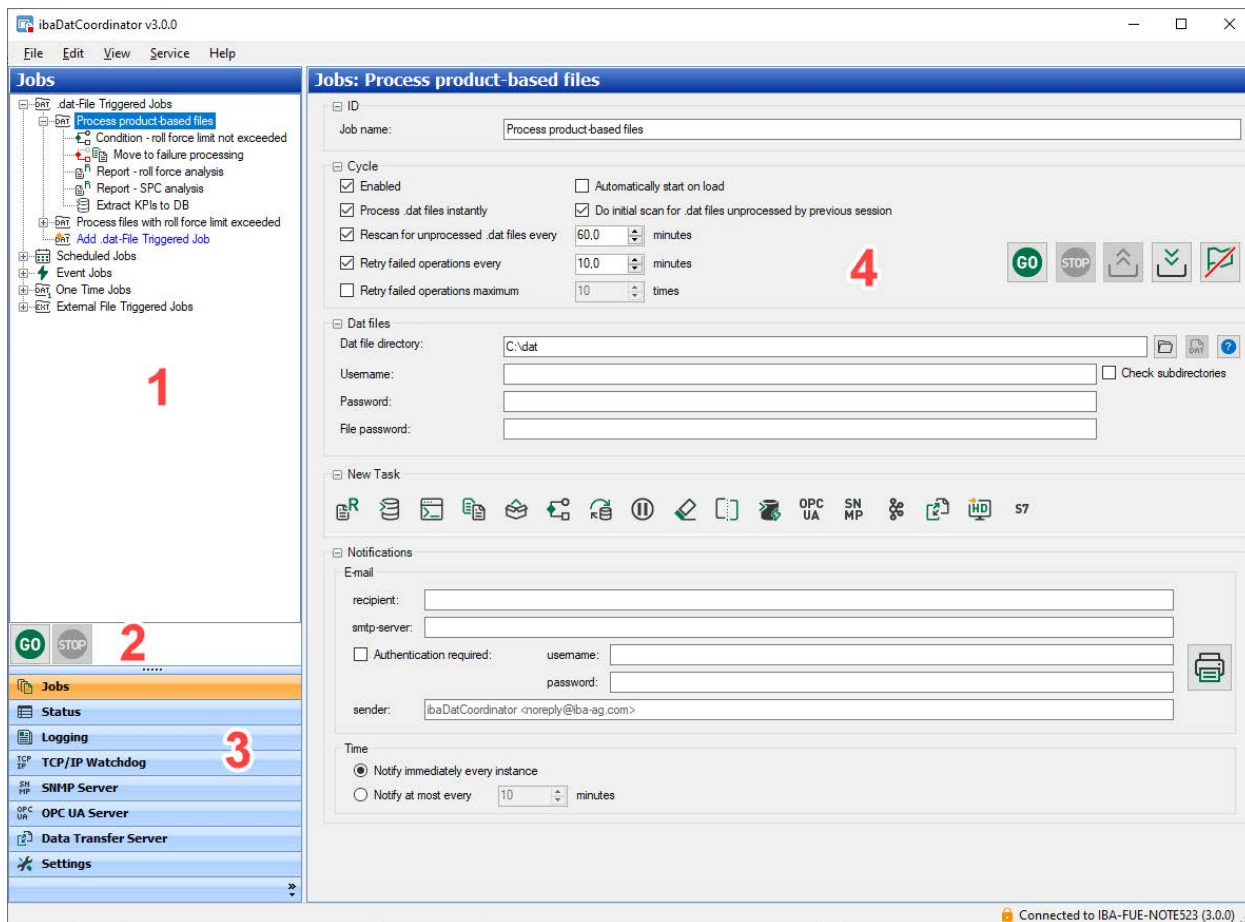
Plugins

If plugins are to be used in the *ibaDatCoordinator*, these must be installed 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.

3 User interface and configuration

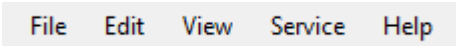
After the *ibaDatCoordinator* program or client has started, the main window is opened.



The main window is made up of 4 sections:

1. Navigation area, e.g. list of jobs and tasks
2. Start and stop button (for all jobs)
3. Navigation menu
4. Display and configuration area

3.1 The menu bar



The menu bar contains the main control and navigation functions.

Click the *ibaDatCoordinator* icon in the top-left corner to display the menu. You can then select the options in this menu to minimize or move the *ibaDatCoordinator* or *ibaDatCoordinator* client window, as well as reduce it to its minimum size or close it entirely.

File

The *File* menu is primarily used for handling configuration files.

New	With this option, you can create a new <i>ibaDatCoordinator</i> project. It removes all the configured jobs and creates a new project.
Open	Opens an <i>ibaDatCoordinator</i> project (XML file) that has been saved before.
Save	With this option, you can save the changes in the currently opened project.
Save as...	Saves a created project as an XML file.
Exit	With "Exit", you can close the <i>ibaDatCoordinator</i> client. If <i>ibaDatCoordinator</i> is installed as service, the service and hence all jobs continue running.

Edit

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 and task list.

View

The *View* menu provides all functions that are also available in the navigation bar. See also chapter [↗ Navigation menu](#), page 24.

In addition, language switching is possible here.

Service

The *Service* menu is only available if *ibaDatCoordinator* was installed as a service (client server). Select *Service – Connect* to connect the *ibaDatCoordinator* client to another server.

Help

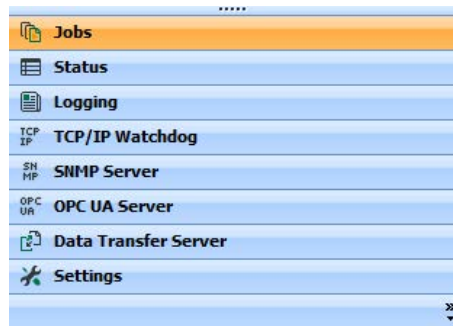
In the *Help* menu, you can open the *ibaDatCoordinator* manual as html file.

In a support case, you can generate a ZIP file with *Save information for iba support*. In this file, all relevant settings and configurations can be 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.

Moreover, you will find information about the program itself and the version history in the *Help* menu. The version history opens a file in which new functions and as well as bug fixes are documented.

3.2 Navigation menu



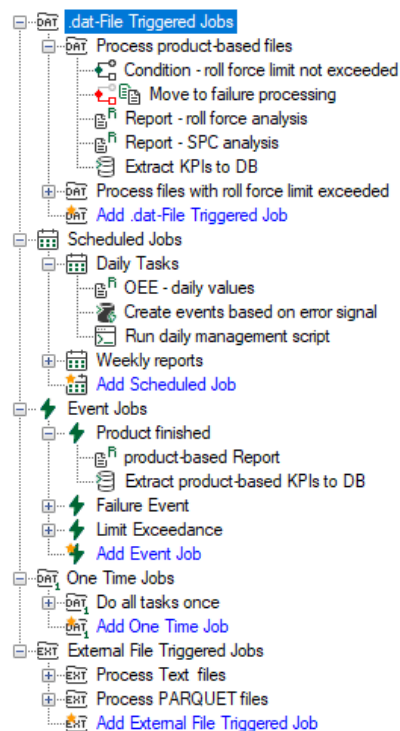
At the bottom left of the main window there is a navigation menu that allows you to switch between the most important views of *ibaDatCoordinator*. The navigation menu and the views are the main components used to configure *ibaDatCoordinator*.

The different views in *ibaDatCoordinator* are as follows:

Jobs	Allows the configuration of jobs and tasks. See also chapter ↗ Processing data - ibaDatCoordinator Jobs , page 56. And see chapter ↗ ibaDatCoordinator - Performing tasks , page 82 for the configuration of the individual jobs and tasks
Status	Displays the status of each job in tabular form. See also chapter ↗ Status , page 25
Logging	Displays all chronologically sorted log entries. See also chapter ↗ Logging , page 27.
TCP/IP Watchdog	Configuration of watchdog telegrams for monitoring, see also chapter ↗ TCP/IP Watchdog , page 32.
SNMP Server	Configuration of the integrated SNMP server, see also chapter ↗ SNMP server , page 34
OPC UA Server	Configuration of the integrated OPC UA Servers, see also ↗ OPC UA Server , page 40.
Data Transfer Server	Configuration of the data transfer server, see also chapter ↗ Data Transfer Server , page 29.
Settings	General settings, see also chapter ↗ Settings , page 44.

3.2.1 Configuring jobs and tasks

If you choose the "Jobs" view in the navigation menu, the tree with the configured jobs and tasks is shown above the navigation menu. The tree consists of job nodes and an "Add job" node. If you click on the "Add job" node, a new job is created. If you choose a job, the configuration dialog for the job is displayed on the right side of the window, where you can make additional configurations.



To each job node, one or more task nodes can be subordinated. Task nodes only appear when you have added tasks to the job configuration. If you choose a task node, the configuration dialog for the task is displayed on the right-hand side of the window, where you can make additional configurations.

Directly above the navigation menu there is a "GO" and "STOP" button.

If you click on one of these buttons, it has an influence on all jobs in the tree. A "GO" and a "STOP" button is also in the configuration dialog of a job. In this case, the chosen action only affects the selected job.

For more information about adding and configuring jobs, see chapter [Processing data - ibaDatCoordinator Jobs](#), page 56 and [ibaDatCoordinator - Performing tasks](#), page 82.

3.2.2 Status

If you select the "Status" view in the navigation menu, you can follow the status of the currently configured jobs. The job can be selected on the left side in the tree. On the right side, the tasks for every DAT file are displayed in columns.

In the status view, an overview for each job can be opened in a table. For this purpose, just select the desired job on the left. In the table, every row stands for a DAT file.

The columns for the file name and the previous number of attempts to process the file are fixed. For each task that is configured in the respective job, another column is added that has the task name as its title. With the checkbox, the corresponding column can be hidden from the table. The <Show all columns> button displays all hidden columns again.

The status is indicated by a symbol:

Empty	The task has not been processed, yet or has not been processed due to a certain condition.
Task symbol	The task is currently being processed.
Task symbol with a blue checkmark	The task has been processed successfully and the condition was TRUE.
Task symbol with a red cross	The task could not be executed or the condition was FALSE.
Task symbol with a "No access" sign	The DAT file currently cannot be accessed.

The screenshot shows the ibaDatCoordinator v3.1.0 application. The main window is titled 'Status: process_dat_files'. On the left, there is a sidebar with a tree view showing the project structure. The 'process_dat_files' task is selected and highlighted with a red box. The main area displays a table with the following columns: 'Finish processing time', 'iba .dat-file', 'Attempts', 'Publish_KPI', 'check_plausibility', and 'Move_to_archive'. The table contains 11 rows of data, each representing a .dat file and its processing status. A red box highlights the 'Show all columns' button in the top right corner of the table. Below the table, a status bar indicates 'The job is started' and 'Connected to IBA-FUE-WKS563 (3.1.0)'.

Above the table, the job name can be found as hyperlink. When clicking on this hyperlink the corresponding job configuration panel is opened. Below the table, the status of the selected job is indicated.

All columns in the table can be sorted and filtered as usual. The columns for the task status can be filtered to show only the files for which the task failed, was successfully processed, timed out, etc.

If you right-click the table, a context menu opens with the following commands:

Open in ibaAnalyzer

The processed file can be opened with *ibaAnalyzer* for further analysis. Note that this option is only available if the file can be found in its original location.

Open in ibaAnalyzer with PDO

Like the previous option but additionally loads the analysis file configured for the task. The PDO-file offered depends on the column which is clicked.

Show full path

For all files shown in the table, the full path or only the filename can be shown.

Clear filters

If any filters are active on the table, they can be reset.

The DAT files that show a constant failure are added to a special list for these files. This list is only generated if the option "Retry failed operations maximum ... times" is enabled in the configuration window for the job and the configured number has been exceeded.

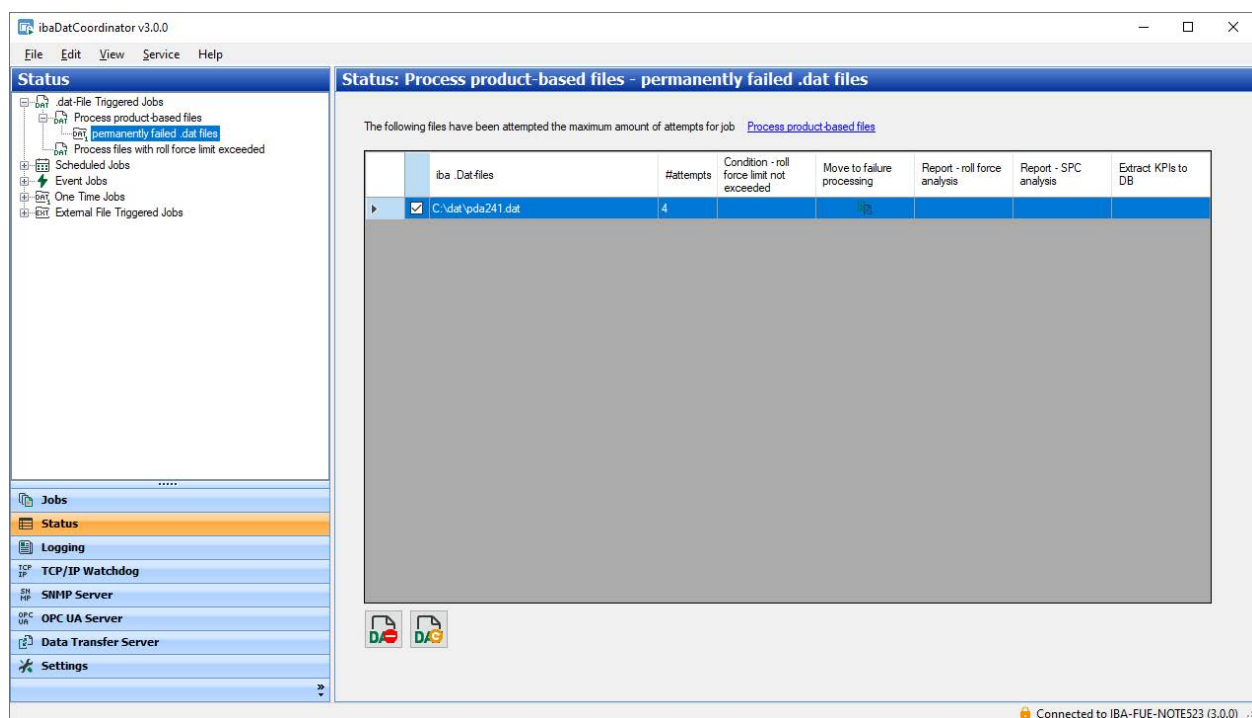
In addition to the normal status view it is possible to select one or more files and to delete or reprocess them.



Deleting files that show a constant failure.



Cleaning up the files that show a constant failure.



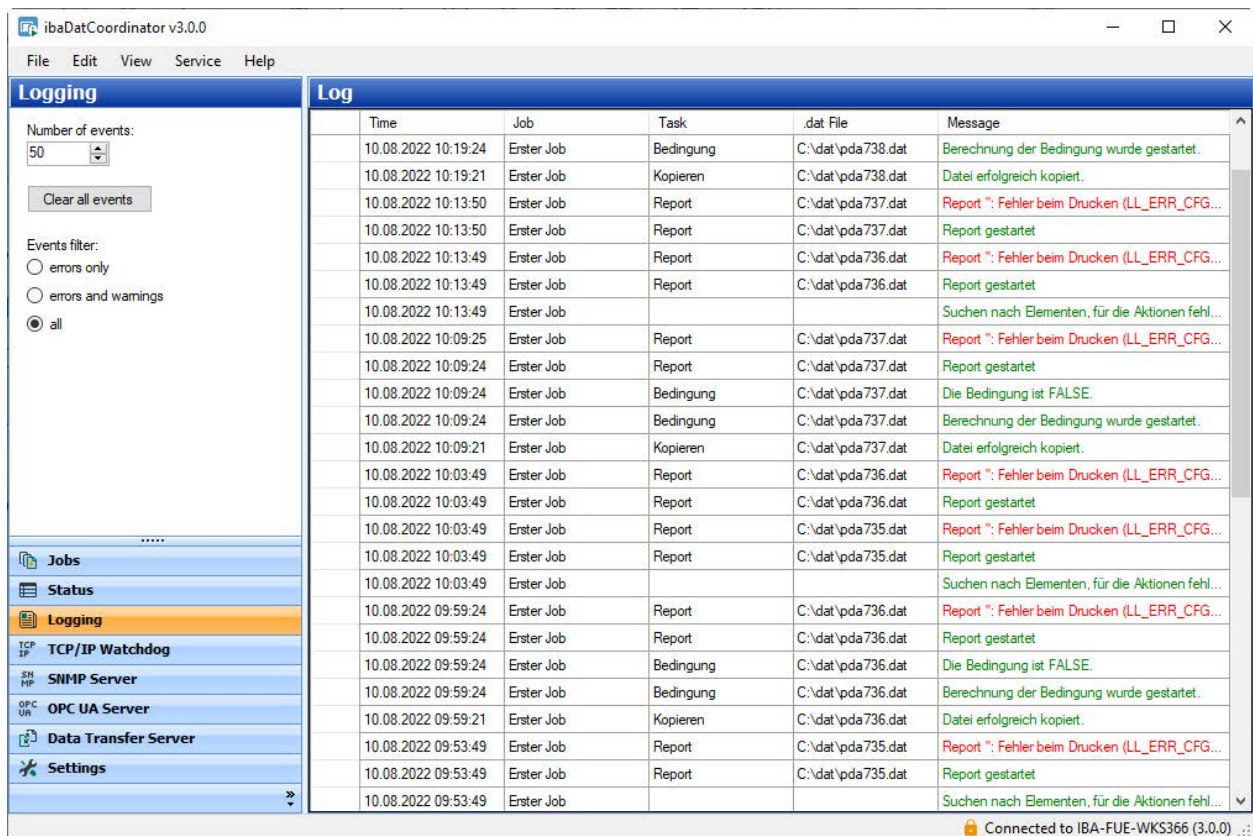
3.2.3 Logging

In the "Logging" view, the current logging messages generated during the job execution are displayed. The log messages contain warnings, information and errors.

While the jobs are being executed, *ibaDatCoordinator* generates a number of warning messages, information messages and error messages, as well as changes to the SNMP server status, that are all written to a log file.

The number of visible entries can be altered by changing the number of events on the left. Using the option fields, you can filter the events to show errors only, errors and warnings, or all events.

The <Clear all events> button deletes the list, but not the corresponding log file.



Each event shown contains the following information:

- A timestamp when the entry was made in the log file
- The job that *ibaDatCoordinator* has executed at the moment the entry was made (empty, if not applicable)
- The task that the job has executed (empty, if not applicable)
- The DAT file that was executed by the task (empty, if not applicable)
- The actual message. Each message type has a different color.
 - Green: Information
 - Orange: Warning
 - Red: Error

All columns of the status table can be sorted and filtered. If any filter is active on the table, an additional line containing the filter definition is shown at the bottom of the table.

If you right-click the table, a context menu opens with the following commands:

Open in ibaAnalyzer

The processed file can be opened with *ibaAnalyzer* for further analysis. Note that this option is only available if the file can be found in its original location.

Show full path

For all files shown in the table, the full path or only the filename can be shown.

Clear filters

If any filters are active on the table, they can be reset.

Log files

During the execution of a task, a log of the internal processes is created by *ibaDatCoordinator* and saved on the hard disk as log files.

The <Open log file> button opens the current log file with the standard text editor.

You can easily access the folder containing the log files from the start menu:

Start – iba – ibaDatCoordinator – Log files

If *ibaDatCoordinator* is installed as service, the log files will be stored in the following directory:

`C:\ProgramData\iba\ibaDatCoordinator`

If *ibaDatCoordinator* is installed as a program, the log files will be stored in the application data folder of the corresponding user:

`C:\User\username\AppData\Local\iba\ibaDatCoordinator`

The name of the current log file is `ibaDatCoordinatorLog.txt`. Previous log files have names with date and time included.

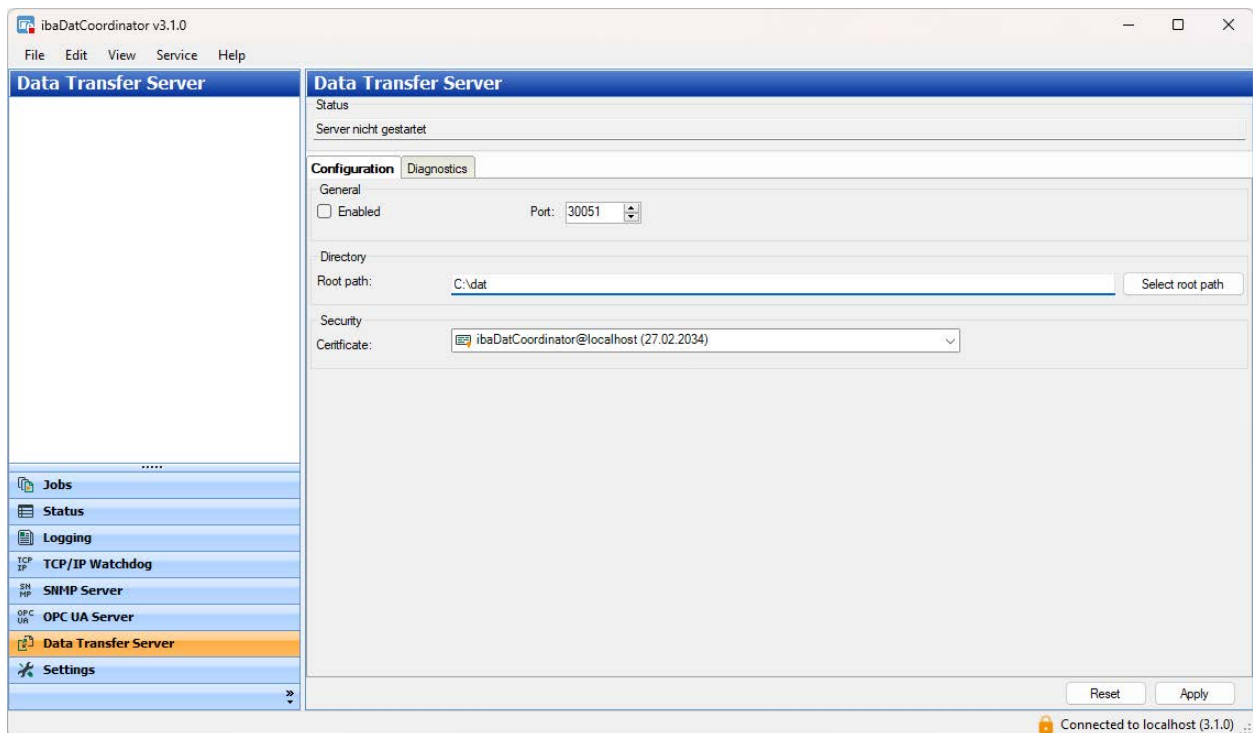
3.2.4 Data Transfer Server

The *Data Transfer Server* view can be used to configure *ibaDatCoordinator* to receive files from another *ibaDatCoordinator* instance.

Note



Only receiving files can be configured in this menu. To configure sending files to another *ibaDatCoordinator* instance see chapter ↗ *Transfer Data task*, page 135.



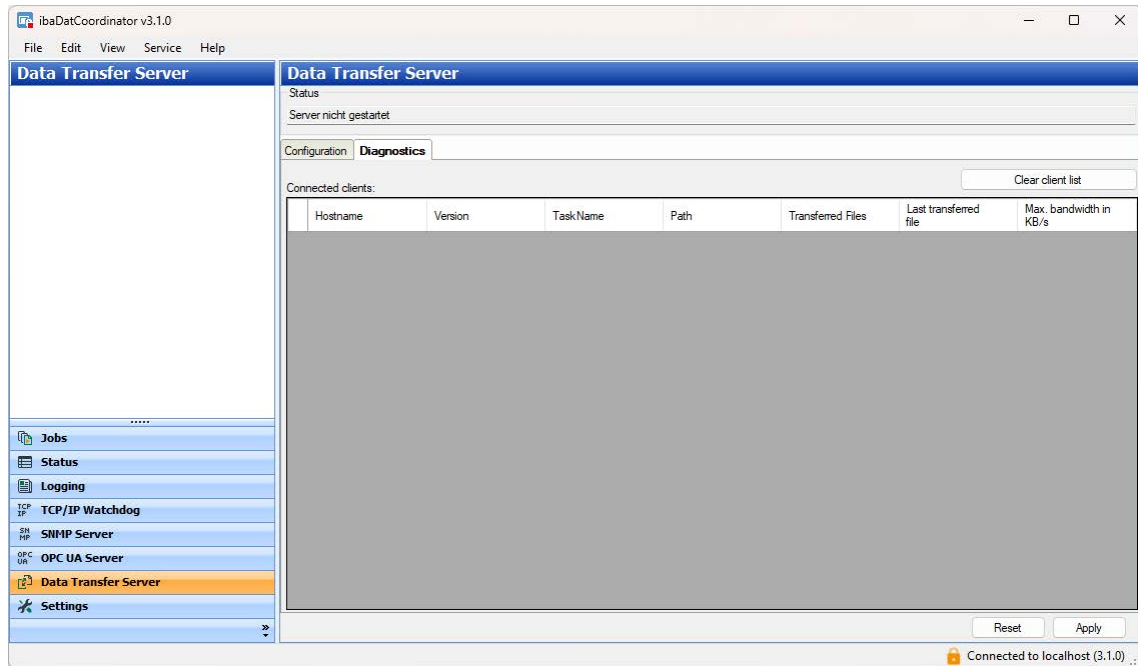
Configuration

1. The *Status* area shows whether the server has already been started or stopped.
2. You can enable or disable the *Data Transfer Server* by checking the *Enabled* checkbox in the *Configuration* tab under *General*. You can select any free TCP port for communication to this *ibaDatCoordinator* instance in the *Port* field. The default value is port 30051.
3. You can select the path where the sent files will be saved in the *Directory* section. The path setting from the respective sending *Transfer data* task (see chapter [Transfer Data task](#), page 135) is relative to the path specified here. The selection is possible both directly in the text field, and by clicking the <Select root path> button.
4. A certificate for the encryption of the connection can be selected in the *Security* section. 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 data* tasks (see chapter [Transfer Data task](#), page 135)!

Selecting the "Manage certificates" option will take you directly to the *Settings* tab. Existing certificates can also be imported or exported there in the certificate management. When exporting the certificate, it is not necessary to export the private key. See also chapter [Certificates](#), page 49.

5. The changes of settings are applied only with <Apply>. You can reset the settings to the default values with the <Reset> button.



The *Diagnostics* tab displays all incoming connections and corresponding information that have transferred data to the server since the last server startup. In addition to the host name and the *ibaDatCoordinator* version, the name of the task, the absolute destination path, the number of transferred files, the last transferred file and the configured maximum bandwidth are displayed.

Note

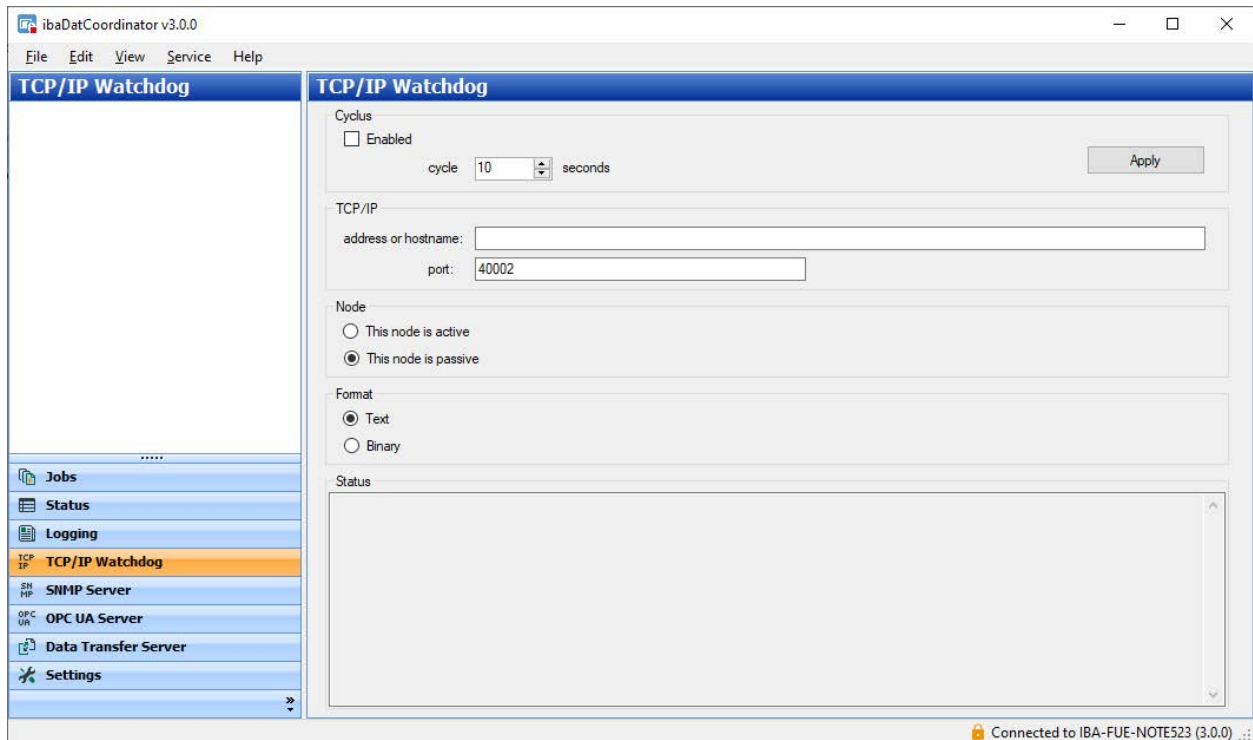


A value of "0" for the maximum bandwidth means "no restriction".

The <Clear client list> button can be used to reset the display.

3.3 Monitoring

3.3.1 TCP/IP Watchdog



In the "TCP/IP Watchdog" view you can configure a watchdog that sends a TCP/IP message to the external program at regular intervals. These messages can be monitored with appropriate applications.

These messages are sent regardless if a job is running or not. The watchdog message is sent via TCP/IP and a specified port. An active or passive connection can be established to the recipient of the watchdog message. In passive mode, several systems can receive the messages. The transmission cycle can be changed and 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 byte, comprising the information for 16 jobs. The format of the watchdog message is described in more detail in the Appendix.

In the status window, all connections from and to *ibaDatCoordinator* are displayed. A change in the configuration is only activated with <Apply>.

3.3.1.1 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 byte + 16 * 7 * 4 byte = 464 byte)

The table below contains the description for each integer value.

Structure of binary watchdog message:

	Byte	Description
	0	A message counter; every time a message is sent, this value is incremented by 1.
	4	The 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

	Byte	Description
	44	Job 2
	72	Job 3
	100	Job 4

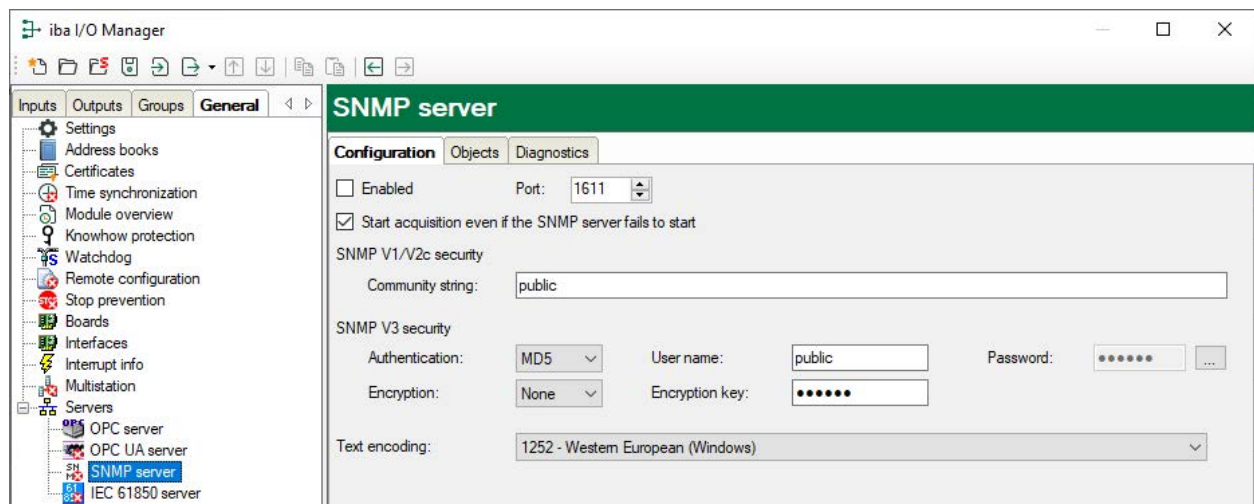
	436	Job 16

3.3.2 SNMP server

SNMP (Simple Network Management Protocol) was developed to monitor and control network elements (e.g. Router, 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.

3.3.2.1 Configuration tab



Enabled

You can enable or disable the SNMP server here.

Port

The designated port for the communication of the SNMP Server. The default value is 1611.

Community string

For the protocol versions, V1 and V2c, only a "community string" can be predefined for authentication; it acts as a password for the data traffic between the SNMP server and SNMP client. The default value for the community string is "public".

Authentication, user name and password

For the V3 protocol version, the safety standard is higher and there are multiple methods of authentication and encryption for selection. Two standards are supported for the authentication:

- MD5 (message-digest algorithm 5, hash function, 128 bit)
- SHA (secure hash algorithm, SHA1 class, hash function, 160 bit)

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

Encryption, key

For encryption, you can choose between:

- no encryption
- DES (data encryption standard)
- AES (advanced encryption standard)

Note

The best way to determine what settings should be used here is to consult your network administrator. Make sure that the configured port is not already used by other SNMP servers running on the computer.

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

Text encoding

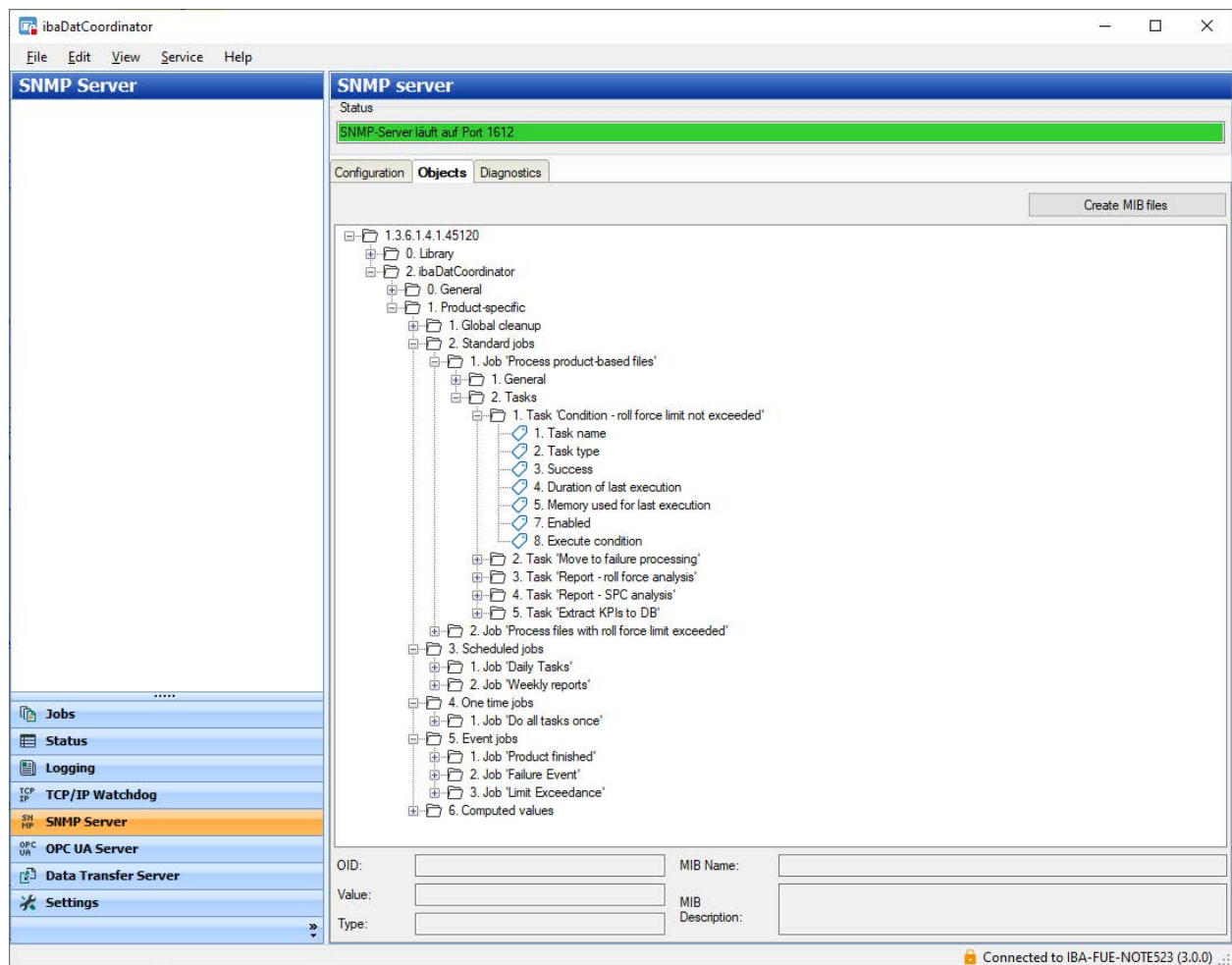
For a proper transmission of ASCII data (text channels) you can set here the appropriate text encoding regarding region and language.

Date and time format

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

- Binary (according to SNMPv2-TC DateAndTime, encoded in multiple bytes)
- String (YYYY-MM-dd HH:mm:ss:ffff)

3.3.2.2 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 license 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

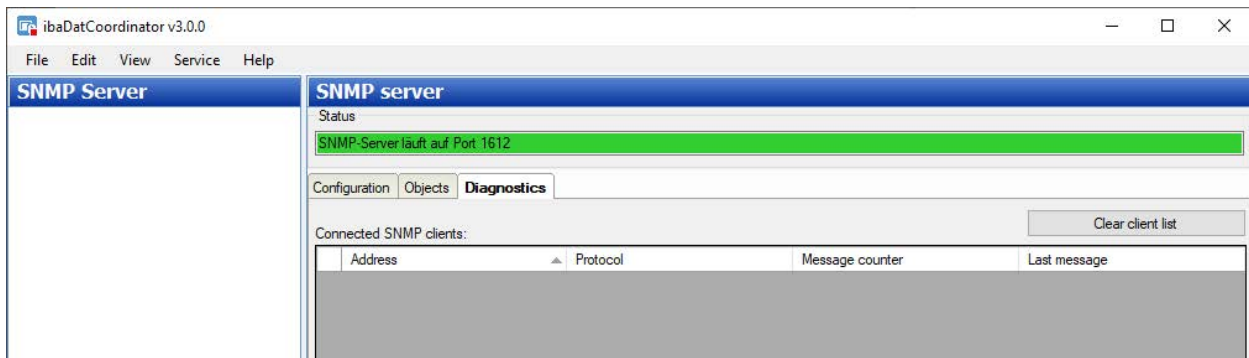
3.3.2.3 System-OIDs 1.3.6.1.2.1.1.

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"

3.3.2.4 Diagnostics tab



The diagnostics area contains information on the status of the SNMP server.

Connected SNMP clients

All clients which have accessed the SNMP server in the last hour are listed here. The clients that have accessed the SNMP server in the last 10 minutes are marked with a green background, the rest with a gray background.

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

3.3.3 OPC UA Server

In the "OPC UA Server" view, you can configure the OPC UA server integrated in *ibaDatCoordinator*.

Thus, status information from *ibaDatCoordinator* is provided for other systems. If an OPC UA Publish Task (license required) is additionally configured, calculated values are also published this way.

Note

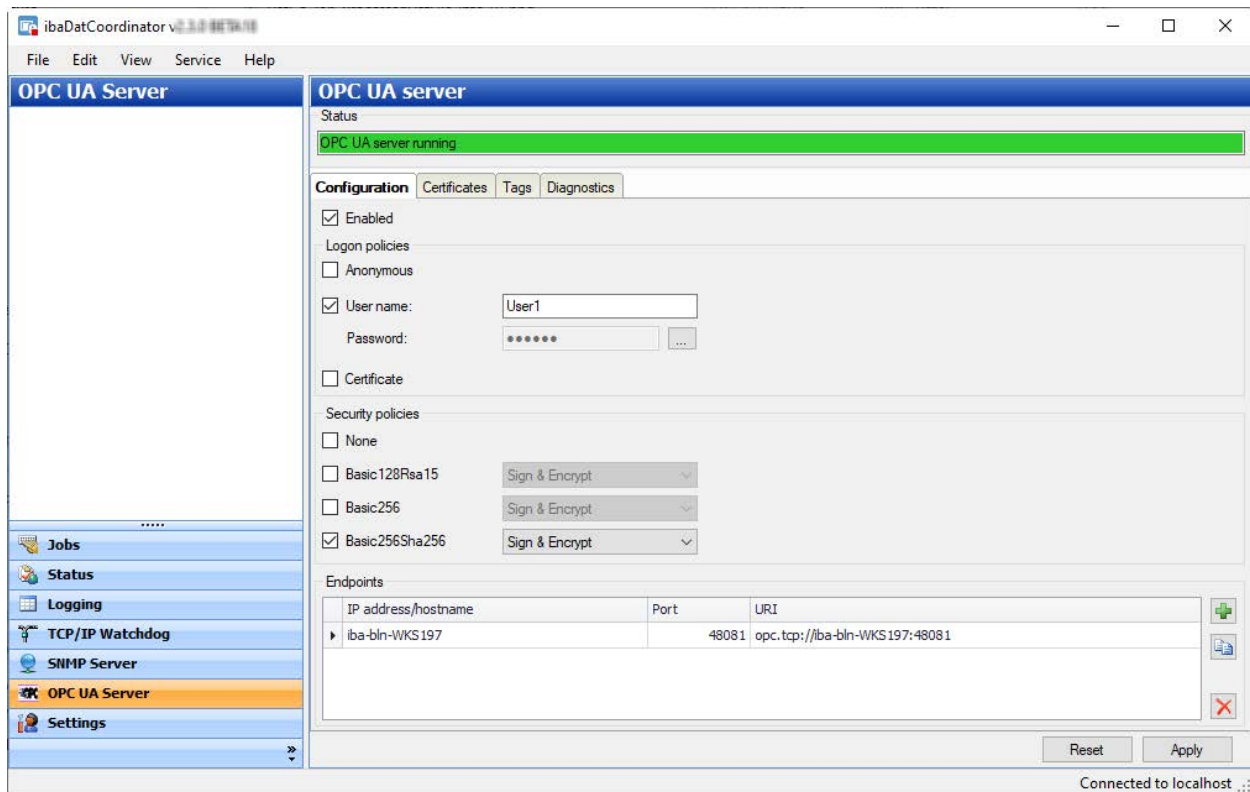


The basic function for status monitoring of *ibaDatCoordinator* via OPC UA Server can be used without additional license. For publishing values calculated with *ibaAnalyzer*, a license activation is necessary.

Before an OPC UA client can connect to *ibaDatCoordinator* (OPC UA server), an application certificate must be configured first. Certificates can be provided by the server as well as by the client. 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 interactively accepted or rejected by the user. Accepted certificates are automatically entered in the table in the *Settings* tab and marked as trusted. If the certificate is rejected, no communication will be established.

3.3.3.1 OPC UA server - Configuration



Enabled

Check this box to enable the OPC UA server function.

OPC UA server certificate

Select here from the drop-down list 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 the certificates is described in chapter [Certificates](#), page 49.

Login policies

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

Anonymous

If this option is enabled, clients can log in 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 user name and password as configured in this 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.

End points




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.

IP address/hostname	Port	URI
thisPC	48080	opc.tcp://thisPC:48080
thisPC	48081	opc.tcp://thisPC:48081
179.29.0.100	49000	opc.tcp://179.29.0.100:49000

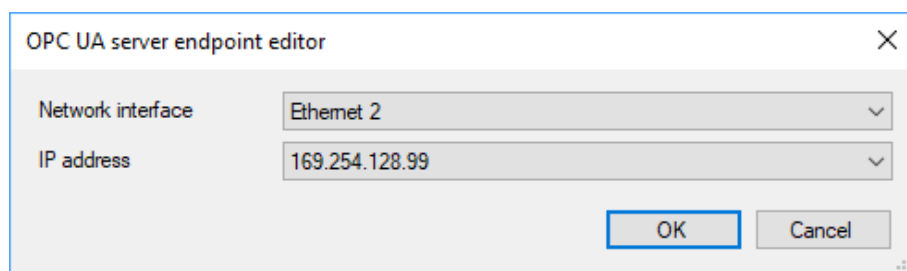
In the example above, OPC UA clients can connect to the OPC UA server from any network using port 48080 or 48081. In addition, clients can also connect to IP address 172.29.0.100 via port 49000.

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 this button to edit the endpoint.



Select the desired network interface via which the OPC UA clients should communicate. The drop-down 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.

3.3.3.2 OPC UA server - Tags

This tab displays the tags that can be published by the OPC UA server. These are basically:

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

For the currently marked tag there will be displayed under the tree with the tags:

- Node ID: quasi path 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

3.3.3.3 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.

The screenshot shows the 'Diagnostics' tab of the ibaDatCoordinator interface. At the top, there are three tabs: 'Configuration', 'Tags', and 'Diagnostics'. The 'Diagnostics' tab is active, showing a green status bar that reads 'OPC UA server running'. Below the status bar, there is a section for 'Connected OPC UA clients' with an 'Open log file' button. This section contains a table with three columns: 'Name', 'ID', and 'Last message time'. Below this, there is a section for 'Subscriptions' containing a table with four columns: 'ID', 'Monitored items count', 'Publishing interval', and 'Next sequence number'.

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

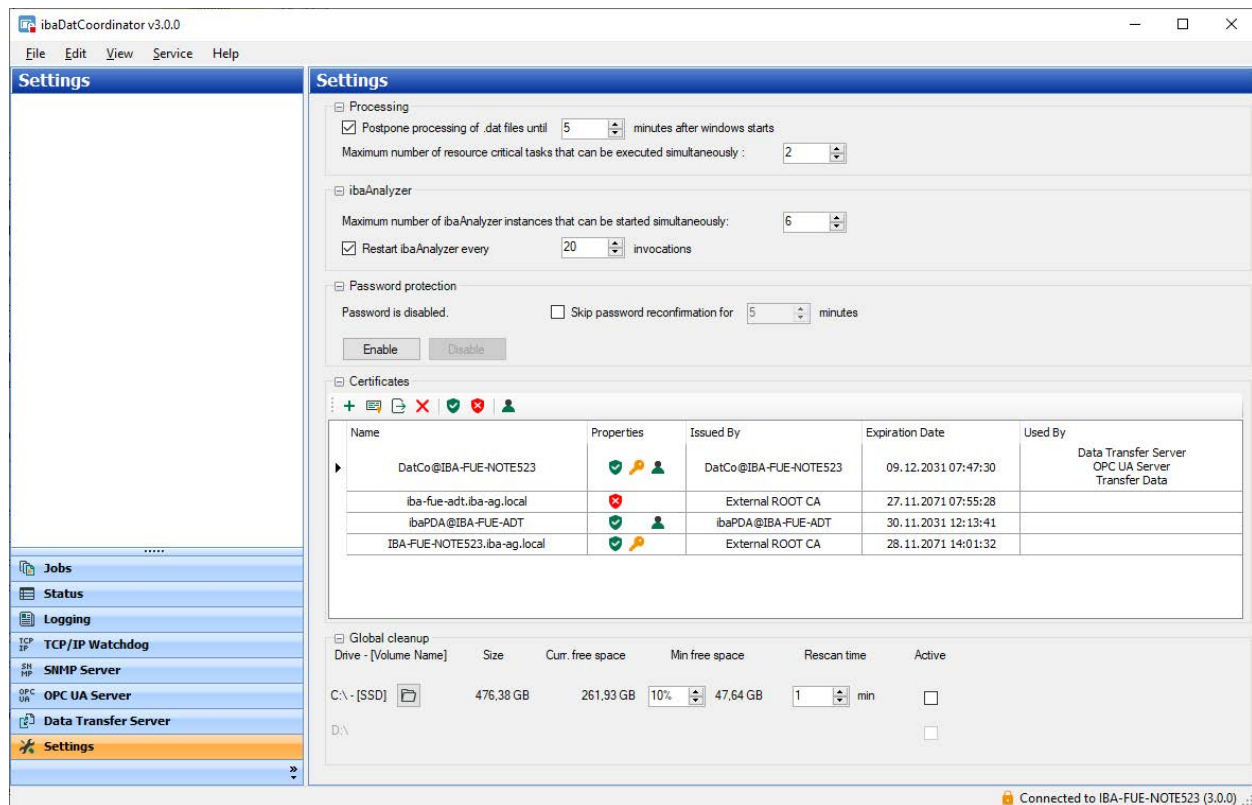
For each connection, the name and ID of the session are displayed along with the time stamp of the most recent communication.

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.

3.4 Settings

In the "Settings" view, you can adjust a number of general settings for *ibaDatCoordinator* and other functions.



3.4.1 dat file processing

Postpone processing of .dat-files until...

The option "Postpone processing of .dat-files until x minutes after windows starts" is used to make sure that Windows can start without disturbances and that all important applications and services are started before *ibaDatCoordinator* scans the DAT file directory for non-processed DAT files.

When Windows starts, all services are loaded first of all. When the *ibaDatCoordinator* service starts, *ibaDatCoordinator* waits for the time interval that has been specified in the settings before the program starts searching for non-processed DAT files.

This can prevent connection problems with services that have not been started yet. Such important applications include for example databases, network connections or printer drivers.

Maximum number of resource critical tasks....

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 e.g. 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 several tasks in parallel, you should click on <Optimize system resources> in the "Service" section or in the *ibaDatCoordinator* status application to be on the safe side.

3.4.2 ibaAnalyzer

Executable

This displays the path to the `ibaAnalyzer.exe` application (read-only). When you click on the button with the *ibaAnalyzer* symbol, *ibaAnalyzer* is started. Hence, you can verify your setting.

Note



This row is only available if *ibaDatCoordinator* was installed as a program. In case of a client-server installation, this option is located in the *ibaDatCoordinator* status application.

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. Default setting: 5.

If the maximum permissible number of instances has been reached in operation, other tasks that need *ibaAnalyzer* to be processed have to wait.

Restart ibaAnalyzer every ... invocations

This option can be activated if *ibaAnalyzer* is to be restarted after processing a fixed number of DAT files.

3.4.3 Password protection

The password protection for *ibaDatCoordinator* provides protection against unintended and unauthorized actions.

For a convenient way of working after you have authorized for the first time, you can select the option "Skip password reconfirmation for...minutes" and set the waiting time for a new password query. This suppresses a new password query within the set time when executing protected functions. 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 <Change> in the password protection area of the dialog.
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 of the dialog.
3. Enter the current password.
4. Click <OK>.

3.4.4 Global cleanup

The "Global cleanup" function is used to make sure that there is a minimum free storage space on the local drives of a computer. When using the global cleanup function, the following rules must be observed:

- Only data files (DAT files) are cleaned up by this function.
- The cleanup function only works with local drives, incl. external drives and USB media on the respective computer. Network drives are not cleaned up.
- The cleanup function has to be enabled individually for each drive. To enable the global cleanup for a drive, click on the check box "Active" (checkmark).
- To use the cleanup function on the system disk or partition (C:\), a subdirectory has to be selected that is to be subject to the cleanup function.
- The minimum storage space that has to be free is set for each drive as percentage. 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.
- If the minimum has been reached, the oldest data files are deleted on the drive, no matter which subdirectory they are saved in.
- It is checked periodically in configurable time intervals (rescan duration) if the minimum has been reached.
The minimum rescan duration is 1 min.
Pay attention to the "growth rate" of the storage space for data files and the settings for the minimum free storage space in order to choose the correct rescan duration.
- To clean up other file types, use a cleanup task in which you can specify the desired file types (file extension).

Note



In order to prevent overlapping cleanup actions, disable the task-specific cleanup strategies if you want to use the global cleanup.

3.4.5 Certificates

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

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

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

Certificates can 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 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".

3.4.5.1 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
▶ DigiCert Global Root G2	✓	DigiCert Global Root G2	15.01.2038 13:00:00	
ibaPDA@MyComputer01	✓ 🔑 👤	ibaPDA@MyComputer01	13.04.2032 16:23:23	
ibaPDA@Workstation01	✓ 🔑	ibaPDA@Workstation01	22.06.2032 12:55:20	
ibaPDA0003	✓ 🔑	ibaPDA0003	22.06.2032 12:56:26	OPC UA Server

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.

Certificates				
Name	Properties	Issued By	Expiration Date	Used By
▶ DigiCert Global Root G2	✓			
ibaPDA@MyComputer01	✓ 🔑 👤	ibaPDA@MyComput		
ibaPDA@Workstation01	✓ 🔑	ibaPDA@Workstati		
ibaPDA0003	✓ 🔑	ibaPDA0003		OPC UA Server

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 *ibaPDA003@D* is used by the OPC UA server in *ibaPDA*. This means that this certificate was selected during configuration of the OPC UA server.

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

3.4.5.2 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 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.

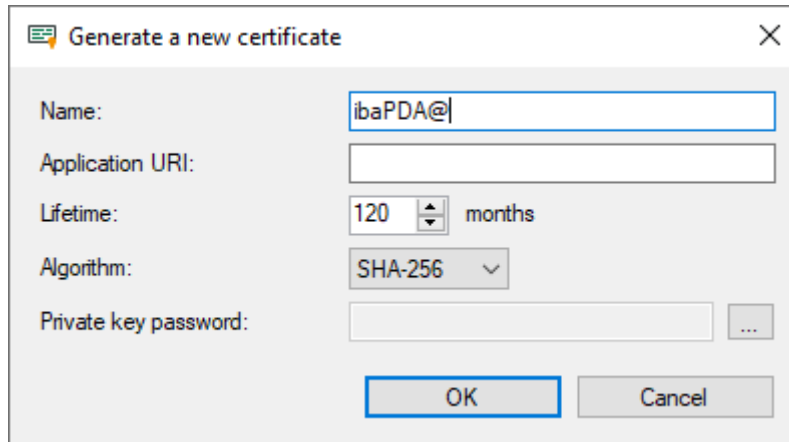
Table 2: Buttons in the toolbar for certificate management

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.

3.4.5.2.1 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).


3.4.5.2.2 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.

3.4.5.2.3 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.

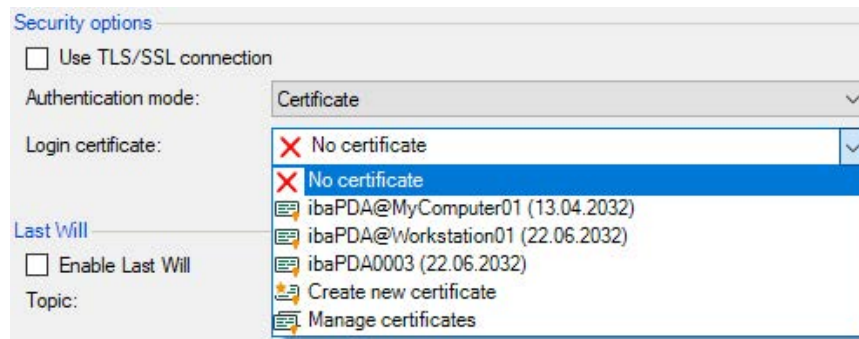
If the password is incorrect, the certificate will not be exported.

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.

3.4.5.3 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.

3.4.5.4 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.

3.4.6 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

Database type

Select the database type used from the drop-down list.

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-DB* manual.

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 Processing data - ibaDatCoordinator Jobs

If you want to configure *ibaDatCoordinator*, you have to add jobs to the *ibaDatCoordinator* project.

To create and configure a job, please decide first which job type you need.

Click the corresponding blue node in the tree structure

- Add .dat File Triggered Job
- Add Scheduled Job
- Add Event Job
- Add Time period Job
- Add One Time Job
- Add External File Triggered Job

Click on the newly added job node.

The basic settings for this job are shown in the display and configuration area.

The configuration of the basic settings for the different job types are described in the following chapters.

The different jobs that were created in the "Job" window run in parallel and can contain multiple tasks that run serially.

The job nodes can be moved and added 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.

If you click with the right mouse button on the job tree, a context menu is opened. In this menu, you will find commands for deleting, cutting, copying and pasting jobs and for adding a task to the currently selected job. Some of the commands are only available if the job is not currently running.

The task nodes can also be moved via drag & drop. If you drag a task from one job to another or within one job, the tasks will be executed in a different sequence. If you execute drag & drop and keep at the same time the <Ctrl> key pressed, a clone of the moved job or task node is created.

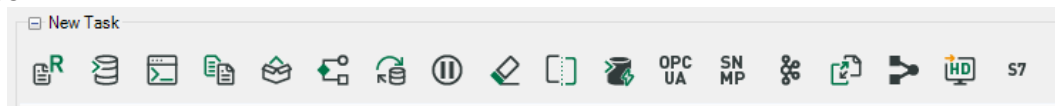
4.1 General settings

All job types have some settings in common, which are described below. The job type specific settings are described in the corresponding chapters.

ID

In this part of the configuration, a name for the job can be specified. The default name is "New Job". To make things easier, we recommend setting a name which is appropriate for its usage.

New task



In this part of the configuration, you can add a task to the job. A job can contain multiple tasks and will process them in sequential order. The tasks are each represented by a symbol.

	Report task
	Extract task
	Skript task
	Copy task
	Upload task
	Condition task
	Update data task
	Pause task
	Cleanup task
	Synchronization task
	Splitter task
	HD offline event task
	OPC UA Publish task
	SNMP Publish task
	Kafka Publish task
	Merge task
	ibaHD import task
	S7 writer task


Note

An exception is the external file triggered job. Here each job has only one fixed task that converts external files into a DAT file.

The resulting DAT files can then be processed with a dat file triggered job as usual.

Notifications

In this part of the job configuration, you can define whether notifications should be sent when the respective job is executed.

Messages can be sent by e-mail. The configuration can be tested by sending a test message with the  button.

If you want to send an e-mail to a recipient, you have to enter the e-mail address and the smtp-server as well as – depending on the settings of your e-mail-account – an additional user name and password. You can also enter a sender address to inform the recipient about the origin of the e-mail.

Note

If you want to enter a sender address, you need to adhere to a certain format, even if the address is just made up and does not really exist.

E-mail address, e.g. `ibaDatCoordinator@iba-ag.com`, or

Name and e-mail-address in pointed brackets, e.g.

`ibaDatCoordinator <noreply@iba-ag.com>`

If you do not enter an e-mail address, the recipient will see the default address *ibaDatCoordinator <noreply@iba-ag.com>*.

4.2 dat-file triggered job

With this kind of job, *ibaDatCoordinator* monitors one or more folders where DAT files are stored. By means of a file system monitoring or by scanning the folders in regular intervals *ibaDatCoordinator* recognizes when new files have been added. As soon as *ibaDatCoordinator* detects a new DAT file that has not been processed yet, the job begins processing the defined tasks.

ibaDatCoordinator monitors with each execution the status of the tasks for each job and tags the processed DAT files. Info fields are added to the DAT files to show whether a job has been completely processed (processed) or failed (failed) and how many attempts have already been made to process the job.

A job has failed if at least one of its tasks could not be executed. As long as a job could not be processed successfully, *ibaDatCoordinator* tries to repeat the execution.

You can define a time interval in the job configuration in which the DAT files are processed again when the processing has failed. Furthermore, the number of reprocessing attempts can be limited. If this number of the reprocessing attempts is exceeded, these files are marked as definitively failed. You decide whether to delete these files or reset the inserted status information for reprocessing.

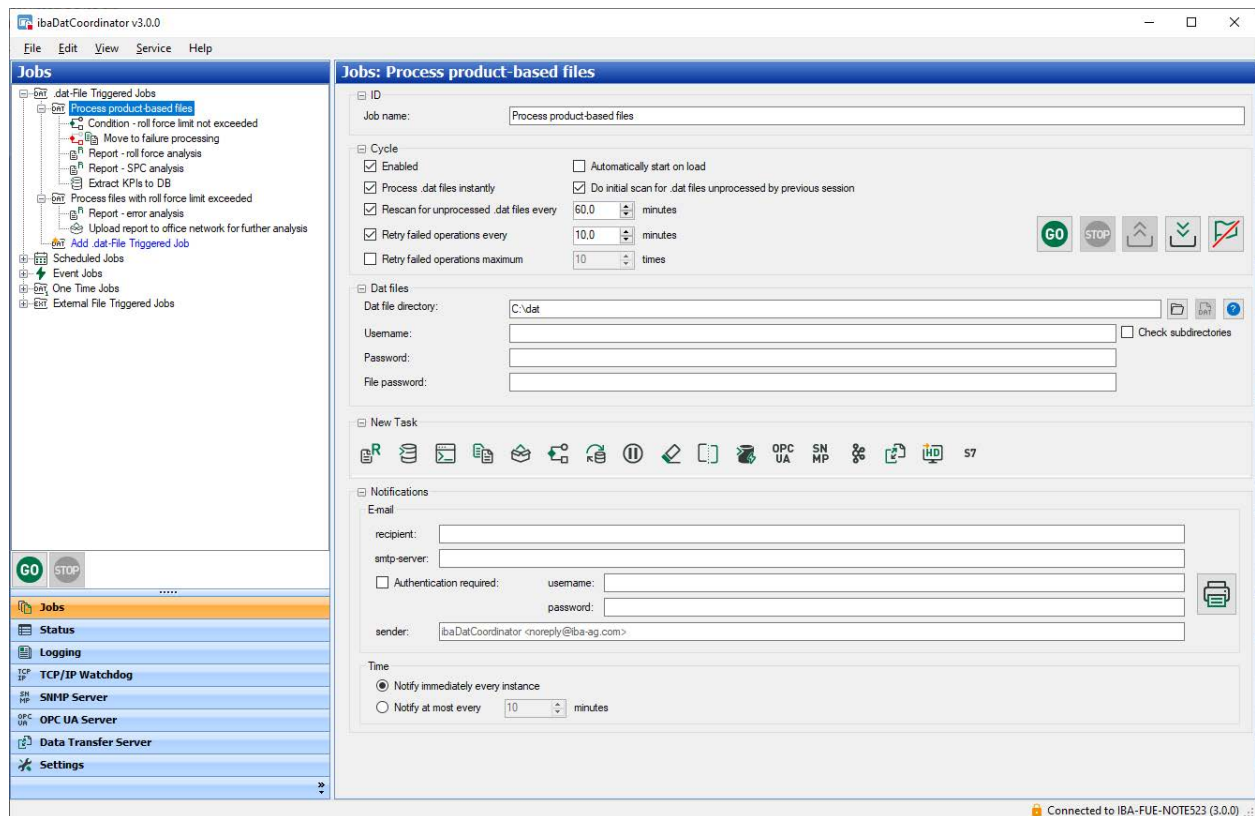
The information added by *ibaDatCoordinator* in the DAT file can easily be found in *ibaAnalyzer* in the info node of the dat file: \$DATCOOR_status and \$DATCOOR_times_tried.

Note



If a job's execution must be repeated, all of its tasks will be reprocessed again, including those tasks which have been processed successfully before.

To add a new dat file triggered job, click on the corresponding button in the job and task configuration tree.



In addition to the default settings for each job, the cycle (when and how often to scan for new files) and the directory to monitor must be configured for the .dat file triggered job.

Cycle

In the *cycle* subcategory, you can configure the parameters for the job timing. The minimum time is 0.1 minutes (6 seconds). The following options can be selected or deselected:

Enabled

If this option is deselected, the job is disabled entirely.

Automatically start on load

If you choose this option, the job starts automatically after the XML-configuration file has been loaded. This can be done either manually or with starting the application or the *ibaDatCoordinator* service.

Process .dat files instantly

If this option is enabled when *ibaPDA* closes a DAT file, then *ibaDatCoordinator* instantly starts processing the closed DAT file.

If this option is not enabled, then new or unprocessed files can only be searched for and found by the so-called "Rescan" (see below) at regular intervals.

If DAT files are saved in the network, *ibaDatCoordinator* might miss newly created DAT files when the network is not active. In this case, we recommend 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 with *ibaDatCoordinator*, the "Process .dat files instantly" option must be disabled!

Do initial scan for .dat files unprocessed by previous session

If *ibaDatCoordinator* is started and this option is enabled, the DAT file directory is scanned for unprocessed DAT files.

If this option is not enabled, then new or unprocessed files can only be searched for and found by the so-called "Rescan" (see below) at regular intervals.

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

With this option, you can determine the time *ibaDatCoordinator* scans the directory for unprocessed DAT files. If you disable this option, the rescan function is deactivated. As a result, no DAT files will be processed that are not picked up by the "process dat files instantly" option.

Tip

If you only want to process the current DAT files, e.g. to generate a report, the rescan option should not be enabled. We recommend enabling this option, in case you want to process all DAT files.

Retry failed operations every...minutes

Here you can set the time after which the failed tasks on DAT files that have been marked as failed, are reprocessed.

ibaDatCoordinator periodically scans the directory to perform the configured tasks with files marked as failed.






If the option is disabled, only one attempt is made to process a file at a time and no directory scan is 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 "Permanently failed dat files".

If this option is not enabled, *ibaDatCoordinator* will not stop trying to process the failed files if the option "Repeat failed operations every ... minutes" is enabled.

Control elements:

	Starts the current job (only this one).
	Stops the current job (only this one).
	While the job is running, configuration changes are only applied if a manual upload is performed.
	To undo the configuration changes since the last time the job was started, you can download and apply the currently running configuration.
	Removes all markings from the DAT files that they have already been processed. After the marking has been removed, <i>ibaDatCoordinator</i> treats these files as newly created files.

Dat files

Here you can determine the source directory for the DAT files. You can enter the name of the directory or select the directory by means of the browser button . If you want to access a network drive, you can enter here the network path in UNC notation with user name and password. By clicking the button with the question mark you can check if the specified source directory can be accessed.

Note



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

Note



If you want to use network shares, we recommend entering the UNC path description (\\Host\Network share or folder) for addressing the file directory.

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

\\<IP-addr. or hostname of ibaPADU-S-IT>\RamDisk

(and optionally "\dat",

if it has been configured in the DAT_FILE_WRITE function module of the *ibaLogic* project with the default directory path = C:\dat (ibaLogic-V4_v4.2.4 and higher))

The user name is 'padu', the default password is '1234'.

(You can change the password via the web interface of *ibaPADU-S-IT*.)

We also recommend defining a time period for the beginning of the dat file processing after booting (for additional information, see chapter [➤ Settings](#), page 44).

If you want to monitor the subdirectories of the source directory, select the option "Check sub-directories". This is required if the DAT files are stored in subdirectories, that are separated e.g. by year, month, week, day and/or hour.

File password

In order to process password protected DAT files by *ibaDatCoordinator* you should enter the correct DAT file password here. The password will be stored in the job automatically.

4.3 Scheduled job

The type of the scheduled job has been developed especially for the access to data recordings that were generated by *ibaHD-Server*. However, scheduled jobs can be configured even without an HD query, only with the common tasks. For a scheduled job, *ibaDatCoordinator* executes the defined tasks in a configurable cycle. The measurement data for the job (there is no data file in the conventional sense) are retrieved from the HD store cyclically in a time interval that can be configured. For this purpose, *ibaDatCoordinator* generates an HD query (hdq file) based on the settings for the HD store and the time interval. This hdq file is then transferred to *ibaAnalyzer* for processing the tasks, whereupon the actual data query is then performed. The HD query can be handled and processed in *ibaAnalyzer* like a common data file. The time interval for which the data is queried is relative and always refers to the time when the task is executed (trigger).

Other tasks that are to be performed cyclically but without measurement data, e.g. cleanup tasks or script tasks, do not require an HD query. In such a case, do not configure the HD server connection.

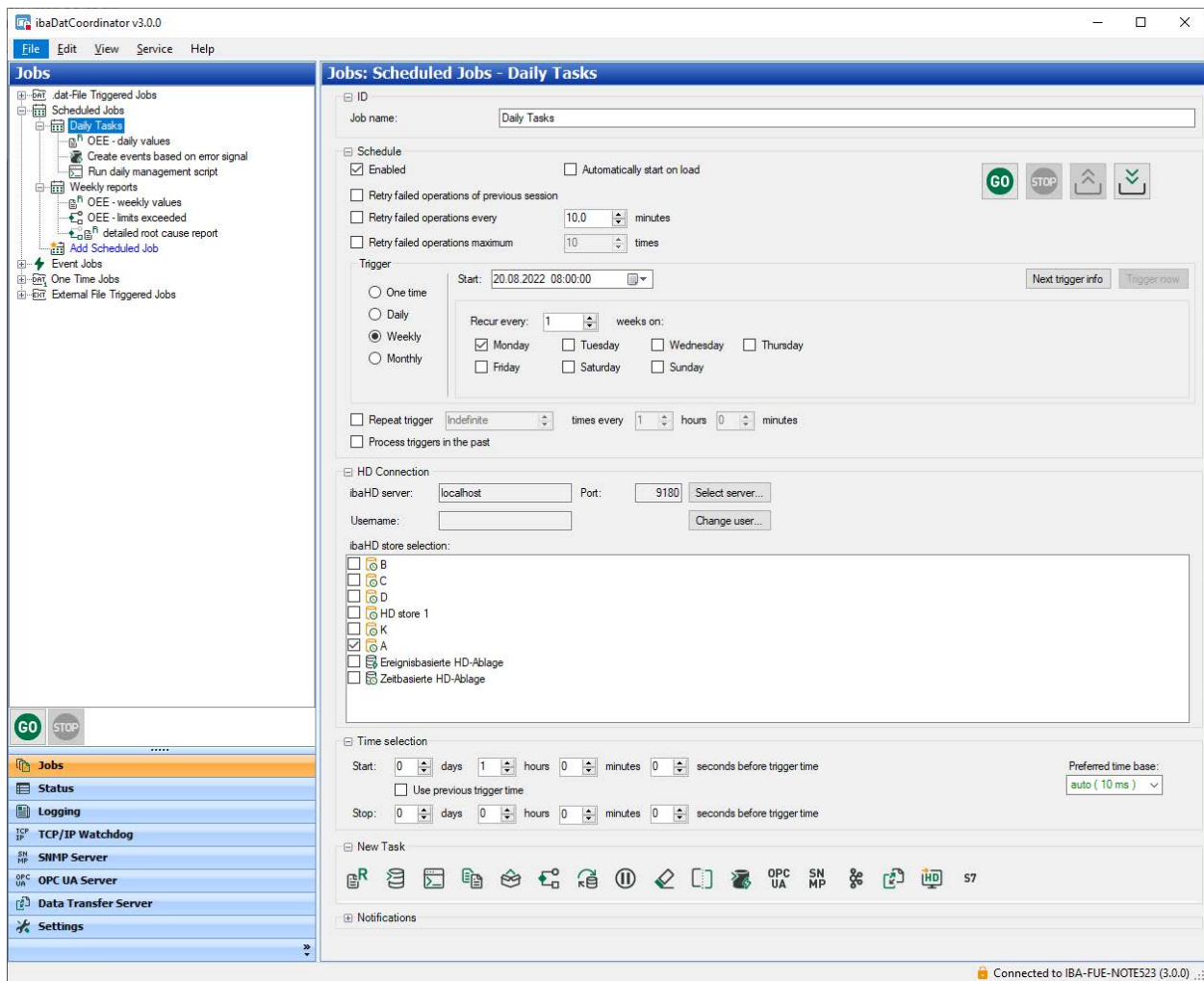
Even for scheduled jobs, *ibaDatCoordinator* monitors the status of the processing and can repeat failed actions in configured time intervals or for a configured number of times.

In addition to the default settings for each job, the processing schedule, HD connection and query period must be configured for scheduled jobs.

Note



It is also possible to execute tasks cyclically without using a HD data query. In this case, *ibaDatCoordinator* behaves like a simple scheduler.



Schedule

In the "Schedule" area, you can do the settings for the starting time and the period of execution.

Enabled

If this option is deselected, the job is disabled entirely.

Automatically start on load

If you choose this option, the job starts automatically after the XML-configuration file has been loaded. A configuration file is loaded as follows:

- Via the "File" menu
- If *ibaDatCoordinator* has been installed as a service, the XML file is loaded when the service is started.
- If *ibaDatCoordinator* has been installed as standalone version, the last stored session is loaded as soon as *ibaDatCoordinator* is started.

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 make sure that operations that could not have been completed in the previous session, are to be retried.





Retry failed operations every...minutes

Here, you can define the time after which the failed tasks with the HD queries that are marked as failed tasks are retried. *ibaDatCoordinator* will only do those tasks with the HD queries that have been marked as failed.

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 "Permanently failed dat-files". If this option is not enabled, *ibaDatCoordinator* will not stop trying to process the failed HD queries.

Control elements:

	Starts the current job (only this one).
	Stops the current job (only this one).
	While the job is running, configuration changes are only applied, if a manual upload is performed.
	To undo the configuration changes since the last start of the job, you can download the currently executed configuration and apply it.

Trigger

With the trigger settings, you can define when *ibaDatCoordinator* will execute a job, and hence process the configured tasks.


Start

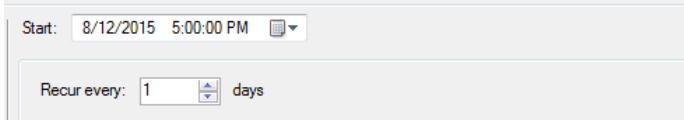
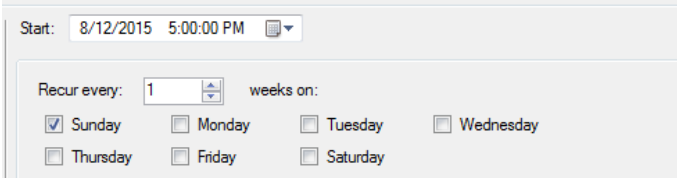
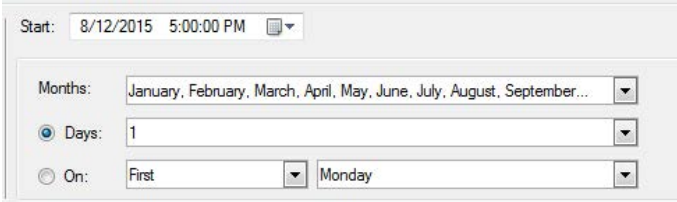
Enter here the first starting time for the job execution. You can use the calendar to select a date.

You have to enter the time manually in HH:MM:SS format.

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, the following settings are available:

One time	<div> Start: 8/12/2015 5:00:00 PM  </div>
----------	--

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

Repeat trigger

With this option, you can schedule the job execution cycle up to hours or minutes. If this option is enabled, the job is executed in the schedule you have defined here in addition to the rough plan (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 fired once more in the set interval.
- Number of repetitions = 2: After the first trigger, the trigger will be fired twice, each time at the set interval.
- More values corresponding
- Number of repetitions = unlimited: After the first trigger, the trigger will be fired again and again at the set interval until a new (main) trigger is fired.

The latter option is helpful if, for example, you want a task to be executed on a daily basis every 10 minutes over 24 hours. In this case, you do not have to calculate the number of repetitions (1440).

With the "Repeat trigger" option, you can have a job executed several times a day, (e.g. for creating shift reports). Select the daily execution, beginning at 6 a.m. with a trigger repetition 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.

<Next trigger info>

This button opens an info window in which the next 10 calculated trigger times are listed 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>).

HD connection

In the "HD connection" area, you can adjust 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.

Click on the <Select server...> button. In the selection dialog click on the desired *ibaHD-Server* in the table in order to connect *ibaDatCoordinator* with the server. Close the dialog with <OK>.

If the user management is activated on the *ibaHD-Server* in question you must then enter or choose the username and enter the correct password.

For more information about ibaHD connection and user management see [↗ ibaHD server connection and user management](#), page 144

Then, choose the desired ibaHD store in the ibaHD store selection pane, where the measurement data are stored. If you select more than one store, the query criteria will be applied to all selected stores. All data in the query result will be subject to processing by the job.

Note



If you want to have tasks executed cyclically, which do not require measurement data (e.g. cleanup tasks or script tasks) just leave the "server" field in the "HD connection" area empty or change the setting to "Not connected".

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

In the "Start" section, you must define the beginning of the query time period, in the "Stop" section, the end of that period. "Start" always lies further in the past.

Both entries refer to the trigger time.

Use previous trigger time

If you enable this option, the time of the last trigger is always used as starting time for the query period. This way, you can reliably make a gapless query, provided that in the "Stop" section only zeros are set (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 configuring ibaHD queries in *ibaAnalyzer* you can choose your preferred time base here. Depending on the time range settings and the number of contained samples in the HD query, the time bases which are offered for selection may vary.

In case you do not want to have a specific time base for the query results, just keep the "auto" setting. This ensures that the number of samples in the query result provide for maximum resolution, which may be very important for further calculations.

Reference

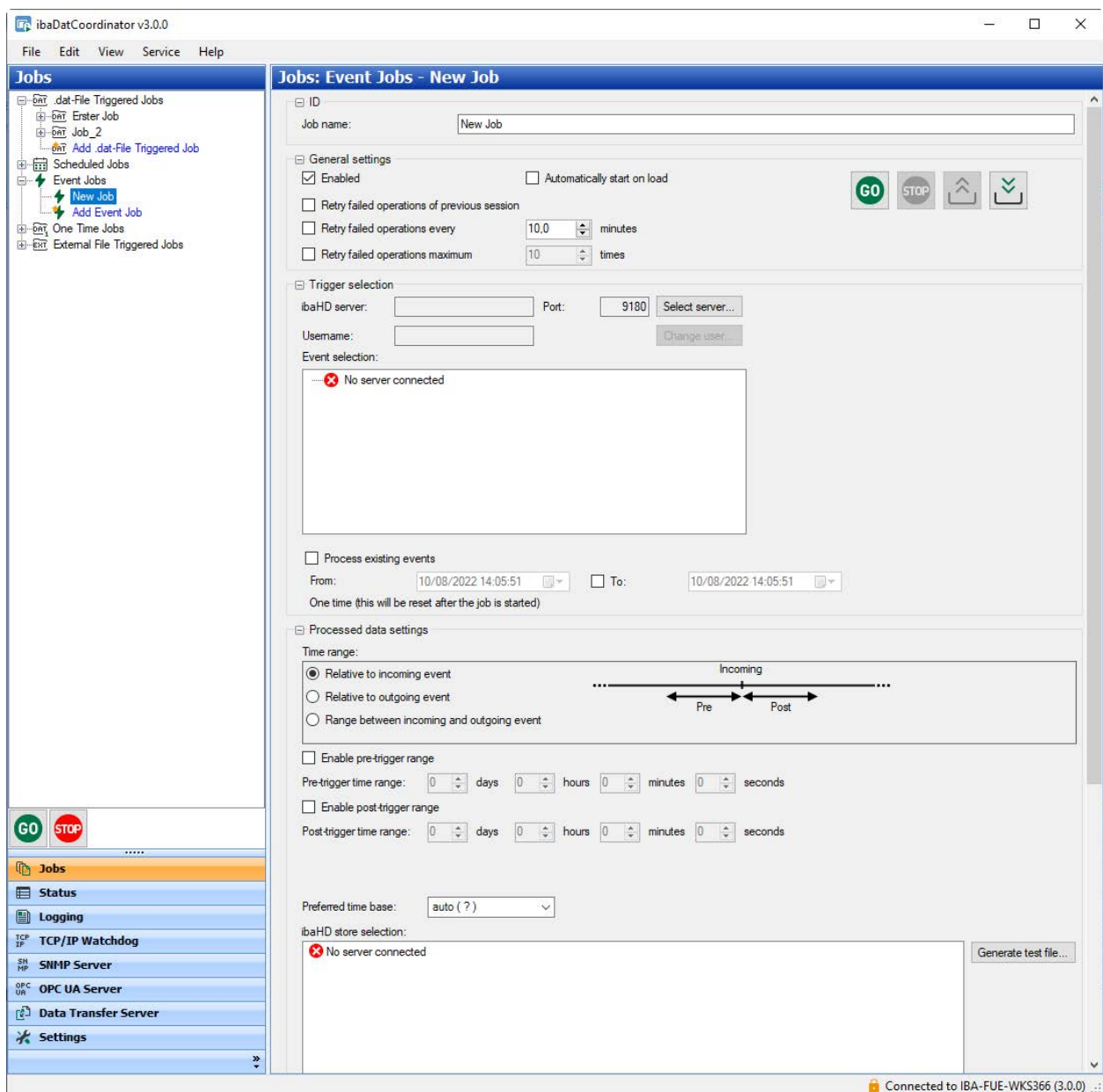
For more information about functioning, configuration and query of an HD server, please see the *ibaHD-Server* manual.

4.4 Event jobs

An event job is a job, which is executed upon the occurrence of a certain event. You can only use events, which have been configured as events in an eventbased data storage on an *ibaHD-Server*. *ibaDatCoordinator* requires access to the *ibaHD-Server*. To detect the occurrence of the events, each job periodically queries the *ibaHD-Server*.

In order to configure and use an event job *ibaDatCoordinator* service requires access to an *ibaHD-Server* with an event-based data store.

Event jobs are executed when certain defined events occur. In order to detect the occurrence of these events, *ibaDatCoordinator* periodically queries an *ibaHD-Server*.



For the settings in the categories "ID", "New task" and "Notifications", the explanations in the chapter "dat-file triggered job" apply. The settings in the category "General settings" correspond to those of the category "Cycle" of the dat-file triggered job.

The specific event job settings are:

Trigger selection

ibaHD server

Click on the <Select server ...> button. Click on the desired ibaHD server in the table to connect *ibaDatCoordinator* to it. Close the dialog with <OK>.

If the user management is enabled on the ibaHD server you should then enter or select a user name and enter the correct password.

If the connection to an ibaHD server has been established, the event-based stores and the available events are displayed in the "Event selection" pane.

You must select (checkmark) at least one event to get a valid configuration. Each occurrence of a selected event will trigger the job execution.

Processed data settings

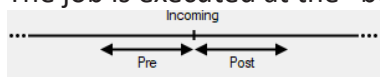
In this section you specify the source for the data processing in detail, i.e. the ibaHD store with the desired data, trigger behaviour and time range.

Time range

Determine, when exactly the job should be executed on an occurrence of an event. Choose one of three options:

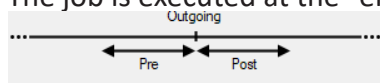
■ Relative to incoming event:

The job is executed at the "beginning" of a selected event (e.g. rising edge).



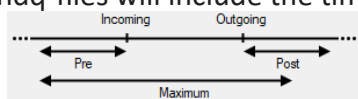
■ Relative to outgoing event:

The job is executed at the "end" of a selected event (e.g. falling edge).



■ Range between incoming and outgoing event:

This option is only useful in case the job should create hdq-files. When using this option, the hdq-files will include the time period between incoming and outgoing event.



Similar to the conditional HD queries in *ibaAnalyzer* you can set pre- and post-trigger ranges here as well. The time ranges are relative to the monitored event occurrences.

Enable pre-trigger range

Enable this option if you want to include the time span prior to an event occurrence. The size of the time range can be set right below. When executing the job, the data of this time range up to the specified event occurrence (incoming or outgoing) is in the query result.

Enable post-trigger range

Enable this option if you want to include the time span after an event occurrence. The size of the time range can be set right below. When executing the job, the data of this time range starting from the specified event occurrence (incoming or outgoing) is in the query result.

Maximum time range

The setting for this time range is only visible when you have selected the third option (Range between incoming and outgoing event).

The maximum time range serves as a fall back in case the incoming event already passed by but the outgoing event has not occurred before the maximum time range expired. Note, that the maximum time range includes the pre- and post-trigger time ranges.

Preferred time base

Similar to configuring ibaHD queries in *ibaAnalyzer* you can choose your preferred time base here. Depending on the time range settings and the number of contained samples in the HD query, the time bases which are offered for selection may vary.

In case you do not want to have a specific time base for the query results, just keep the "auto" setting. This ensures that the number of samples in the query result provides the maximum resolution, which may be very important for further calculations.

ibaHD store selection

Select the HD stores whose data should be queried. If available, you may select more than one store.

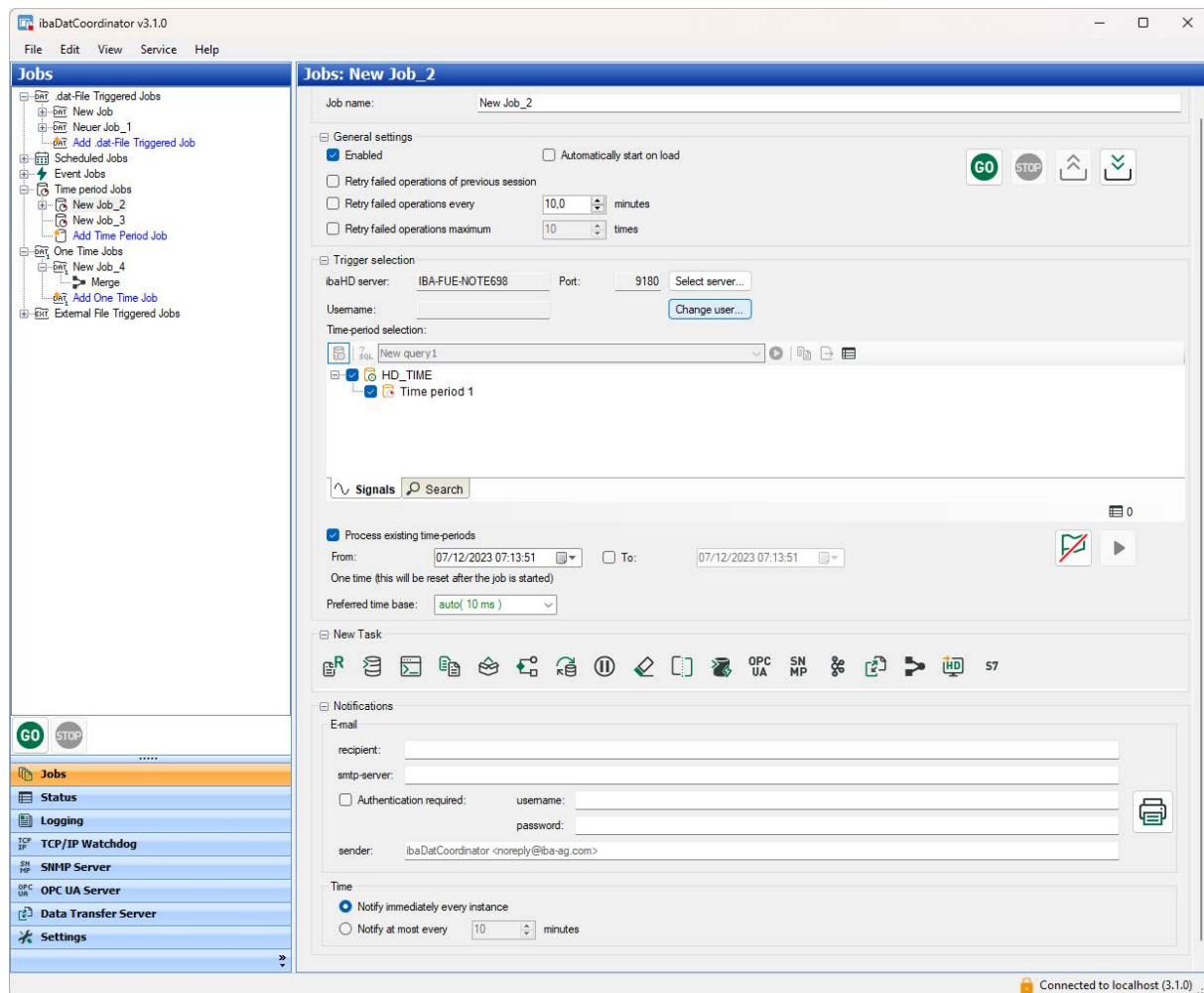
For testing purposes you can generate an hdq-file by clicking on the <Generate test file..> button. For generating such a file, a random occurrence of one of the selected events will be used or, if no such occurrence could be found, the current time.

After completion of these settings, you can add the requested tasks to the job. The measured data, which are included in the hdq-file will then be subject to the tasks.

4.5 Time period job

Time periods mark a range of time inside a timebased ibaHD store. Time periods are grouped in time period stores. A timebased ibaHD store can contain multiple time period stores and each time period store contains multiple time periods.

Similar to DAT files, time periods can be processed by *ibaDatCoordinator* when finished. In addition to the standard settings for each job, the HD connection, the time period store and the query period must be configured for time period jobs.



Enabled

If this option is deselected, the job is disabled entirely.

Automatically start on load

If you choose this option, the job starts automatically after the XML-configuration file has been loaded. This can be done either manually or by starting the application or the *ibaDatCoordinator* service.

Retry failed operations of previous session

Enable this option if you want to make sure that operations that could not have been completed in the previous session, are to be retried.

Retry failed operations every...minutes

Here you can set the time after which the failed tasks with time periods marked as failed tasks are performed again. *ibaDatCoordinator* will only execute tasks with time periods that are marked as failed 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 this option is not enabled, *ibaDatCoordinator* will not stop trying to process the tasks that are considered to have failed.

Trigger selection

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

Click the <Select server...> button. In the selection dialog, click on the desired *ibaHD-Server* in the table so that *ibaDatCoordinator* connects to the server. Close the dialog with <OK>.

If user management is enabled on the relevant HD server, you must then enter or select a user name and enter the correct password.

For more information on the HD connection and user administration, see chapter [↗ ibaHD server connection and user management](#), page 144.

In the *Time period selection* window, select the desired time periods to be processed. Further information on handling and filter options for the time period table can be found in chapter [↗ Properties of the time period table](#), page 75.

Process existing time periods

By default, *ibaDatCoordinator* automatically processes new time 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 pressing "GO", the corresponding time-periods are processed and the option is reset to be inactive.


Preferred time base

A time period is usually processed with *ibaAnalyzer*. Therefore, the preferred time base used to query timebased 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 period are reset. 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 to be processed immediately. The periods are always processed directly, regardless of whether they have already been processed or not. If the marker that these have already been processed has not yet been set, this marker is set.

4.5.1 Properties of the time period table

When pressing the  button in the time-period selection area, the time-period table with advanced options is shown.

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...](#)

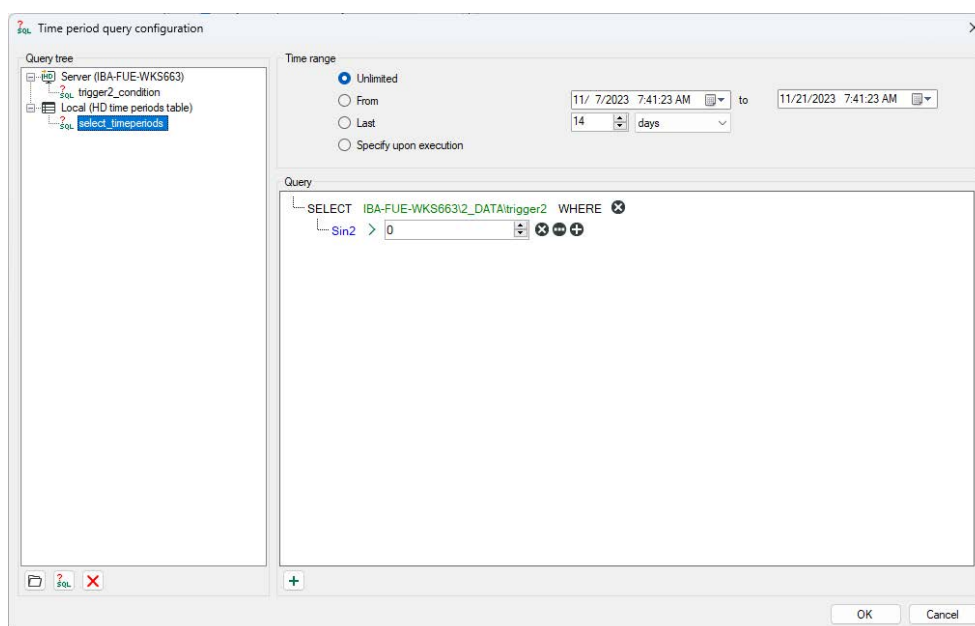
The table shows the latest time periods from the defined time range. The table can be sorted and filtered and computed columns can be shown. The filter has no effect on the processing.


Operating elements of 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/hides the time period table

By default, all time periods of the selected time period store are processed. By using SQL type queries, specific time-periods can be processed.

Pressing the button  opens the query editor. You can add local queries which are stored in the job configuration. Or you can save queries on the *ibaHD-Server* so that all connected clients can use the same server queries.



Select a query from the drop-down list. Pressing the button  executes the query. The table then shows the time periods that fulfill this condition. If the job is started now, only these time periods are processed.

Time-period selection:

select_timeperiods

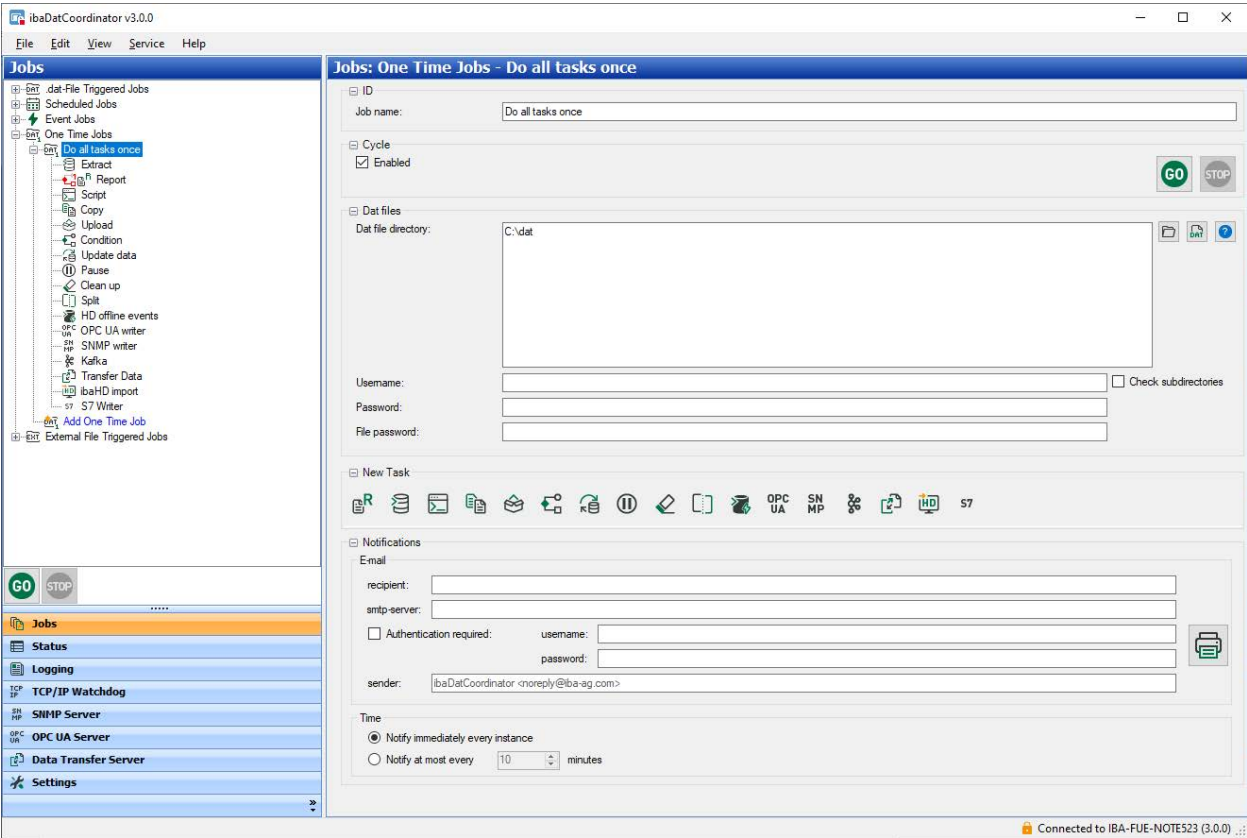
Name	Start time	End time	Sin2	sin4
trig2_2167	11/20/23 9:43:57 AM	11/20/23 9:44:03 AM	0.000214545	0.6488556
trig2_2165	11/20/23 9:43:43 AM	11/20/23 9:43:49 AM	0.008685051	0.7967904
trig2_2163	11/20/23 9:43:29 AM	11/20/23 9:43:35 AM	0.00825568	0.9059212
trig2_2161	11/20/23 9:43:15 AM	11/20/23 9:43:21 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...](#)

A description of the query configuration can be found in chapter [↗ Configuration of queries](#), page 145.

4.6 One time job



With a "One time job", the defined tasks can be executed exactly once, i.e. using a data file. Every reprocessing has to be started manually. This is especially helpful for test purposes, commissioning or in rare individual cases.



For the settings in the categories "ID", "Cycle", "New task" and "Notifications" the explanations from the chapter "dat-file triggered job" are valid. See also chapter [dat-file triggered job](#), page 59.

Dat files

For one-time processing, you can select individual DAT files or directories containing DAT files. You can drag the files or directories using drag & drop from Windows Explorer to the "Dat file directory" field or use the two buttons in the dialog.

	With this button, you can select a directory which is then added to the file list.
	With this button, you can select one or more DAT files, which are then added to the file list.

If the files are stored on a network drive, you can also enter the login data consisting of user name and password in order to gain access.

If you click on the button with the question mark, you can verify the login data for the source directory, optionally with consideration of the subdirectories.

4.7 External File Triggered Jobs

The "External Files Triggered Job" provides the ability to automatically convert external files in various formats to DAT files. The resulting DAT files can be processed as usual with a dat-file triggered job. This allows such files to be integrated directly into the iba workflow.

An additional license is required for the "External File Triggered job". There are separate licenses for each of the different file formats. A license is required for each "External File Triggered job".

Note



For reading the files into *ibaAnalyzer* an additional license *ibaAnalyzer-E-Dat* is required.

The screenshot displays the ibaDatCoordinator v3.0.0 application window. The left sidebar shows a tree view of jobs, with 'Process Text files' selected under 'External File Triggered Jobs'. The main panel is titled 'Jobs: External File Triggered Jobs - Process Text files' and contains the following configuration sections:

- ID:** Job name: Process Text files
- Cycle:**
 - ☒ Enabled
 - ☒ Process external files instantly
 - ☐ Do initial scan for .dat files unprocessed by previous session
 - ☒ Rescan for unprocessed files every 60.0 minutes
- External files:**
 - External file directory: C:\Text_files
 - Username:
 - Password:
 - ☐ Check subdirectories
- File Format:**
 - File extension: txt
 - Analysis to read datafile: C:\pdo_files\my_analysis.pdo
- Notifications:**
 - Email:**
 - recipient:
 - smtp-server:
 - ☐ Authentication required:
 - username:
 - password:
 - sender: ibaDatCoordinator <noreply@iba-ag.com>
 - Time:**
 - ☒ Notify immediately every instance
 - ☐ Notify at most every 10 minutes
- Action on success:**
 - ☐ Delete
 - ☒ Move
- Directory for successfully processed source files:**
 - Target directory:
 - Username:
 - Password:

At the bottom right, a status bar indicates 'Connected to IBA-FUE-NOTES23 (3.0.0)'.

In addition to the default settings for each job, the cycle (when and how often to scan for new files), the directory to monitor and the file format must be configured.

Cycle

Enabled

If this option is deselected, the job is disabled entirely.

Process external files instantly

If this option is enabled, then *ibaDatCoordinator* instantly starts processing the closed external file. If this option is not enabled, then new or unprocessed files can only be searched for and found by the so-called "Rescan" (see below) at regular intervals.

Do initial scan for external files unprocessed by previous sessions

If *ibaDatCoordinator* is started and this option is enabled, the external file directory is scanned for unprocessed files.

If this option is not enabled, then new or unprocessed files can only be searched for and found by the so-called "Rescan" (see below) at regular intervals.

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 files every ... minutes

With this option, you can determine the time *ibaDatCoordinator* scans the directory for unprocessed files. If you disable this option, the rescan function is deactivated. As a result, no files will be processed that are not picked up by the "process external files instantly" option.

Tip

If you only want to process the current external files, e.g. to generate a report, the rescan option should not be enabled. We recommend enabling this option, in case you want to process all external files.

Control elements:

	Starts the current job (only this one).
	Stops the current job (only this one).
	While the job is running, configuration changes are only applied if a manual upload is performed.
	To undo the configuration changes since the last time the job was started, you can download and apply the currently running configuration.

External files

Here you can determine the source directory for the external files. You can enter the name of the directory or select the directory by means of the browser button. If you want to access a network drive, you can enter here the network path in UNC notation with user name and password. By clicking the button with the question mark you can check if the specified source directory can be accessed.

File Format

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

- TEXTFILE
- DAS
- COMTRADE
- PARQUET

For text files several file extensions are possible (*.txt or *.csv). These can be entered manually in the *File extension* field. A PDO file is additionally required for text formats to be read in correctly. 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 stored in the analysis.

Since, in contrast to DAT files, no information can be stored in the files whether the file has already been processed by *ibaDatCoordinator*, another mechanism is used. If the processing is successful, you can select in the *Action on success* area whether the original file will be deleted or moved to another folder. The path to the *Move* option can be configured.

Another feature of the "External File Triggered Job" is that there is only one possible task to convert to DAT format, which is added automatically.

Configuration of the "Convert External File" task

A "Convert External File" task is automatically created for each job. It is not possible to create any other tasks, since the job is intended exclusively for converting files.

The screenshot shows the ibaDatCoordinator v3.0.0 application window. The left sidebar contains a tree view of job categories: .dat-File Triggered Jobs, Scheduled Jobs, Event Jobs, One Time Jobs, External File Triggered Jobs, Process Text files, Process PARQUET files, and Add External File Triggered Job. The 'Process Text files' category is expanded, showing 'Convert text file to DAT' and 'Add External File Triggered Job'. The main window displays the configuration for the 'Convert text file to DAT' job.

Jobs: External File Triggered Jobs - Process Text files - Convert text file to DAT

ID
Task name: Convert text file to DAT
☐ This task is resource critical

Execute
☐ disabled ☐ on success ☐ on 1st failure
☒ always ☐ on failure

Notify
☒ disabled ☐ on success ☐ On 1st failure file
☐ always ☐ on failure ☐ On 1st failure task

Extract analysis file
 [Browse] [Refresh] [Help]

Target
 Target directory: [Browse] [Help]
 Username:
 Password:

Subdirectories
☐ Each hour ☒ Each day ☐ Each week
☐ Each month ☐ Keep original structure ☐ None
☐ Use infofield/expression for subdirectory: ☐ Remove blanks at the end
 Start: Length: ☐ Remove all blanks
 Time used to create directory: [v]
☐ Use multiple levels of subdirectories ☐ Use 4 numbers for the year

Cleanup strategy
☐ None ☒ Limit subdirectories to
☐ Limit disk space usage to Mb ☐ Minimal free disk space Mb

☐ Overwrite existing files in target directory
☐ Set 'Date Modified' time of output file to match .dat file
☐ Use infofield/expression for output file name: ☐ Remove blanks at the end
 Start: Length: ☐ Remove all blanks

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

At the bottom right, it says "Connected to IBA-FUE-NOTES23 (3.0.0)".

Extract analysis file

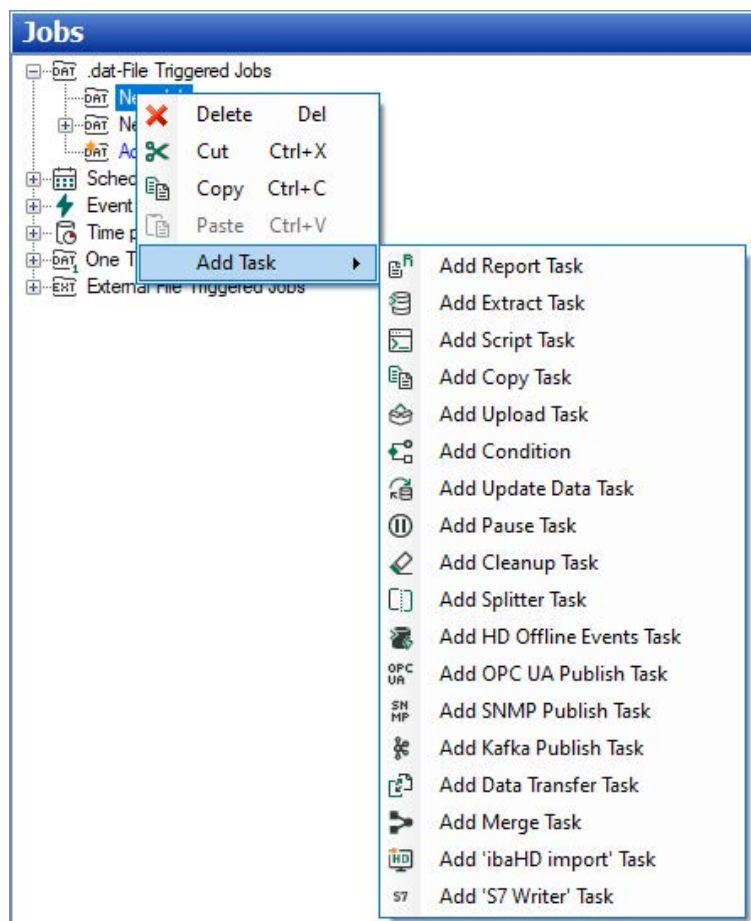
The desired extraction profiles can be defined in a PDO file (selection of signals and sampling rate). If no PDO file is specified, all signals are written to the DAT file in full resolution.

5 ibaDatCoordinator - Performing tasks

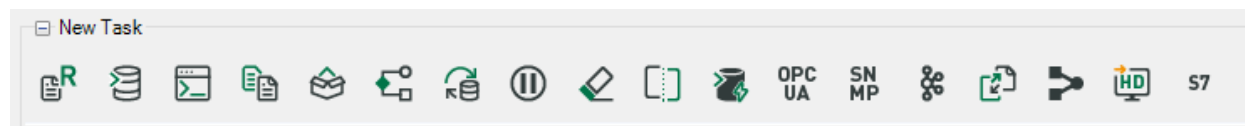
Tasks can only be added to a job if the job has been stopped.

If you click with the right mouse button on the job tree, a context menu is opened. In this menu, you will find a command for adding a task to the currently selected job.

Click on the required task in the following menu.



Alternatively, you can add a task to a job by highlighting the job node on the tree and clicking on one of the task icons.



Note

Some types of task are subject to licensing and/or require the installation of a plugin.

Those are:

- Update data task
- Synchronization task

Hence, the selection of available tasks may look different on your system.

A new task is always added under the job node at the last position.

You can change the sequence later. The task nodes can be moved later via drag & drop. If you drag a task from one job to another or within one job, the tasks will be executed in a different sequence. If you do drag & drop and keep the <Ctrl> button pressed at the same time, a clone of the dragged task node is created.

The positioning of the tasks is important as it corresponds – from top to the bottom – to the processing sequence. Especially if the execution of a task depends on the result of a previous task, you should pay attention to the correct sequence.

5.1 General settings for the task

This chapter and the following chapters describe the different tasks that can be added to a job. The general settings that apply to almost all tasks are outlined first. You will then find an explanation of the more specific settings for each type of task.

All tasks have three settings in common:

- ID
- Execute
- Notify

A number of tasks also have more settings for file processing in common: the target settings.

ID

Task name

You can enter or change the name of the task in this field.

This task is resource critical

You can mark a task as highly resource-intensive with this option.


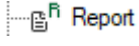




Elsewhere in the settings you will then have the option to limit the number of the resource-intensive tasks that are processed simultaneously in order to stop the system from freezing due to overload.

Tasks with a high demand on resources may, for example, be report or extraction tasks, which are processed in *ibaAnalyzer* with extensive and complex analysis files and/or a high data volume.

If this option is enabled, the respective task is only processed if at the time the task is executed the number of active tasks with high resource requirements does not exceed the defined maximum.

Execute and Notify

You can choose between several "Execute" and "Notify" options. The function of these options is different for "Execute" and "Notify".

	Symbol	Execute	Notify
Disabled		The current task will never be executed.	A notification will never be sent.
Always		The current task will be executed routinely. If the execution fails, it will be retried in the next cycle.	Whenever a DAT file is processed, a notification will be sent regardless of the task status.
on success		The current task is executed if the previous task has been executed successfully or the previous condition is TRUE.	If the current task is executed successfully or the result of a condition is TRUE, a notification will always be sent.
on failure		The current task is executed if the previous task failed or the previous condition is FALSE.	If the current task fails or the result of a condition is FALSE, a notification is always sent.
on 1st failure		The current task is executed as soon as the previous task has failed once or if the result of a previous condition is FALSE.	-
on 1st failure task or file			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. For a job with several tasks a notification is sent each time if any task fails for the first time or any condition is FALSE for the first time.

The "disabled" and "always" options are constantly available in the "Execute" options. The other three options in the "Execute" options are only available if an active previous task is defined.

The "Notify" options are always available.

Monitor ibaAnalyzer

The last options – mostly at the bottom of the task window – refer to the monitoring of *ibaAnalyzer*. These settings are available only for tasks in which *ibaAnalyzer* is involved, i.e. report, extraction, condition and split. You can enable two options: "Memory limit" and "Time limit".

- If you enable the "Memory limit" option, the memory space occupied by *ibaAnalyzer* is monitored. If *ibaAnalyzer* uses more storage capacity than allowed, the task will be interrupted and the DAT file that is currently being processed will be marked as failed for this task.
- If you enable the "Time limit" option, the time that is needed to process a task will be monitored. If the maximum permitted time is exceeded, the task will be interrupted and the DAT file that is currently being processed will be marked as failed for this task.

Target directories and subdirectories

When you configure a copy task or generate a file with a report, extraction, update or splitter task, you must define a target directory. To access a network drive, you can also enter the login data of a write-authorized user, if required.

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

Target

Target directory:

Username:

Password:

Subdirectories

☐ Each hour
 ☐ Each day
 ☐ Each week
 ☐ Each month
 ☐ Keep original structure
 ☒ None

☐ Use infofield/expression for subdirectory:
☐ Remove blanks at the end

Start: Length:
☐ Remove all blanks

Time used to create directory:

☐ Use multiple levels of subdirectories
 ☐ Use 4 numbers for the year

Cleanup strategy

☒ None
 ☐ Limit subdirectories to

☐ Limit disk space usage to Mb
 ☐ Minimal free disk space Mb

☐ Overwrite existing files in target directory

☐ Use infofield/expression for output file name:
☐ Remove blanks at the end

Start: Length:
☐ Remove all blanks

None

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

Each hour/day/week/month

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. A subdirectory that has been created for a day is e.g. named 150821 for the 21/08/2015. The files for the following day would then be saved in the directory 150822. If you want to save the files on a weekly basis, the system calendar (Win-

dows) 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.

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 infofield/expression for subdirectory

Enable this option, if you want to use the content of one or more infofields contained in the DAT file for naming the subdirectories. Thus you can create individual and dynamic subdirectory names. For example, you can store the output files distinguished by customer name, lot number or material code.

For a single infofield enter the name of the infofield in the adjacent entry field and specify the part to be used for the name by means of the start index and the length (number of characters), if required. If you want to use the complete content of the infofield, leave start index and length on 0 (zero).

There is also an option to remove blanks at the end or all blanks from the string of the infofield.

In case of structured infofields, be sure to separate the string sections with a period when entering the name of the infofield.

Instead of a single infofield you may enter an expression consisting of text functions as they are available in the expression builder. This might be a single function *ToText(...)* in order to convert a number into a string and use it in the subdirectory name. You can also combine multiple infofields to one text by concatenating them with the *Concat* function.

You'll find an example for the use of infofields and expressions in ➤ *Using infofields for directory names/filenames*, page 150

Time stamp for the name of the directory

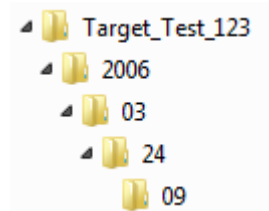
With this setting, you determine how the directory name for the storage of the task result is to be created. Click on the button and select one of the options:

- Current system time
- "Date created" from dat file
- "Date modified" from dat file
- "Start Time" in the 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. Example for subdirectories with several levels and a year with 4 numbers:



Cleanup strategy

The settings for the cleanup strategy can be found for all tasks that generate an output file, such as "Copy files", "Extraction in file" and "Report as 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 takes into account only DAT files. 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 to...

This is only relevant if a subdirectory structure has been selected. Here, the maximum number of remaining directories is defined. When the maximum is reached, the oldest folder will be removed first.

Limit disk space usage to...

The maximum amount of hard disk space used for the specified directory is defined here. The value is defined in megabytes (MB). If this limit is reached, the oldest target files are deleted or overwritten.

Minimal free disk space...

Here, you can define the size of the minimum disk space that has to remain free. The value is defined 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.

Important 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 be in conflict, which can cause undesirable side-effects, such as the wrong data being deleted.

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, we recommend defining the storage limit 10% under the desired hard disk capacity.

Overwrite existing files in target directory

If a file is written to the defined directory, the name of the DAT file is taken and a task-specific file extension is added. If this name already exists in the target directory, an index is added to the file name. If the file name 45124527.dat already exists, for example, and the same DAT file is copied once again to the target directory, a new file named 45124527_00.dat is generated. The index will be incremented by one each following time. If this option is enabled, *ibaDatCoordinator* does not add an index, but overwrites the file with another file that has the same file name.

Note

Duplicate or multiplicate files are created, if a file-generating task (extraction, copy, report) is executed again when a job is repeated with the same source file. A job is repeated if one of its tasks could not be executed and thus the job failed. When a job is repeated, all of its tasks are executed again, including those tasks which had been processed successfully before. In consequence, extraction, copy or report would create files with the same data and name.

In case of an extraction into a database there is a corresponding option in the extractor dialog in *ibaAnalyzer* (*Delete duplicate file table rows*).

Use infofield/expression for output file name

Enable this option, if you want to use the content of one or more infofields contained in the DAT file for naming the output files. Thus you can create individual and dynamic file names. For example, you can include e.g. customer name, lot number or material code in the output file-name.

Enter the name of the infofield in the adjacent entry field and specify the part to be used for the name by means of the start index and the length (number of characters), if required. If you want to use the complete content of the infofield, leave start index and length on 0 (zero).

There is also an option to remove blanks at the end or all blanks from the string of the infofield.

In case of structured infofields, be sure to separate the string sections with a period when entering the name of the infofield.

Instead of a single infofield you may enter an expression consisting of text functions as they are available in the expression builder. This might be a single function *ToText(...)* in order to convert a number into a string and use it in the subdirectory name. You can also combine multiple infofields to one string by concatenating them with the *Concat* function.

You'll find an example for the use of infofields and expressions in [➤ Using infofields for directory names/filenames](#), page 150

5.2 Report task

The screenshot displays the ibaDatCoordinator v3.0.0 software interface. On the left, a 'Jobs' tree lists various tasks, with 'Report' highlighted under 'One Time Jobs'. The main window is titled 'Jobs: One Time Jobs - Do all tasks once - Report'. It contains several configuration sections: 'ID' with 'Task name' set to 'Report'; 'Execute' with 'always' selected; 'Notify' with 'disabled' selected; 'Report analysis file' pointing to 'C:\pdo_files\my_analysis.pdo'; 'Target' set to 'File' in 'pdf' format; 'Target directory' set to 'C:\Reports'; 'Subdirectories' set to 'Each day'; and 'Cleanup strategy' set to 'None'. There are also checkboxes for 'Remove blanks at the end' and 'Remove all blanks'. At the bottom, there are settings for 'Monitor ibaAnalyzer' with memory and time limits.

With this task, you can generate reports automatically. The report can be printed via a default printer or issued as file in one of the formats mentioned below. The report is generated using an analysis file (PDO file) which was created with *ibaAnalyzer*.

The following outlines the report-specific settings. For all other settings, see chapter [General settings for the task](#), page 83.

Analysis file for report task

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 adjacent *ibaAnalyzer* button to check whether *ibaAnalyzer* can open the PDO file.

Selecting the output medium

Select whether you want the report to be printed using a default printer or written to a file. If the report is written to a file, there are different document formats available in a drop-down list.

The following document formats are available:

File extension	File type	Remark
.pdf	(Acrobat) Portable Document Format	-
.htm	Hypertext Markup Language file	html v3.2-compatible no overlapping objects allowed Turned RTF not supported
.mht .mhtml	Multi Mime HTML	html v3.2-compatible no overlapping objects allowed Turned RTF not supported
.txt	Text file	Table export only
.xls	Microsoft Excel	Excel version-independent (native export) RTF text embedded as picture Overlapping objects are not fully supported
.rtf	Rich Text Format file	Turned RTF and images are not supported
.tif .tiff	Tagged Information File Format graphics	-
.emf	Enhanced Meta File graphics	-
.jpg .jpeg	Joint Pictures Expert Group graphics	-
.bmp	Standard Windows Bitmap graphics	
.xml	Extensible Markup Language	

Date of the output file

If you enable this option, the modification date of the original DAT file is adopted as date of the output file.

Note



If *ibaDatCoordinator* runs in service mode, 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 you want to print something from a network printer, the *ibaDatCoordinator* service may have to be run from a suitable domain user which has sufficient access rights to the network printer.


Reference



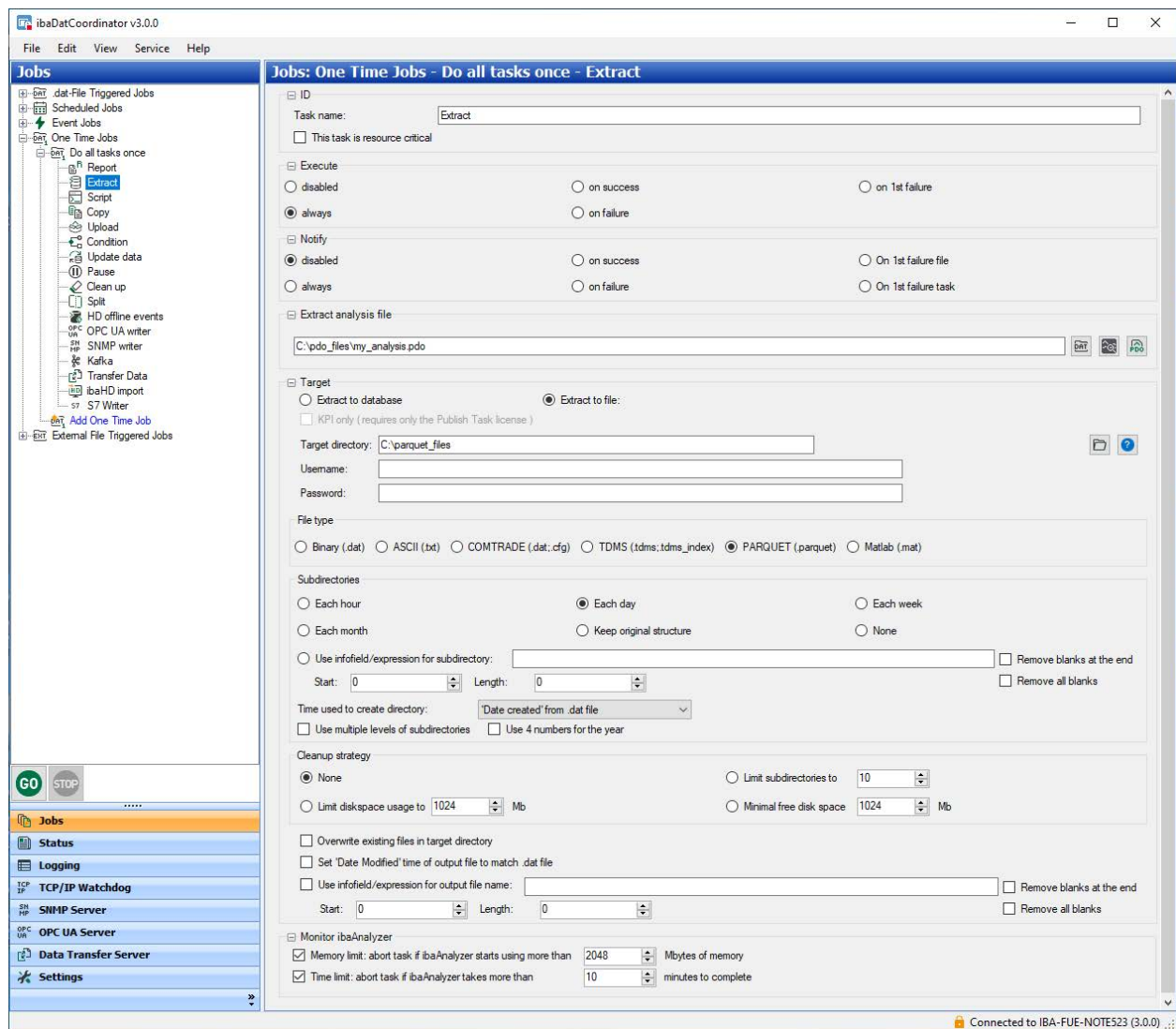
For information about generating the PDO files and the definition of the report, please see the *ibaAnalyzer* manual.

Configuration steps

For configuring the report task, proceed as follows:

1. Select the analysis file (*.pdo) for the report generation by defining the directory or search using the  button.
2. Then you can click on the *ibaAnalyzer* symbol to make sure that the correct analysis file has been loaded. The analysis file is opened in *ibaAnalyzer* and can be edited there.
3. Define the target of the report
Please define if the target is a printed report or a report saved as file. If you have selected the "Print" option, you can skip the following steps.
4. Specify the output file type.
5. Define the target directory or select it via the browser button.
6. If the target directory is located on a remote server, enter a user name and a password in order to gain access to the remote server.
7. Check access to the target directory by clicking on the <?> button.
8. Define the organizational settings for the target directory.

5.3 Extraction task



If you have a license for *ibaAnalyzer*, an extraction task can extract data from a DAT file and store it in a database or another file. The extractions are done with an analysis file created with *ibaAnalyzer*. For more information, please see the manual for *ibaAnalyzer-DB* or *ibaAnalyzer-File-Extract*.

The following describes the extraction-specific settings. For all other settings, see chapter [General settings for the task](#), page 83.

Extract analysis file

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 adjacent *ibaAnalyzer* button to check whether *ibaAnalyzer* can open the PDO file.

Target

Extract to database/Extract to file

When configuring a DB extract task (*Extract to database*) in ibaDatCoordinator, only a PDO-file is required which contains all relevant settings for the DB extract.

One additional option *Extract only „Computed Columns“ and „Info Columns“* is available to extract only computed columns and info columns, i.e. to fill only the "File table". If this option is checked, *ibaAnalyzer* will ignore the profile settings in the PDO-file and extract only the computed columns and info columns to the database and no channel or segment data.

The advantage is that this option can be used with an *ibaDatCoordinator-Publish* license instead of a full *ibaDatCoordinator-DB* license.

When selecting *Extract to file*, further settings are required.

File type

Select here the required type for the output file.

File type	Remark
Binary (.dat)	iba DAT file
ASCII (.txt)	ASCII file, tab-separated format
COMTRADE (.dat, .cfg)	ASCII-file specifically for measurement purposes in energy technology
TDMS (.tdms, tdms_index)	National Instruments file format
PARQUET (.parquet)	Apache Parquet format
Matlab (.mat)	Matlab format

Reference



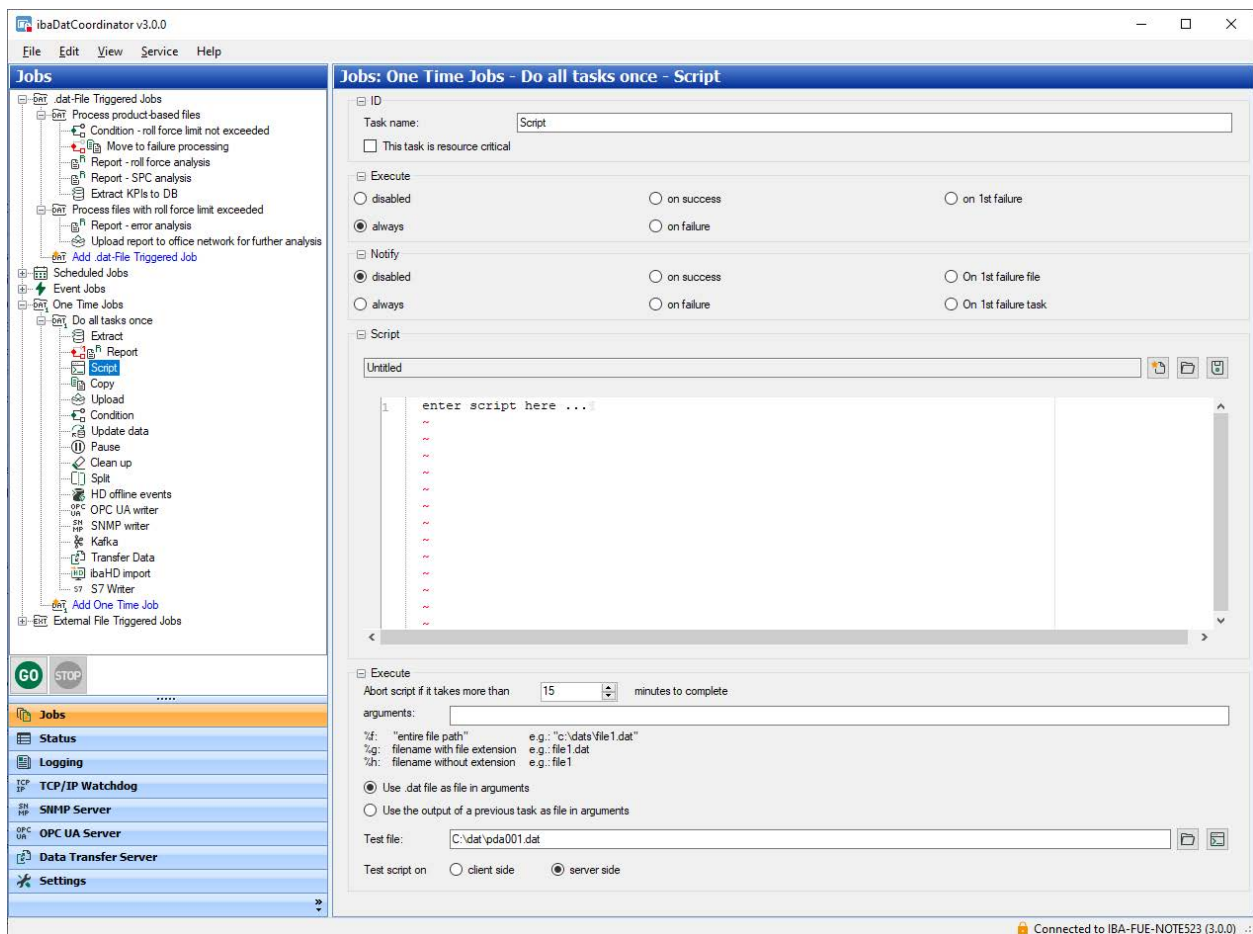
Further information on the file types can be found in the *ibaAnalyzer-DAT-Extractor* manual.

Configuration steps

If you want to extract the data automatically from the DAT files, proceed as follows:

1. Select the *Extract analysis file* by entering the path or search for the file using the browse button.
2. Then, you can click on the *ibaAnalyzer* symbol to check whether the correct analysis file has been loaded. The analysis file is opened in *ibaAnalyzer* and can also be edited here.
3. Configure the target of the extraction:
Select the type of the extraction (database or file) and skip the following steps if "Extract to database" has been selected.
4. Define the target directory.
5. If the target directory is located on a remote server, enter a user name and a password in order to gain access to the remote server.
6. Check access to the target directory by clicking the <?> button.
7. Choose the file type of the exported file (binary, ASCII, COMTRADE, TDMS, PARQUET or Matlab). The selected file type must correspond to the file type in the analysis file.
8. Define the organizational settings for the target directory.

5.4 Script task




A script task allows you to run a script you have created. Scripts that can normally also be run by Windows command prompt, such as batch files, VBS (Visual Basic Script) and JVS (Java Script), are supported.

A script can be written using *ibaDatCoordinator*. However, you can also open an existing script using the browser button. If *ibaDatCoordinator* identifies the code, the program highlights the syntax. To use the script in *ibaDatCoordinator* it has to be stored in an accessible location.

Execute

In *ibaDatCoordinator* you can use arguments in this script. There are three types of iba-specific arguments. You can use these arguments via 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 of a previous task; the used file depends on the selection in *ibaDatCoordinator*:

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

You can test the script by choosing a DAT file. This file can be selected via the browser button or the path. If you have selected the test file, click  to start the test. 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 note, that the iba support desk can only give limited support in case of problems with user-created scripts due to their individual program code.

5.5 Copy task

In a copy task, you can copy, move or delete the current data file or the file that has been issued by a previous task.

Configuration steps

1. Select the file type on which you want to execute the copy task. This can be a DAT file or the file issued by a previous task. The output of a previous task may be a report or an extracted file.
2. Choose the actions that should be executed on the file by the copy task (copy, move or delete).
3. Check the "Create ZIP file" option if you want the file to be converted to a ZIP archive.

4. Define the target for the file that is to be copied or moved.
5. Enter the target directory or select the target using the browser button.
6. If the target directory is located on a remote server, enter a user name and a password in order to gain access to the remote server.
7. Test the access to the target directory by clicking the <?> button.
8. Define the organizational settings for the target directory.

5.6 Upload task

An upload task, similar to the copy task, can be used to upload files to a remote server. The task can send either the currently processed DAT file or the result of a previous task (e.g. a PDF file).

You can select the transmission protocol in the *Protocol* field. The <?> button can be used to test the respective connection settings. The connection settings differ depending on the protocol.

FTP

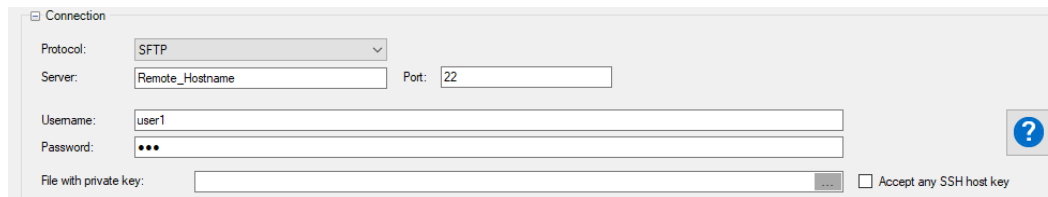
For the connection via FTP, the server and the port (default setting: 21) are required. User name and password can be entered for authentication. Alternatively, you can use the anonymous login. Then the username is "anonymous" and the password is "anonymous".

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 FTPS. In the latter case, a certificate can

be specified to establish the connection. Alternatively, any certificate can be accepted by the server.

You can switch between active and passive mode under *FTP Mode*.

SFTP

The SFTP Connection dialog box is titled "Connection". It contains the following fields: "Protocol" (a dropdown menu set to "SFTP"), "Server" (a text box containing "Remote_Hostname"), "Port" (a text box containing "22"), "Username" (a text box containing "user1"), "Password" (a text box with masked characters "•••"), and "File with private key:" (a text box with a browse button "..."). There is a checkbox labeled "Accept any SSH host key" which is currently unchecked. A blue question mark icon is located on the right side of the dialog.

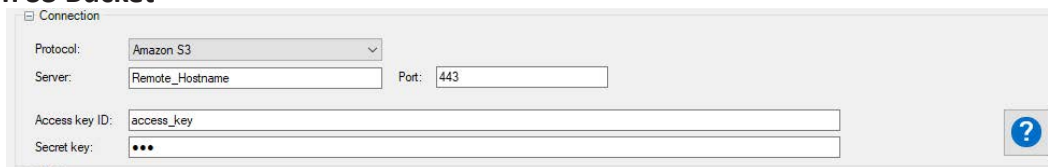
For the connection via SFTP the server and the port (default: 22) are required. Authentication can be done either with username and password or with a "private key".

SCP

The SCP Connection dialog box is titled "Connection". It contains the following fields: "Protocol" (a dropdown menu set to "SCP"), "Server" (a text box containing "Remote_Hostname"), "Port" (a text box containing "22"), "Username" (a text box containing "user1"), "Password" (a text box with masked characters "•••"), and "File with private key:" (a text box with a browse button "..."). There is a checkbox labeled "Accept any SSH host key" which is currently unchecked. A blue question mark icon is located on the right side of the dialog.

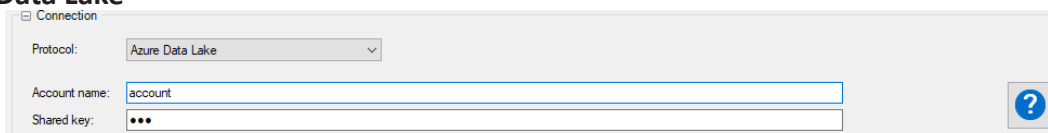
For the connection via SCP the server and the port (default: 22) are required. Authentication can be done either with username and password or with a "private key".

Amazon S3 Bucket

The Amazon S3 Connection dialog box is titled "Connection". It contains the following fields: "Protocol" (a dropdown menu set to "Amazon S3"), "Server" (a text box containing "Remote_Hostname"), "Port" (a text box containing "443"), "Access key ID:" (a text box containing "access_key"), and "Secret key:" (a text box with masked characters "•••"). There is a checkbox labeled "Accept any SSH host key" which is currently unchecked. A blue question mark icon is located on the right side of the dialog.

In addition to the server and port for the connection to Amazon S3, the "Access key ID" and the "Secret key" of an AIM user with the appropriate permission are required. For more information, please refer to the AWS (Amazon Web Services) documentation.

Azure Data Lake

The Azure Data Lake Connection dialog box is titled "Connection". It contains the following fields: "Protocol" (a dropdown menu set to "Azure Data Lake"), "Account name:" (a text box containing "account"), and "Shared key:" (a text box with masked characters "•••"). There is a checkbox labeled "Accept any SSH host key" which is currently unchecked. A blue question mark icon is located on the right side of the dialog.

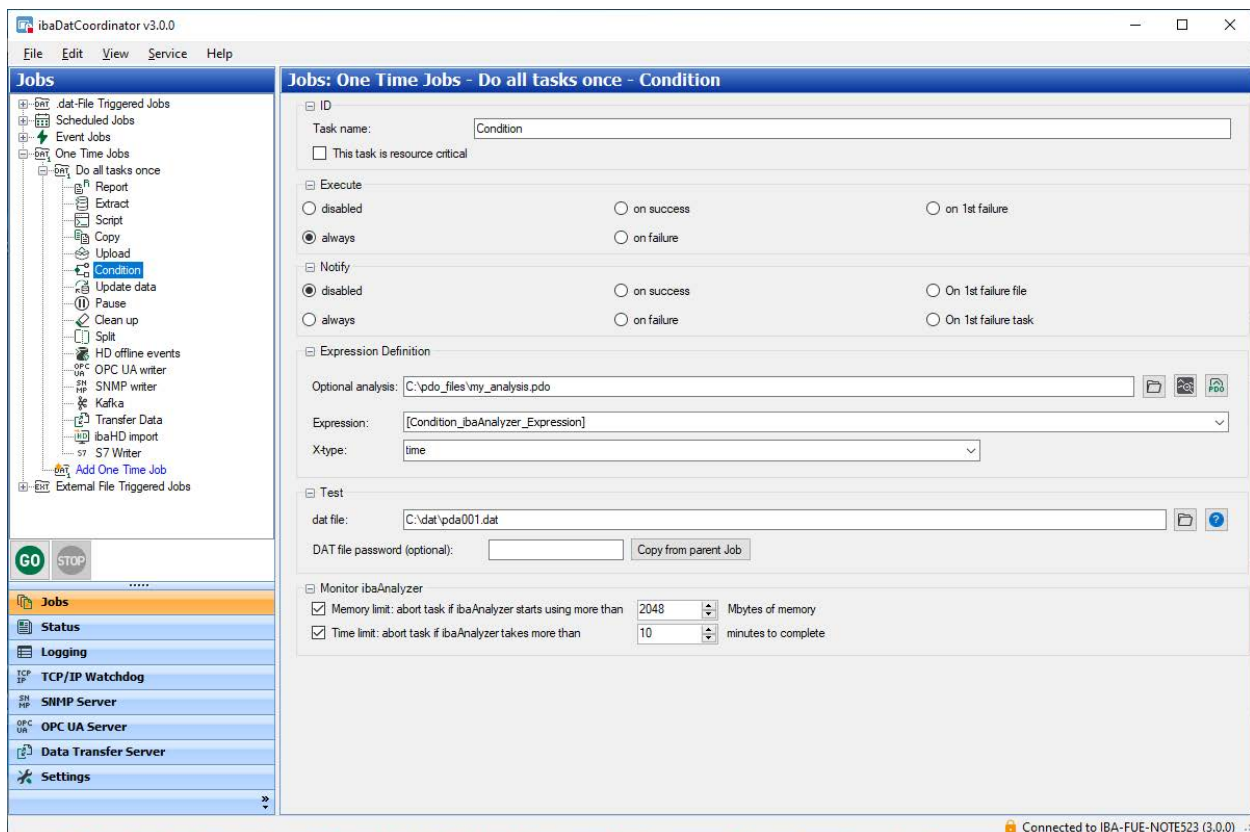
The "Account name" and "Shared key" for the data lake must be specified to connect to the Azure Data Lake. For more information, please refer to the Azure documentation.

Remote path und Target

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

The option "Create zip archive" can be used to pack the DAT file into a ZIP archive before uploading.

5.7 Condition task



You can use a condition, if you want to control the execution of a subsequent task by using an expression whose result may be TRUE or FALSE.

The condition can be generated either directly from the signals in the DAT file or in an analysis file (*.pdo) with logical signal definitions and sub-expressions. You can define a PDO file for this purpose.

The result of the expression might also not be TRUE or FALSE only. If this is the case the following rules apply:

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

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 type of the X-axis defines how *ibaAnalyzer* calculates an expression. *ibaAnalyzer* can interpret an expression as time, length, frequency or inverse length.

Note

The condition only affects the directly following task which is arranged on first position beneath the condition in the job's tree view!

Furthermore, the execution option for the task which should be executed or not, depending on the condition, should be set to "on success" or "on failure".

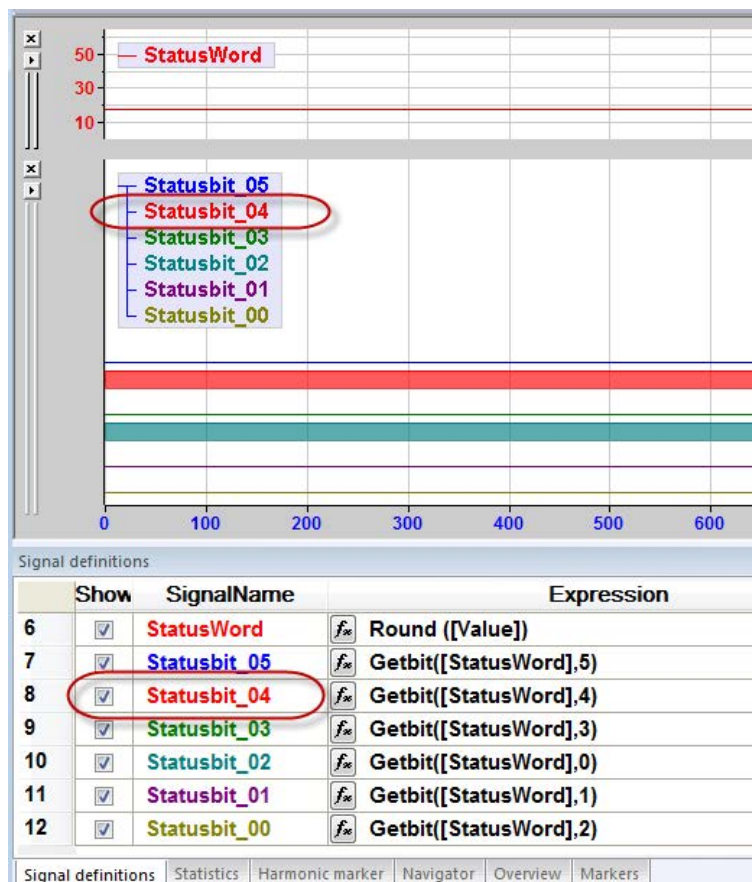
A dependency is indicated by an indented task icon in the tree.

Expression definition**Optional analysis**

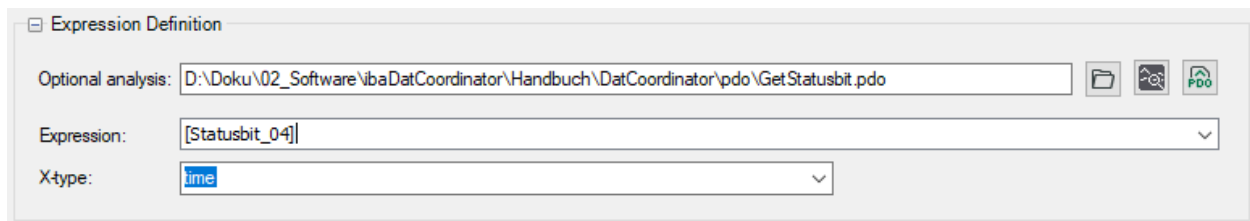
Enter here the complete path and file name of an analysis file from which you want to use an expression or a calculated result as condition. *ibaDatCoordinator* then opens the DAT file and loads the analysis file defined here, in which a suitable expression is found. In the "Expression" field, only the name of the (completed) expression has to be entered.

Example:*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](#)).



The following settings have to be made in *ibaDatCoordinator*.



Expression Definition

Optional analysis: D:\Doku\02_Software\ibaDatCoordinator\Handbuch\DatCoordinator\pdo\GetStatusbit.pdo

Expression: [Statusbit_04]

X-type: time

As "Statusbit_04" is already a binary signal, and hence can only be TRUE or FALSE, only the name has to be entered under "Expression".

Expression

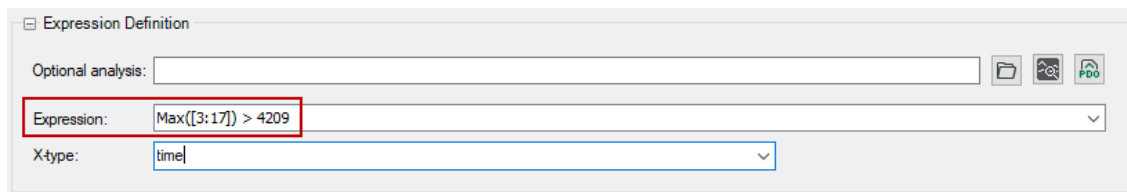
If you want to configure the condition without an analysis file, simply enter a signal that is contained in the data file in the "Expression" field. You can also enter a formula or an expression, in which signals from the data file are used. In this case, the command syntax of *ibaAnalyzer* must be taken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

Example:

Condition without analysis file

Subsequent tasks are only to be processed if the maximum of signal [3:17] in the data file is greater than 4209.

In *ibaDatCoordinator*, the following settings have to be made:



Expression Definition

Optional analysis:

Expression: Max([3:17]) > 4209

X-type: 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". For this purpose, click the test button <?> .

Configuration steps

If you want to create a condition, carry out the following steps:

1. Optional: Open the *ibaAnalyzer* analysis file (PDO). When defining an analysis, the expression can have subexpressions defined in the analysis, which are needed in the conditional expression.
2. Create an appropriate expression.
3. Select the type of the X-axis for this expression.
4. Test the expression by clicking the <?> button.

5.8 Update data task

The screenshot shows the 'Update data' task configuration in the ibaDatCoordinator v3.0.0 interface. The left sidebar lists various job types, with 'Update data' highlighted. The main window displays the configuration for the 'Update data' task, which is part of a 'One Time Jobs - Do all tasks once' set.

Task name: Update data

☐ This task is resource critical

Execute:

- ☐ disabled
- ☒ always
- ☐ on success
- ☐ on failure
- ☐ on 1st failure

Notify:

- ☒ disabled
- ☐ always
- ☐ on success
- ☐ on failure
- ☐ On 1st failure file
- ☐ On 1st failure task

Database info:

Database provider: Computer: ☒ Local machine ☐ Database server

Database name: Database server:

Authentication: ☒ Use Windows NT authentication ☐ Specify authentication info

Username: Password:

Table name:

Target:

Target directory: Username: Password:

Subdirectories:

- ☐ Each hour
- ☐ Each month
- ☐ Each day
- ☐ Keep original structure
- ☐ Each week
- ☒ None

☐ Use infofield/expression for subdirectory:

Start: Length:

Time used to create directory:

☐ Use multiple levels of subdirectories ☐ Use 4 numbers for the year

☐ Remove blanks at the end ☐ Remove all blanks

Cleanup strategy:

- ☒ None
- ☐ Limit subdirectories to
- ☐ Limit disk space usage to Mb
- ☐ Minimal free disk space Mb

☐ Overwrite existing files in target directory

☐ Set 'Date Modified' time of output file to match .dat file

Buttons: GO, STOP

Bottom status bar: Connected to IBA-FUE-NOTES23 (3.0.0)

The "Update data" task is subject to licensing.

With an "Update data" 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. This task is only available if you have a special license for *ibaDatCoordinator*.

Database info

You can configure the connection to the database in this section.

Database provider

Select the type of database used from the drop-down list.

Computer

Select the computer and the database server on which the database is located. If the database is located on the same computer on which *ibaDatCoordinator* is also running, you can choose "Local machine".

If the database is located on a computer in the network, choose "Database server" and enter the name of the computer or the IP address as well as – if required – the database server (service). The "Database server" selection also works if a database is located on a local computer

and you enter the name of the computer and the database server (service), e.g. "MyComputer\SQLEXPRESS" for an SQL-Express database.

Database name

Enter here the name of the database in which the information for the update data task is saved.

Authentication

If you want to get access to the database or the database server, you might need special login data, especially for access via the network. Please ask your IT or database administrator which credentials are required.

Table name

Enter here the name of the database table in which the data for the update data task can be found.

Information about the database table

The following columns have to be available in the database table:

- 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.

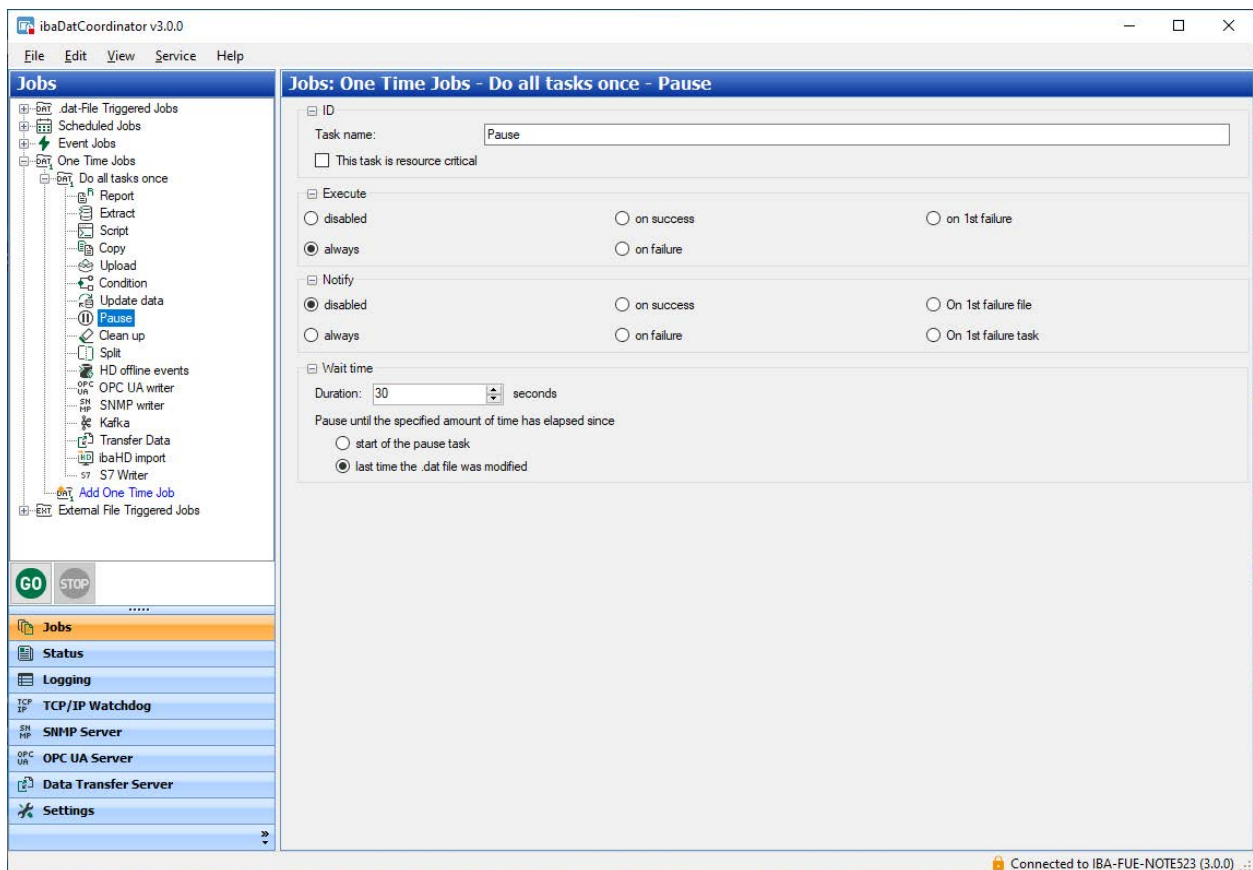
If an info field contains the value NULL, this info field is not generated.

Configuration steps

If you want to update the DAT files and want to rename them, carry out the following steps:

1. Select the type of the database in the database information section:
 - SQL server
 - ODBC database (e.g. MySQL)
 - Oracle
 - DB2-UDB
2. Enter the name of the database.
3. Enter the database server or search the server using the browse button:
 - Local machine or
 - Database Server: Enter the IP address or the name of the computer on which the database can be found. If required, add the database server (service) as well.
4. Specify the authentication method:
 - If you use the Windows authentication (only available with MS SQL), log in on the database server with the account under which *ibaDatCoordinator* was started.
 - If you use another authentication method, you can log in under an account that was specifically created for using the database. Enter the login information (user name and password) for this account.
5. Define the database table you want to retrieve the update information from by entering the name of the table.
6. Check the database configuration by clicking on <Test database connection>. If the database cannot be reached, the specified table does not exist or one or more of the table columns specified above are missing, an error message is displayed.
7. Configure the target for the processed file.
8. Specify the target directory or select it using the browse button.
9. If the target directory is located on a remote server, enter a user name and a password to gain access to the remote server.
10. Test the target directory by clicking <?>.
11. Define the organizational settings for the target directory.

5.9 Pause task



This task allows you to pause a job for a defined period of time.

There are two different ways to define the execution of the pause task:

(since)...start of the pause task

The pause task is always executed and stops the job execution for a certain defined period of time.

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

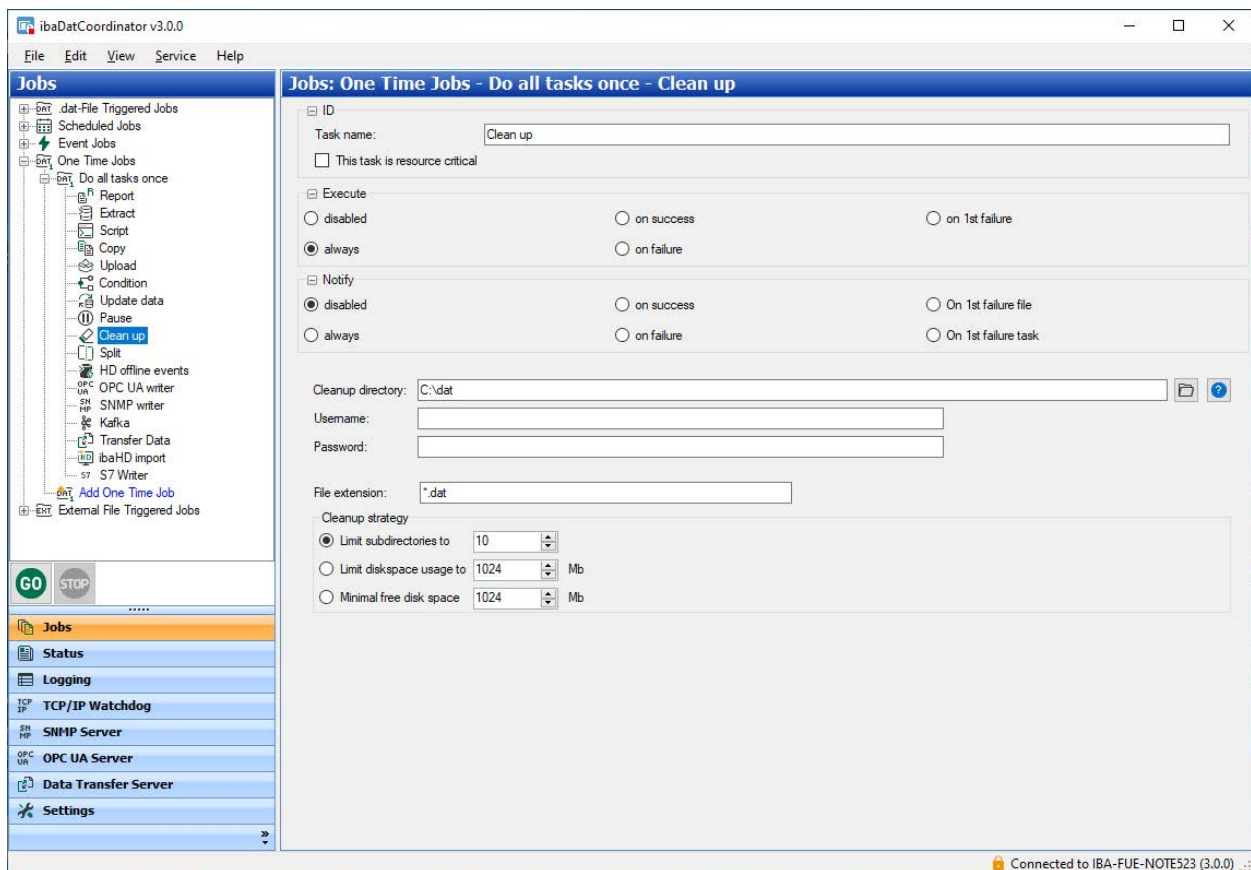
ibaDatCoordinator checks when the DAT file was modified. If more than the defined period of time has passed since the time of the modification, the job is not stopped. Otherwise, the job will be paused for a period of time that corresponds to the remaining period between the specified duration time and the difference between the current time and the modification time.

Configuration steps

If you want to define a pause task, carry out the following steps:

1. Specify the duration of the pause.
2. Select the start condition.

5.10 Cleanup task



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

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, temporary files that are created as part of scripting tasks can always be removed automatically to free up storage space.

The cleanup task can be executed – like other tasks – depending on a previous task or a condition in the course of a job. The cleanup task is especially suited for scheduled jobs, e.g. to clean directories on a weekly clean up.

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

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 by another entry or – separated by a comma – add more extensions, e.g. **.dat, *.pdf, *.jpg*

The only criterion for deleting files is the date of the files.

Cleanup strategy

The specified directory is monitored according to one of the possible cleanup strategies.

- Maximum number of subdirectories

If the maximum number of subdirectories has been exceeded, the oldest subdirectories with all files that have a specific extension, are deleted. If all files in a directory are deleted, the subdirectory is also deleted. If there are files with other extensions than the specified ones are stored in the subdirectory, these files will not be deleted and *ibaDatCoordinator* cannot delete the subdirectory.

- Maximum disk space usage

If the sum of all files in the directory exceeds the specified limit, the oldest files will be deleted, until the data volume falls below the limit again. 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 disk space on the hard disk falls below the defined 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 is only able to 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.

Tip

If you use cleanup strategies, consider 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 to be able to judge which data sets are especially fast-growing.

5.11 Synchronization task

This type of task can be used to synchronize the database and file system. It is only available if the *DBSync* plugin has been installed.

This kind of task is designed to be used by a scheduled job.

ibaDatCoordinator version 1.23 or later is required to use the plugin.

When extracting values from a data file into a database using *ibaAnalyzer-DB*, an entry is created in the index table (ibaFile table) for each data file. This entry contains user-defined values for info fields and calculated columns, as well as a number of information for each data file, among others the file name with complete path specification.

The file name is used by *ibaAnalyzer* or *ibaDatManager* as a reference for the drill-down into the high-resolution raw data.

A file reference could become invalid by moving the original DAT file to another folder or removing the file completely from the system, e.g. via a cleanup task.

The plugin *DBSync* will verify the file references from the index table in the database by checking for the existence of the corresponding data file. The result of this validity check is stored in an additional column of the index table. The status '1' for an entry in 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 *DBSync* plugin allows the deletion of the entries for which no data file could be found.

Installation of the DBSync plugin

ibaDatCoordinator version 1.23 or higher must be installed on the computer.

Execute the installation file *DBSyncPluginInstaller_v1.x.x.exe* and follow the instructions.

Creating a scheduled job

Create a scheduled job for the execution of a synchronization task in order to execute the task on a regular basis, e.g. every 8 hours or once a day. Add a synchronization task to the job.

Configuration of the synchronization task – database connection

Set up the database connection:

The screenshot shows a 'Database info' configuration window. It contains the following fields and options:

- Database provider:** A dropdown menu set to 'Sql-server'.
- Computer:** Radio buttons for 'Local machine' and 'Database server'. 'Database server' is selected.
- Database name:** A text box containing 'ibaDatMgr'.
- Database server:** A text box containing 'IBA-BLN-NOTE345\SQLEXPRESS' with a browse button (folder icon) to its right.
- Authentication:** Radio buttons for 'Use Windows NT authentication' (selected) and 'Specify authentication info'.
- Username:** An empty text box.
- Password:** An empty text box.
- Table name:** A text box containing 'deFile'.
- Test database connection:** A button located to the right of the authentication options.

Note

In order to use the results of the synchronization in the *ibaDatManager* application the database connection must refer to the same database and the same index table, which is configured in *ibaDatManager*.

If you are using different views in *ibaDatManager*, which refer to different index tables then you should create a separate DBSync task for every index table.

Database provider

Selection between SQL-Server, Oracle, DB-2 or ODBC (Access, MySql)

Computer

Local machine or a specific database server

Database name

Database name or DSN (ODBC connection)

Authentication

By operating system (NT authority) or using username and password

Table name

Name of the index table (ibaFile table)

Click the button <Test database connection> in order to verify the database connection.

Configuration of the synchronization task – parameters
Delete records where no file can be found

Disabled: An additional column named `FILE_REFERENCE_VALID` is set to '1' for valid references and '0' for files, which cannot be found.

Enabled: The records in the file table marked with '0' in the column `FILE_REFERENCE_VALID` are going to be deleted from the table.

Replace part of reference file path

Disabled: The path as stored in the database column `_filename` is used for accessing the file.

Enabled: For accessing the file the first part of the reference (path) in column `_filename` ("from") will be replaced by a new path ("to").

From

Part of the original path name, which should be replaced.

To

New part of the path name (where the file is supposed to be stored).

UNC credentials

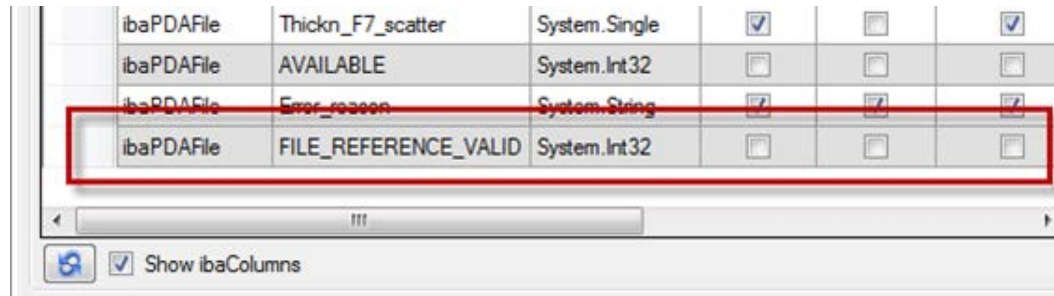
In case *ibaDatCoordinator* needs to access a shared folder the login and password of an authorized user can be set here.

Note



The column FILE_REFERENCE_VALID will be created automatically by *ibaDatCoordinator* if it does not exist.

In *ibaDatManager* you can see the column in the configuration dialog of the data source, provided you have enabled the "Show ibaColumns" option:



During synchronization, all processes will be logged by the DBSync plugin.

For more information about logging, see chapter [Logging](#), page 27.

5.12 Splitter task

With the splitter task, a DAT file can be split into multiple small DAT files. The files contain the same signals as the original DAT file, but with smaller time ranges.

If the original file is protected with a password the resulting files will be protected with the same password as well.

The screenshot shows the ibaDatCoordinator v3.0.0 interface. The left pane displays a tree of jobs, with 'Split' selected under 'One Time Jobs'. The main pane shows the configuration for the 'Split' task. The configuration includes:

- Task name:** Split
- Execute:** ☒ always, ☐ on success, ☐ on failure
- Notify:** ☒ disabled, ☐ on success, ☐ on failure, ☐ On 1st failure file, ☐ On 1st failure task
- Expression Definition:**
 - Optional analysis: C:\pdo_files\my_analysis.pdo
 - Expression: [Split_ibaAnalyzer_Expression]
 - Split from: Rising to rising edge
- Test:**
 - dat file: C:\dat\pda001.dat
 - DAT file password (optional): [] Copy from parent Job
- Target:**
 - Target directory: C:\split
 - Username: []
 - Password: []
- Subdirectories:**
 - ☒ Each day, ☐ Each week, ☐ Each month, ☐ Keep original structure, ☐ None
 - Use infofield/expression for subdirectory: [] Remove blanks at the end, ☐ Remove all blanks
 - Start: 0, Length: 0
 - Time used to create directory: Date created from .dat file
 - ☐ Use multiple levels of subdirectories, ☐ Use 4 numbers for the year
- Cleanup strategy:**
 - ☒ None, ☐ Limit subdirectories to 10, ☐ Limit disk space usage to 1024 Mb, ☐ Minimal free disk space 1024 Mb
- Monitoring:**
 - ☐ Overwrite existing files in target directory
 - ☐ Set 'Date Modified' time of output file to match .dat file
 - ☐ Use infofield/expression for output file name: [] Remove blanks at the end, ☐ Remove all blanks
 - Start: 0, Length: 0
 - ☒ Monitor ibaAnalyzer
 - ☒ Memory limit: abort task if ibaAnalyzer starts using more than 2048 Mbytes of memory
 - ☒ Time limit: abort task if ibaAnalyzer takes more than 10 minutes to complete

Settings which are specific to splitter tasks can be adjusted in the following sections:

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 is optional. If the expression for splitting the DAT file is sufficient, it is not 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 the adjacent *ibaAnalyzer* button 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 2 options to choose from:

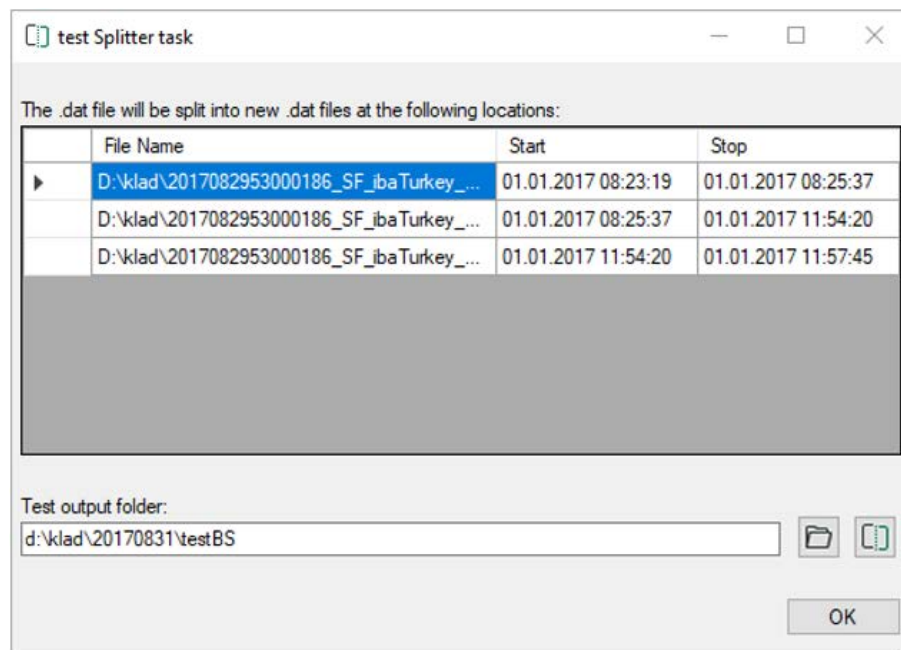
- Rising to rising edge: The first part of the DAT file is split from the beginning of the time range until the first time the expressions is changed from FALSE to TRUE. Furthermore, the file is split from each transition from FALSE/TRUE to the next transition from FALSE/TRUE. The last part starts from 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 when the expression was FALSE. This data will not be represented.


Test

In the "Test" section you can test the splitter task using a test file. Enter the path name and file name of the test file in this 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 splitter task is not supported when the service is running on a remote system.



The potential part files are listed in the table. This can take a while depending on the complexity of the expression, the optional analysis and the volume of data. 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. Use the <Split> button  to begin the splitting of the DAT file according to the split instructions. Instead of the <Split> button, a <Stop> button appears again, which you can use to cancel the process. Click <OK> to leave the dialog.

Target

Here you can specify the target directory for the generated DAT files. Information for authentication in network drives, for setting up subdirectories and for a cleanup strategy can be found in chapter [↗ General settings for the task](#), page 83.

5.13 HD Offline event task

The HD offline event task can be used to generate one or more HD events, based on data files or HD queries.

With each execution of the task the event will be written into an HD event store at the correct time.

If you want to generate multiple events you can also configure multiple tasks.

The task can be configured with every kind of job.

The number of occurrences of an event depends on

- if and how often an offline event task is executed and
- if the event is triggered once or multiple times by a trigger signal (pulse)

An outstanding property of offline events is the ability to store additional information together with the occurrence of the event in the HD store:

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

Hence, the computation of (statistical) key values over a time range, a process phase or for a product can be done without the need of a database extraction solely inside the *ibaHD-Server* including saving the results there too.

Note



In order to benefit comprehensively from the Offline Event function you will at least need the following versions of the interacting software products:

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

5.13.1 Task specific settings

The following describes the task-specific settings. For all other settings, see chapter [General settings for the task](#), 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.

PDO 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 will always be applied to the data file (.dat file triggered job) or HD query (scheduled job, event job) respectively.

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 triggered jobs you should select one of the original data files as a template.

In case of scheduled 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.

Store selection

Store selection

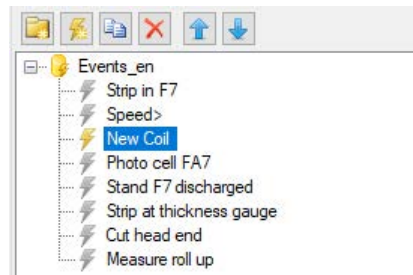
ibaHD server: IBA-BLN-WKS197 Port: 9180 Select server...

Username: Change user...

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 a user management is enabled on the selected server, click on <Change user...> and select a user with write access which can be used by *ibaDatCoordinator* when accessing the *ibaHD-Server*.

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 *ibaHD* event tree shows all available events on the connected *ibaHD* server. The event 'Coil finished' is the defined event in the current task.

It is possible to write data to any of these events from a single HD offline events task. A visual distinction between events is made by different colors of the event icon ("flash").

A yellow event icon indicates that this event is actively written by the current task. A grey icon indicates that the event is available on the *ibaHD* server but currently not written by this task. The event may be written by another task in the same *ibaDatCoordinator* configuration on the same computer or it may be written by another system, e. g. an *ibaPDA* HD event data store.

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

Adding, managing and removing of events

In the upper pane (event tree) you can add an event by using the buttons. If required you may create folders for a better organization of events. Moreover, you can change the order of the events in the tree and copy or remove them.

	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

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 and 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



You'll find a detailed description of the HD Event wizard in the manual of the product *ibaHD-Server*.

5.13.2 Configuration tab

Trigger

Active

Enable this checkbox if the event should be written by the current task. By activation the event icon turns yellow. The event will also be activated automatically as soon as you change its configuration. New events, defined in the current task are automatically active.

Note



You may change the configuration of an existing event, which is not written originally by the current task. These changes will be stored on the *ibaHD-Server* and will apply to all other clients currently using this event. If an *ibaPDA* client is currently writing to this event, the changes only take effect in the *ibaPDA* client if the data acquisition is restarted. If a change is made to the event name, numeric or text fields, the *ibaPDA* client will reject the configuration when simply restarting the acquisition but requires to apply this update in the data storage dialog.

Therefore, it is highly recommended not to change the configuration of existing events that are being written to by other clients. This could lead to misconfigured events in other clients and also data-loss in some cases.

Generate one event occurrence per processed file

Choose this option if you want the event to be written to the HD store only once with every task execution, i.e. with every data file or HD query respectively.

Event time

The event time determines the timestamp of the created event. There are two predefined fields in the drop-down-list, "Start time" and "End time" representing the start and end time of the data file or HD query.

Furthermore, there is the possibility to use an expression or an info field defined in the PDO file instead. You can even enter an expression manually into the field. These expressions must represent the number of seconds (as floating point values) starting from the beginning of the processed data file or HD query.

Note that all signals are evaluated at the selected time. Recommendation: Use constant values whenever a specific value needs to be written.

Generate an event occurrence for every rising edge

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 digital trigger signal.

The trigger signal must be part of the data file or HD query. The signal can be a virtual signal, computed by *ibaPDA* or any other digital signal, e. g. from a PLC. It is not possible to generate the trigger signal in the analysis file.

Non-constant numeric values will be automatically averaged over the time-ranges where 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.

Select the virtual signal for trigger signal in the event configuration tab from the signal tree of the data file. The next time the task is executed each limit violation event will be written to the HD store.

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 technostrng.

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

- Numeric fields		
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

Formatting

Minimum number of digits before decimal point: 1

Number of digits after decimal point: 2


Format positive numbers: add nothing

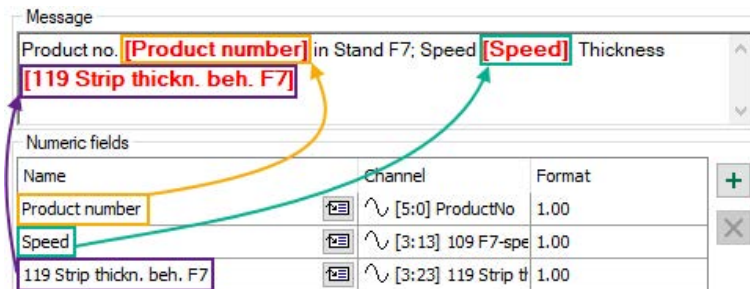
Example: 1.00
-1.00

OK Cancel




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.

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).



The screenshot shows the 'Message' editor with the text: 'Product no. [Product number] in Stand F7; Speed [Speed] Thickness'. Below the message text is a table of 'Numeric fields'.

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

Arrows indicate the process: one arrow points from the 'Product number' field to its placeholder in the message, and another points from the 'Speed' field to its placeholder.

Text fields

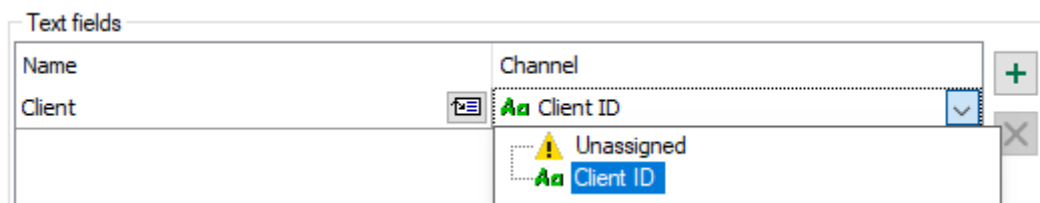
In the table, you can define up to eight placeholders for texts which can be used in the message.

Column Name




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


Column Channel

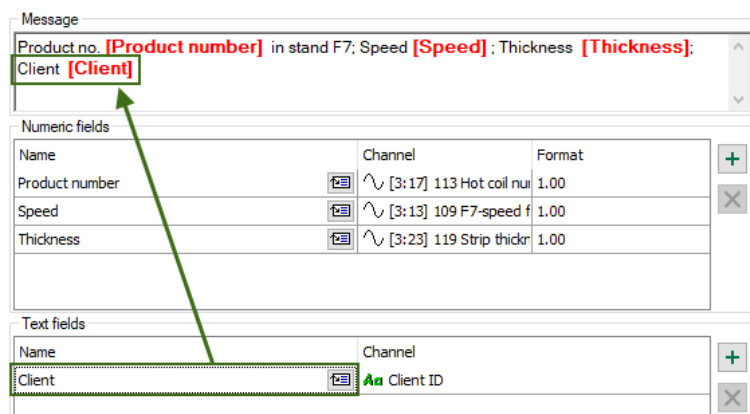
Here, you select the text signal whose current value has to be entered into the message at the triggering of the event.



The screenshot shows the 'Text fields' table with two columns: 'Name' and 'Channel'.




Name	Channel
Client	 Aa Client ID
	 Unassigned
	 Aa 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).




The screenshot shows the 'Message' editor with the text: 'Product no. [Product number] in stand F7; Speed [Speed] : Thickness [Thickness]: Client [Client]'. Below the message text are two tables: 'Numeric fields' and 'Text fields'.

Numeric fields

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 thickr	1.00

Text fields

Name	Channel
Client	 Aa Client ID

A green arrow points from the 'Client' text field to its placeholder in the message text.

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

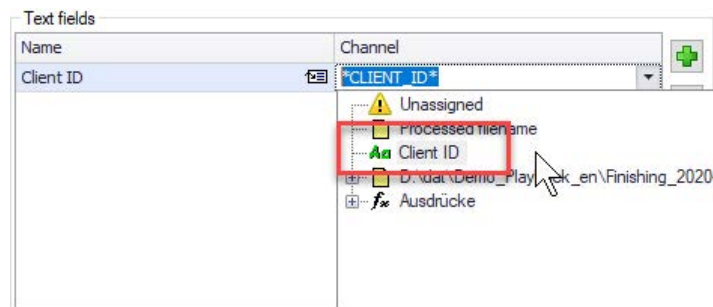
All used expressions need to be timebased! Lengthbased signals can be converted to timebased signals using the function *ConvertBase* ([lengthbased_expression],1,0).

Additional text fields for HD offline events

More items are available in the "channel" column for configuration of offline events.

Client-ID

HD offline events can be generated by different sources but stored in the same event store. In order to be able to identify the origin of an event later in the event table the so called *Client ID* can be inserted in the message text. The Client ID is available as a text signal in the signal tree in the channel column.



Message

Strip **[113 Hot coil number]** left stand F7 at speed **[Speed]**
 Client ID: **[Client ID]**

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 file

The name of the processed data file is another supplementary information which is available by default to be inserted as a text field in the message. It provides a reference to the data file which is the basis for the event.

Text fields

Name	Channel
BundNr	f* [BundNr]
Client-ID	Client ID
Processed data file	*CURRENT_FILE_NAME*

Unassigned

- Processed filename
- Client ID
- D:\dat\Demo_Playback\Fertig_704204_202
- Expressions

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

5.13.3 Client options tab

By setting the client options, you can determine how events are shown on the connected client computers in the event display and which commands are sent to the trend displays when the event occurs.

Other documentation



You'll find a detailed description of the *Client options* tab settings in the documentation of the product *ibaHD-Server*.

5.14 Kafka Publish task

With the Kafka Publish task it is possible to send computed values to any Kafka broker instance.

Note



The Kafka broker is not included in the installation and needs to be provided by the user.

Each task writes to only one topic. If values should be sent to multiple topics, multiple tasks need to be configured.

5.14.1 Connection

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 the Kafka Publish task configuration. The 'Type' is set to 'Kafka'. The 'Cluster address' is empty. The 'Cluster connection security' is set to 'PLAINTEXT'. The 'Schema registry address' is unchecked. The 'Schema registry connection security' is set to 'HTTP'. The 'Message timeout' is set to '3 s'. The 'Acknowledgment mode' is set to 'None'. The 'Additional parameters' section contains a table with 'Key' and 'Value' columns.

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.

5.14.1.1 Kafka type

The following settings are required 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').

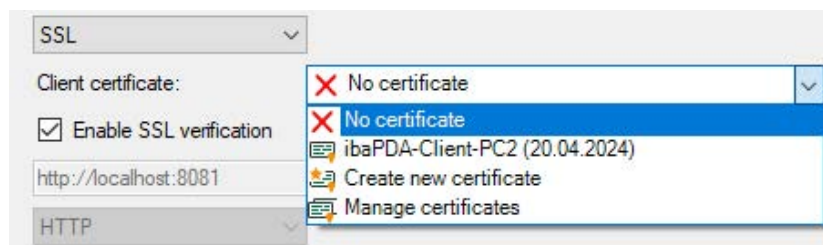
Cluster connection security

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

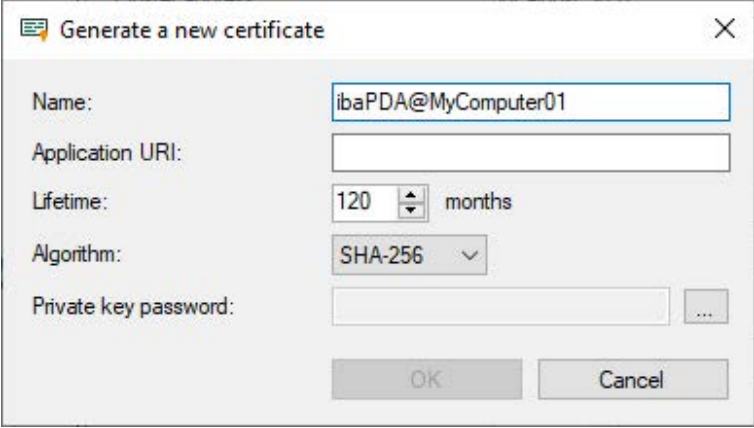
■ SSL



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.

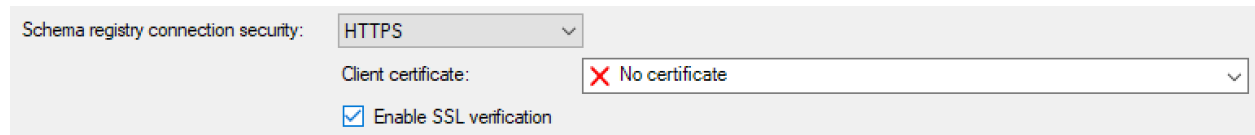
Schema registry connection security

ibaDatCoordinator supports different methods for connection security. Depending on the selected option, further entries are required.

■ HTTP

No further entry required

■ HTTPS



Schema registry connection security: HTTPS

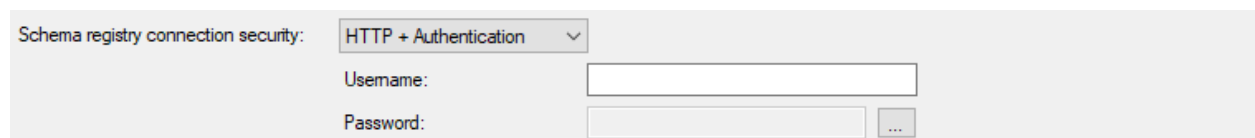
Client certificate: ✗ No certificate

☒ Enable SSL verification

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



Schema registry connection security: HTTP + Authentication

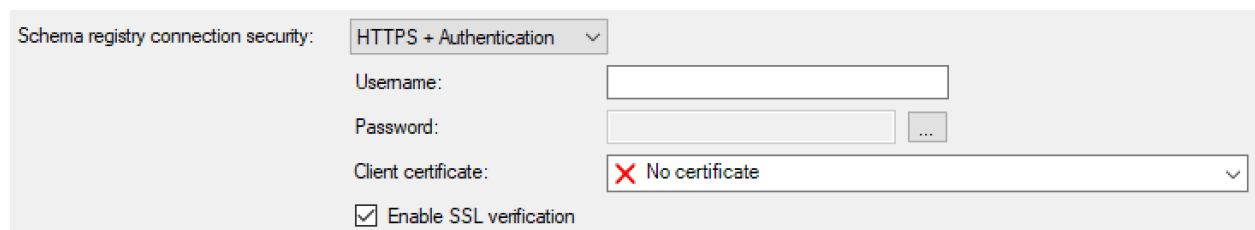
Username:

Password: ...

☐ Enable SSL verification

This method requires the input of username and password.

■ HTTPS + Authentication



Schema registry connection security: HTTPS + Authentication

Username:

Password: ...

Client certificate: ✗ No certificate

☒ Enable SSL verification

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
<input type="text"/>	<input type="text"/>



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



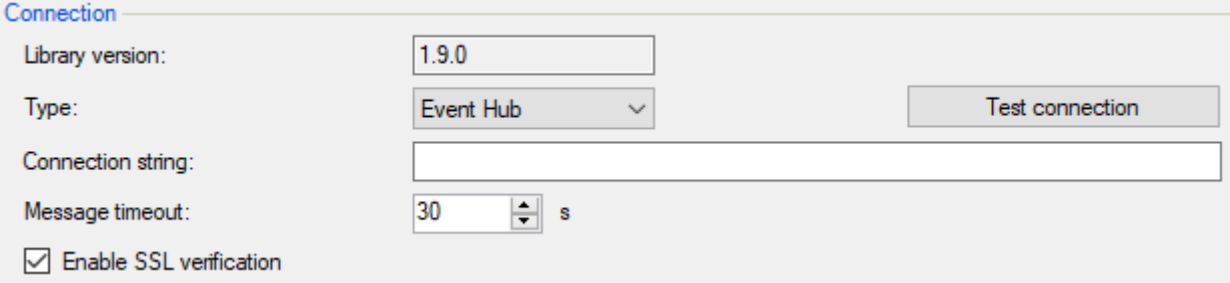
Configured parameters in the list always have priority and overrule other settings.

The documentation of the parameters available in the *librdkafka* library can be found at:

<https://github.com/edenhill/librdkafka/blob/master/CONFIGURATION.md>

5.14.1.2 Type Event Hub

The following settings are required for the **Event Hub** type:



The screenshot shows a 'Connection' dialog box with the following fields and controls:

- Library version:** A text box containing '1.9.0'.
- Type:** A dropdown menu showing 'Event Hub' with a downward arrow.
- Test connection:** A button located to the right of the Type dropdown.
- Connection string:** A large empty text box.
- Message timeout:** A spinner box showing '30' with a unit 's' (seconds).
- Enable SSL verification:** A checked checkbox.

Connection string

You can find the connection string in the properties of the Event Hub in the Microsoft Azure portal.

Message timeout

Time in seconds that *ibaDatCoordinator* waits for a response from the Event Hub.

<Test connection>




Use the <Test connection> button to test the connection to the Event Hub.

Enable SSL verification

When the SSL verification is enabled, a CA certificate is required. Usually, the "DigiCert Global Root G2" certificate is to be selected. This is the current CA root certificate. When installing *ibaDatCoordinator*, this certificate is automatically stored in the certificate store.

5.14.2 Target

In the *Target* tab you will find the configuration for the sent messages and data.

Here you can specify analysis file (PDO file) to use expressions, defined in it. The button  creates a copy of analysis file with name “*_ (ibaDatCoordinator).pdo”, where “*” is an original analysis name, and launches *ibaAnalyzer* with specified file loaded. The button  uploads the modified analysis if changes were done to the copy created with the  button.

Example .dat file

You can open a DAT file to use its data to test expression evaluation by pressing the <?> button. This file does not affect the task execution and is used only for testing. Evaluated values appear in the *Test value* column for each signal.

Topic

The Kafka topic where the messages are sent to is selected here. 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. The key for the messages can be configured using placeholders, which can e.g. be used 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.

Timestamp:	STARTTIME
UTC offset:	<div> <div>Start time</div> <div>End time</div> <div>V:\Test file.dat</div> <div>info</div> <div>Expressions</div> <div>Test expression</div> </div>

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 it to 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]

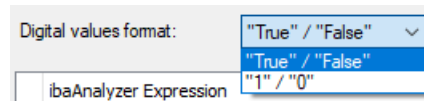
  "My signal 2": 4.56,
  [optional metadata fields of signal 2]
}
```

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 (for example predefined expressions from the PDO file). This expression will be evaluated when the job runs and the result will be published on the server. If a PDO or test-file is loaded, an *ibaAnalyzer* file tree is available to easily select expressions.

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).

The data type corresponding to the signal type must be selected in the *Data type* column: Analog (by default), Text (String) or Digital (or Boolean).

The *Test value* column is not editable. It is used to show the values of the evaluated expressions against a test DAT file. The buttons on the right of the table are used to add/copy/delete and move expressions up and down.

5.15 SNMP Publish task

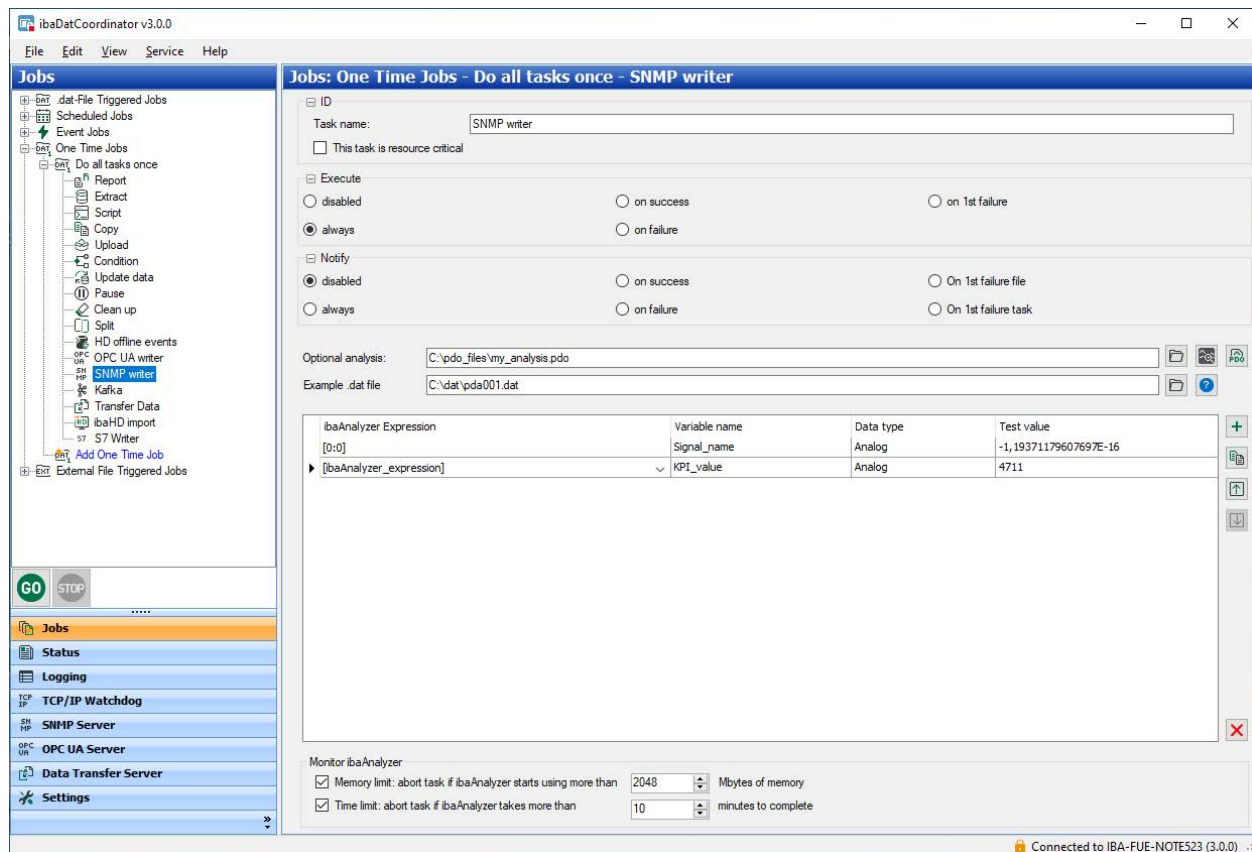
With the SNMP Publish task it is possible to publish computed values via the integrated SNMP Server.

Note






The SNMP Server needs to be enabled for this task to work. See chapter [➤ SNMP server](#), page 34

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. The following settings are available:



Optional analysis

Here you can specify analysis file (PDO file) to use expressions, defined in it. The button  creates a copy of analysis file with name “*_ (ibaDatCoordinator).pdo”, where “*” is an original analysis name, and launches *ibaAnalyzer* with specified file loaded. The button  uploads the modified analysis if changes were done to the copy created with the  button.

Example dat file

You can open a DAT file to use its data to test expression evaluation by pressing the <?> button. This file doesn't affect the task execution and is used only for testing. Evaluated values appear in the *Test value* column for each signal.

Value table

Here you can specify any valid *ibaAnalyzer* expression (for example predefined expressions from the PDO file). This expression will be evaluated when the job runs and the result will be published on the server. If a PDO or test-file is loaded, an *ibaAnalyzer* file tree is available to easily select expressions.

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 column *Data type* the data type must be selected according to the signal type: Analog (by default), text or digital (or Boolean).

The *Test value* column is not editable. It is used to show the values of the evaluated expressions against a test DAT file. The buttons on the right of the table are used to add/copy/delete and move expressions up and down.

5.16 OPC UA Publish task

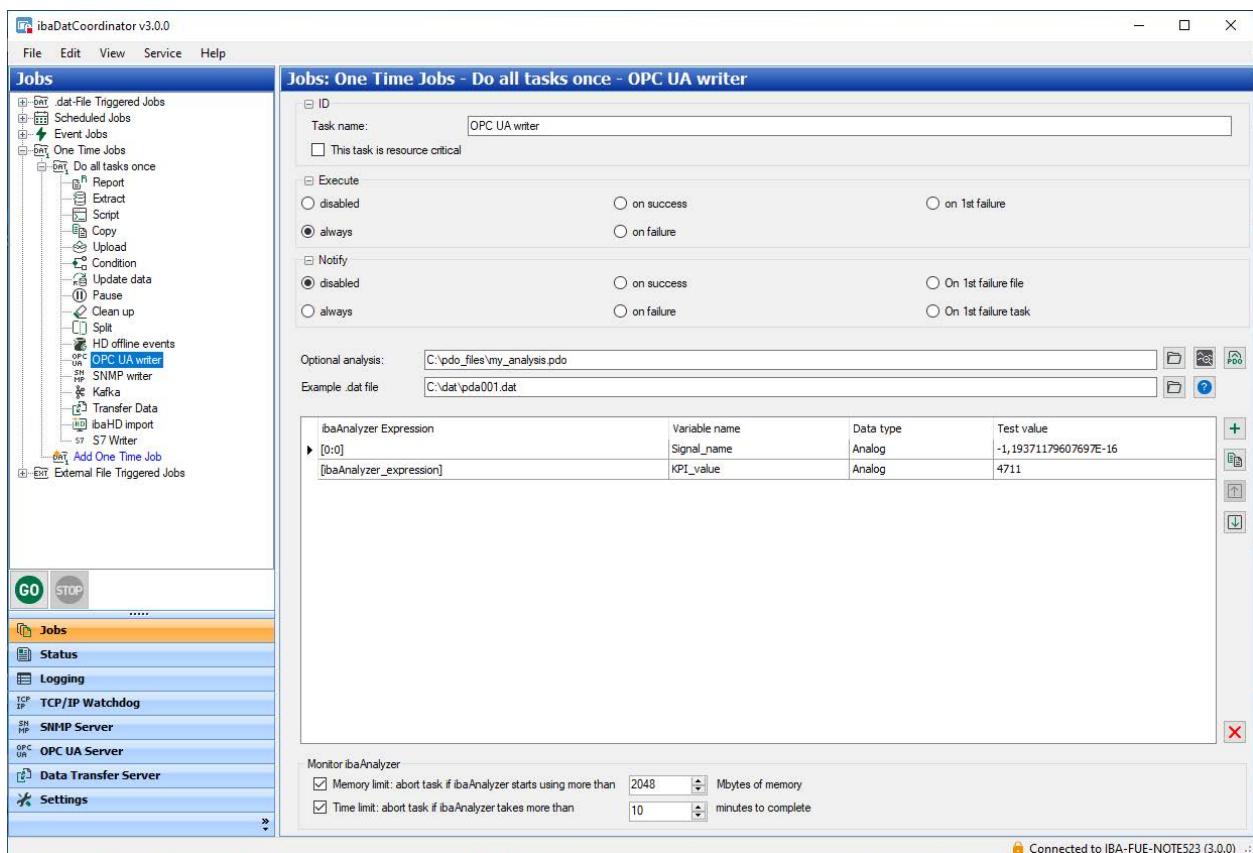
With the OPC UA Publish task it is possible to publish computed values via the integrated OPC UA Server.

Note






The OPC UA Server needs to be enabled for this task to work. See also chapter [↗ OPC UA Server](#), page 40.

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. The following settings are available:



Optional analysis

Here you can specify analysis file (PDO file) to use expressions, defined in it. The button  creates a copy of analysis file with name “*_ (ibaDatCoordinator).pdo”, where “*” is an original analysis name, and launches *ibaAnalyzer* with specified file loaded. The button  uploads the modified analysis if changes were done to the copy created with the  button.

Example dat file

You can open a DAT file to use its data to test expression evaluation by pressing the <?> button. This file doesn't affect the task execution and is used only for testing. Evaluated values appear in the *Test value* column for each signal.

Value table

Here you can specify any valid *ibaAnalyzer* expression (for example predefined expressions from the PDO file). This expression will be evaluated when the job runs and the result will be published on the server. If a PDO or test-file is loaded, an *ibaAnalyzer* file tree is available to easily select expressions.

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).

The data type corresponding to the signal type must be selected in the *Data type* column: Analog (by default), text or digital (or Boolean).

The *Test value* column is not editable. It is used to show the values of the evaluated expressions against a test DAT file. The buttons on the right of the table are used to add/copy/delete and move expressions up and down.

5.17 Transfer Data task

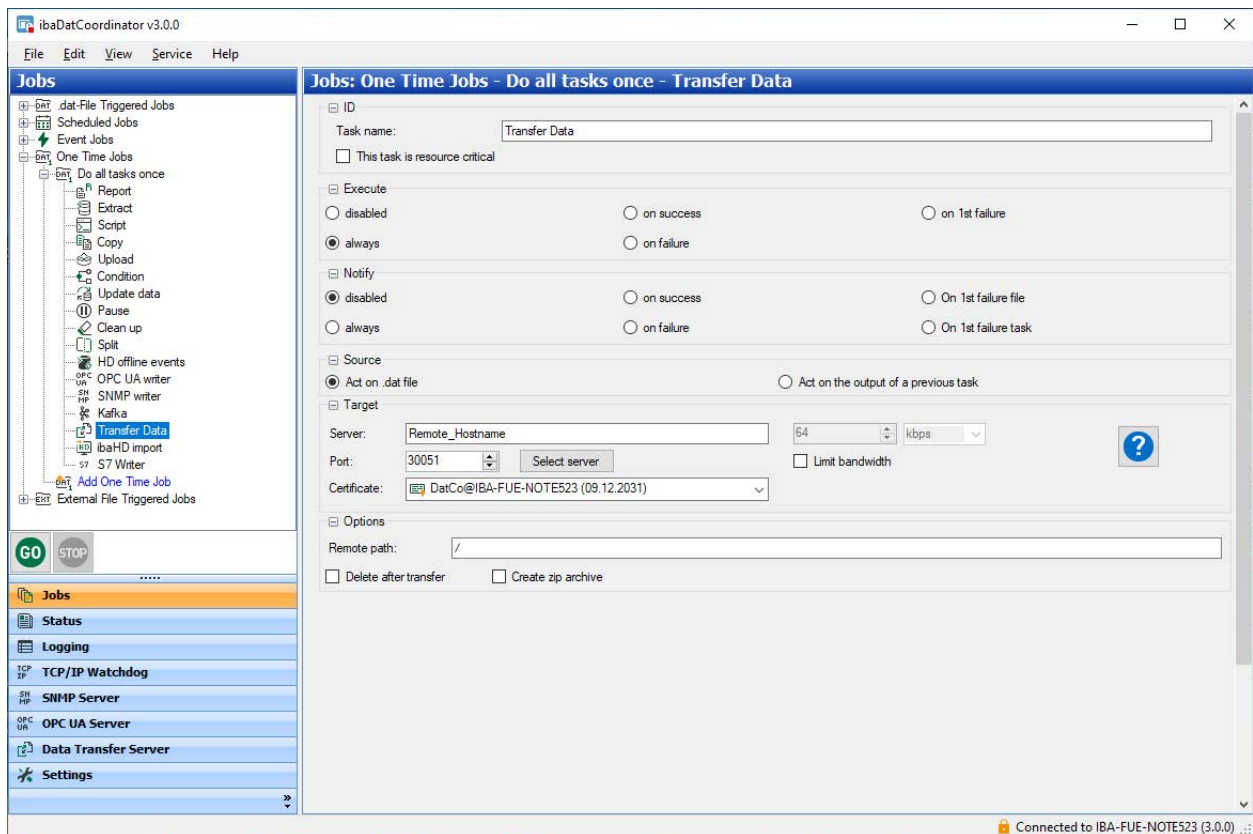
The "Transfer Data" task can be used to transfer files to a remote *ibaDatCoordinator* instance. The task can transfer either the currently processed DAT file or the result of a previous task (e.g. a PDF file).

Note

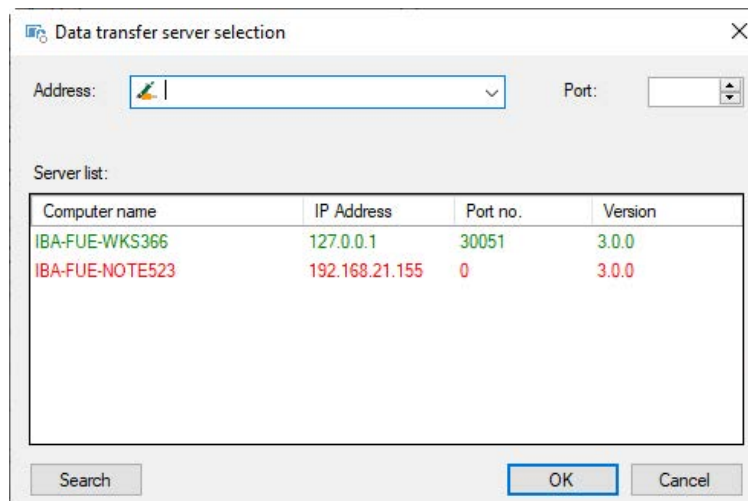


The *ibaDatCoordinator* data transfer server must be activated on the receiving system. See also chapter [Data Transfer Server](#), page 29.

The server name can be entered manually in the *server* field or alternatively the *Data transfer server selection* dialog can be opened by clicking the <Select server> button.



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



When you have selected the desired server, you can double-click the corresponding entry or click <OK> to apply the settings in the client. The <Search> button refreshes the server list.

With the *Limit bandwidth* checkbox you can set the maximum transmission speed. The desired limit can be entered in the numeric field. The unit "mbps" or "kbps" can be selected from the drop-down menu.

The *Certificate* selection can be used to select a certificate if encryption of the connection has been configured on the remote side.

Selecting the *Manage certificates* option will take you directly to the *Settings* tab. Existing certificates can also be imported or exported there in the certificate management. See also chapter [↗ Certificates](#), page 49.

Note

The same certificate must be used on both the data transfer client (in the transfer data task) and the data transfer server. When exporting the certificate, it is not necessary to export the private key. If you do not want encryption, the "No certificate" option must be selected on both sides.

In the *Remote path* field you can select a file path where the data will be stored on the data transfer server.

Check the *Delete after transfer* checkbox if the file should be deleted after processing on the client.

Using the *Create zip archive* checkbox you can select whether the files should be converted into a ZIP archive before transferring them.

5.18 Merge task

You can use the merge task to merge several DAT files into one DAT file. Based on a common expression which needs to evaluate to the same value for all files, multiple DAT files can be opened in parallel and merged into a new file.

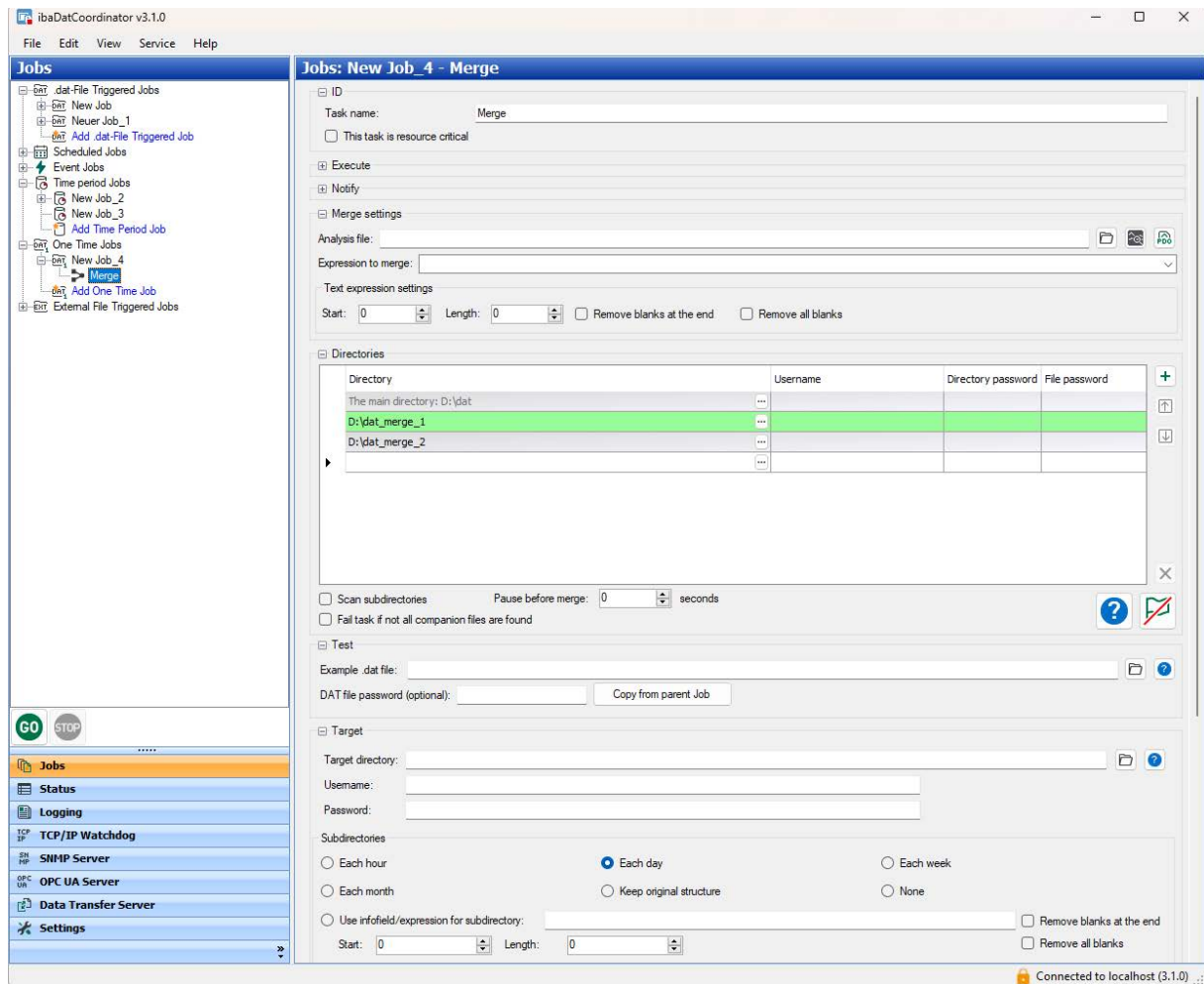
This task requires a license *ibaDatCoordinator-File-Extract* or *ibaAnalyzer-File-Extract*.

The task is triggered by a new DAT file in a configured folder. The *Expression to merge* is evaluated for this file and then *ibaDatCoordinator* starts looking for suitable companion files in the configured directories which evaluate the same expression to the same value.

Example:

For the multiple steps of a steel production line, multiple DAT files are available which contain a "CoilID" infofield. Consequently, for all files belonging to the same coil, an expression of the form *InfoField(0, "CoilID")* evaluates to the same value and can be used as merge expression.

Only files that are available when the task is triggered can be merged. It is recommended to use the files that are expected to be finished last as triggers for the job configuration or to apply the *Pause before merge* option.



Analysis file

Enter here the PDO file containing the extract configuration. Enter the path name and file name of the PDO file or select the file using the browser button. For creating this analysis, 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. The expression can also be defined 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 the start and length of the character string.

Directories

Select the directories from which the files to be merged are taken from. The main directory in the first line is 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 rows. The rows can also be moved using drag & drop. New rows can be created by modifying the empty row at the bottom of the table.

If the directories or files are password-protected, you still need to enter the user name and password.

Scan subdirectories

If the option is 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 the option is activated, the task fails if one or more files are missing. If the option is deactivated, the merge is done without the missing files and a warning is logged.

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 use a test file to test the merge expression. Select a DAT file and click the <?> button to test the expression. If the DAT file is password-protected, you still need to enter the password.

Configuration steps

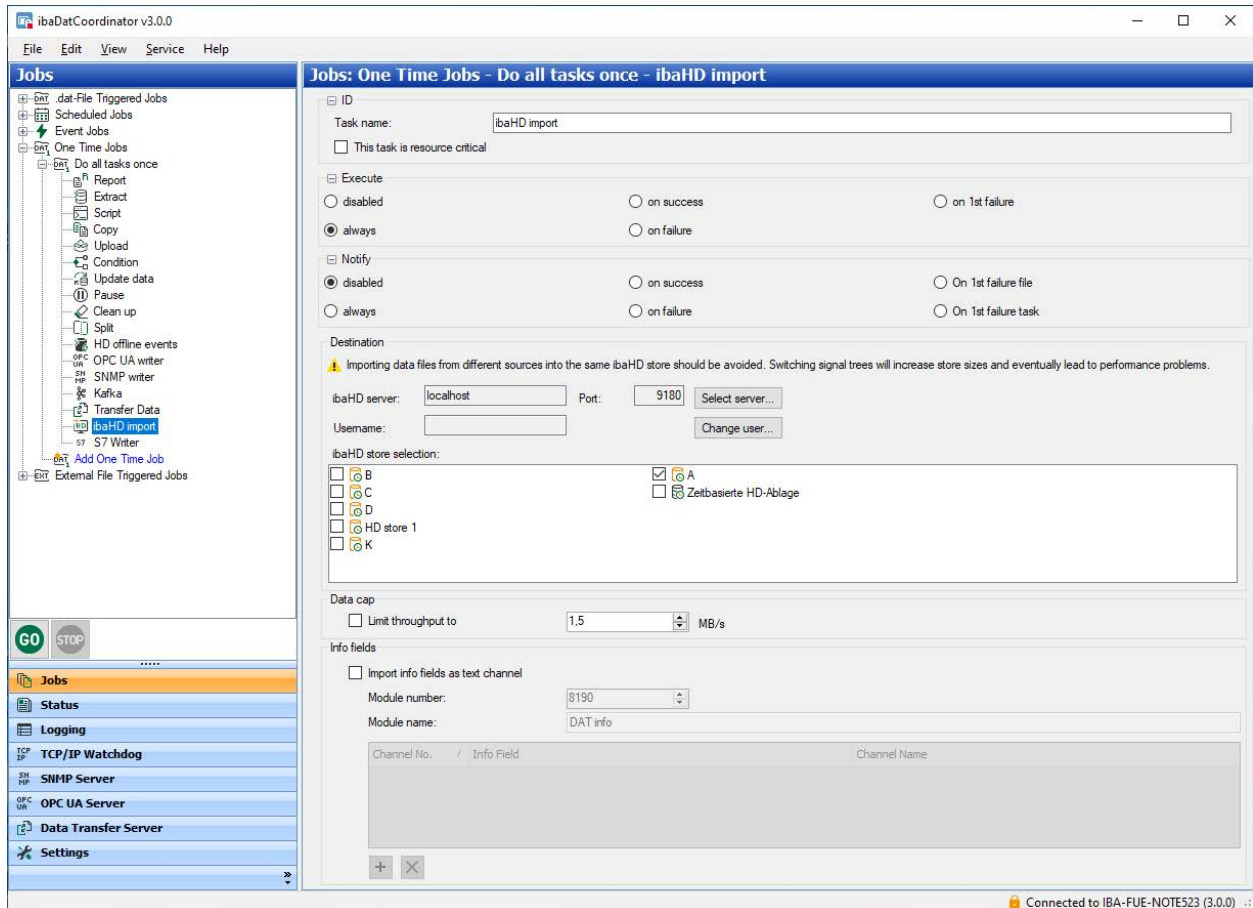
For configuring the merge task, proceed as follows:

1. Select the analysis file (PDO) that contains the configuration for the extraction by specifying the directory or search using the  button.
2. Then, you can click on the *ibaAnalyzer* symbol to check whether the correct analysis file has been loaded. The analysis file is opened in *ibaAnalyzer* and can also be edited here.
3. Select the expression that is to be evaluated for merging. The expression can also be defined in the analysis rule.
4. Select the directories from which the files to be merged are taken from.
5. Configure the target directory of the processed file.
6. If the target directory is located on a remote server, enter a user name and a password in order to gain access to the remote server.

You can test the expression by choosing a DAT file. This file can be selected via the browser button or the path. Once you have selected the test file, click <?> in the *Test* area to start the test.

5.19 ibaHD import task

With the ibaHD 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.



The settings specific to the ibaHD import can be adjusted in the following sections:

Destination

Here you can directly enter the IP address or the name of the HD server and the timebased HD store as a destination for the DAT file or select it from the drop-down menu. The table lists all *ibaHD-Servers* detected in the network. Click the <Search> button to update the list.

Data cap

The transfer rate can be limited. This may be important if data collection is also active at the same time in the target HD store. Enable this option if necessary and enter a maximum limit for the transfer rate in MB/s.

Info fields

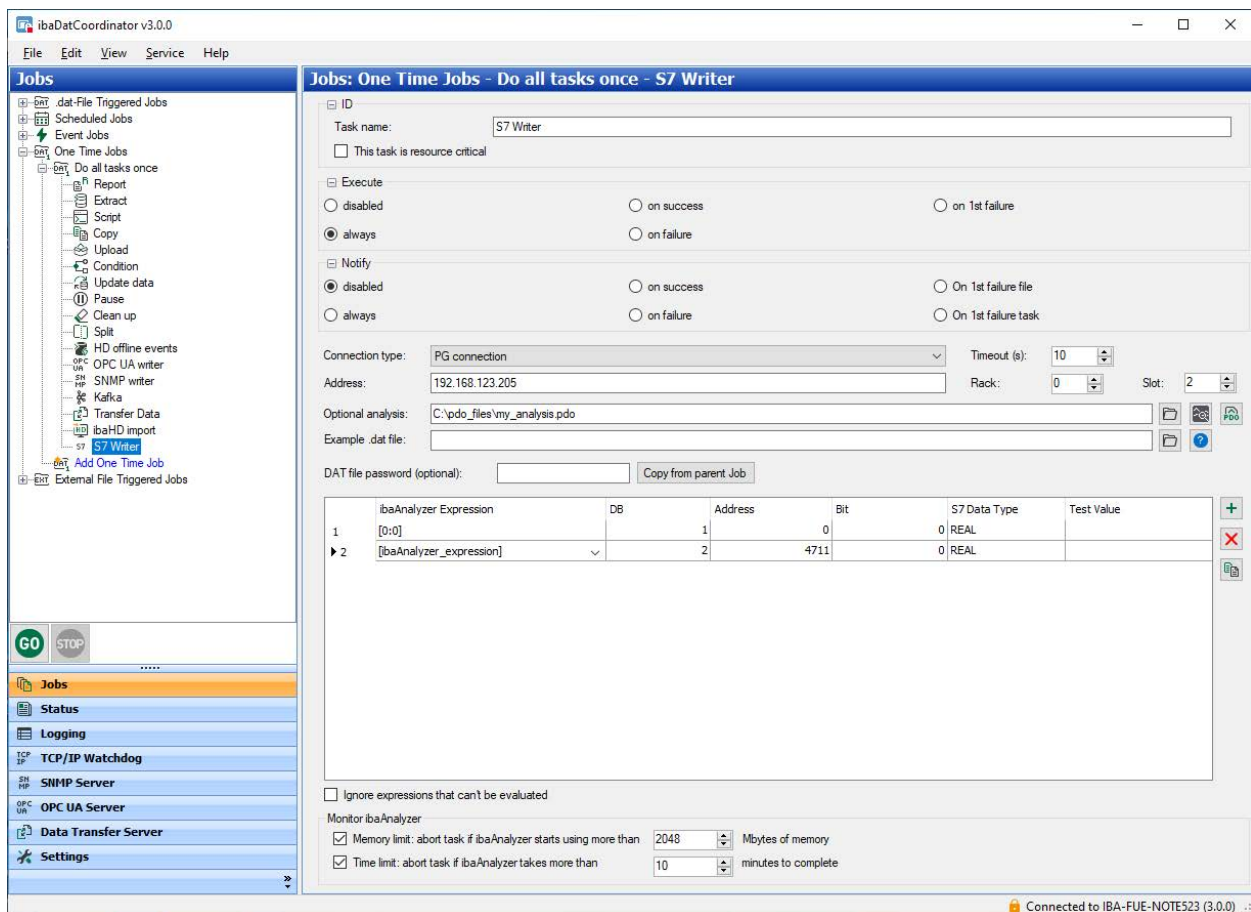
Enter the info fields which are to be imported as a text channel. Enter a module number and a module name.

5.20 S7 writer task

The use of the S7 writer task is subject to licensing. With the S7 writer task, you can extract or calculate data from an iba DAT file and write this data to data blocks (DBs) within an S7 PLC. In this task, *ibaAnalyzer* expressions can be defined using S7 operands. When executing the task, these expressions are evaluated and sent to the S7.

Prerequisite:

- S7 PLC for writing the data
- Connection to the S7 PLC via TCP/IP



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. You can reference these expressions in the expressions for the S7 signals. This enables simpler expressions for the S7 signals which reference expressions in the analysis directly.

In addition to the text field, where you can enter the name of the analysis, there is a button you can use to navigate to an analysis. If the path to the analysis file is correct, you can open the analysis file in *ibaAnalyzer* by clicking the *ibaAnalyzer* button.

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. 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 DB to which the data will be written.
- Address: The address within the DB 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: Data type in which the data will 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 this option is not checked, the task is not executed if one of the expressions cannot be evaluated. If the option is evaluated, the signal whose expression cannot be evaluated is not written. The other signals with valid expressions are written. If there are no expression that can be evaluated, the task is canceled.

Test *ibaAnalyzer* expressions

The *ibaAnalyzer* expressions in the S7 signal table can be tested using a test DAT file. Enter the path for the test file into the text field "Example dat-file" or select a file using the browse button. If the specified DAT file exists and can be accessed, you can click the test button. Otherwise, the button will be grayed out.

After performing the test, the "Test value" column will be filled 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.

Error message	Possible cause/remedy
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.
You have installed the version x.y.z of <i>ibaAnalyzer</i> . The minimum required version is 6.5.0, 6.7.0 or later (x.y.z stands for your current version of <i>ibaAnalyzer</i>)	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 later than or the same as 6.5.0, but older than 6.7.0, this means that the task was executed successfully, but there may be a loss in precision. (Only floating point numbers with simple precision are used.)
Could not evaluate expression for operand X (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 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.

6 Appendix

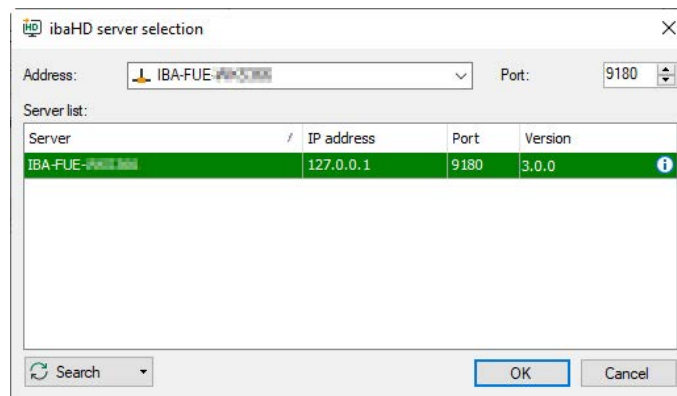
6.1 ibaHD server connection and user management

The jobs “Scheduled job” and “Event job” as well as the “ibaHD import” task require a connection to an ibaHD server.

The HD connection pane always shows the current ibaHD server and port, the currently logged in user to the ibaHD server and the available stores.

In order to set up a connection to an ibaHD server follow these steps.

Click on the <Select server...> button to open the selection dialog for the ibaHD server.



The table lists all ibaHD servers which are active in the network. If the requested ibaHD server is not in the list, try the <Search> button.

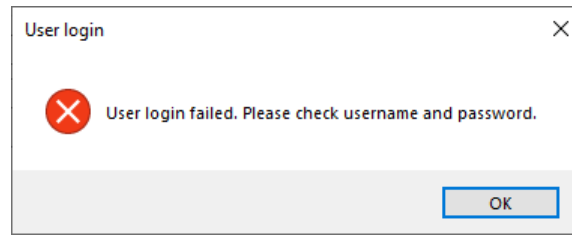
If the requested ibaHD server is in the list, select it. The line is highlighted in green if *ibaDatCoordinator* can establish a connection to the ibaHD server. Close the dialog with <OK>.

If, even after a search the ibaHD server does not show up, enter the IP address or the host name of the ibaHD server computer in the address field and click <OK>.

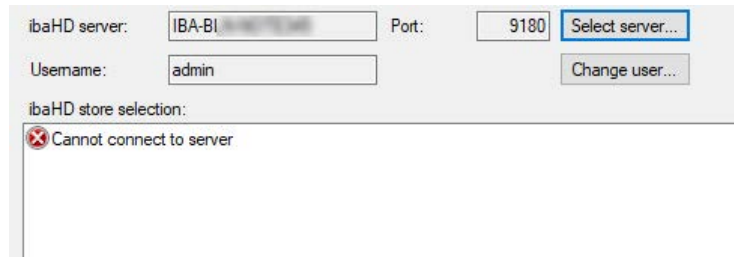
If the user management is enabled on the ibaHD server you should then enter or select a user name and enter the correct password.



If the password is not correct, an error message appears.




If the entry of the user credentials is not correct or if the connection to the ibaHD server cannot be established for some other reason, another error message appears instead of the HD stores.

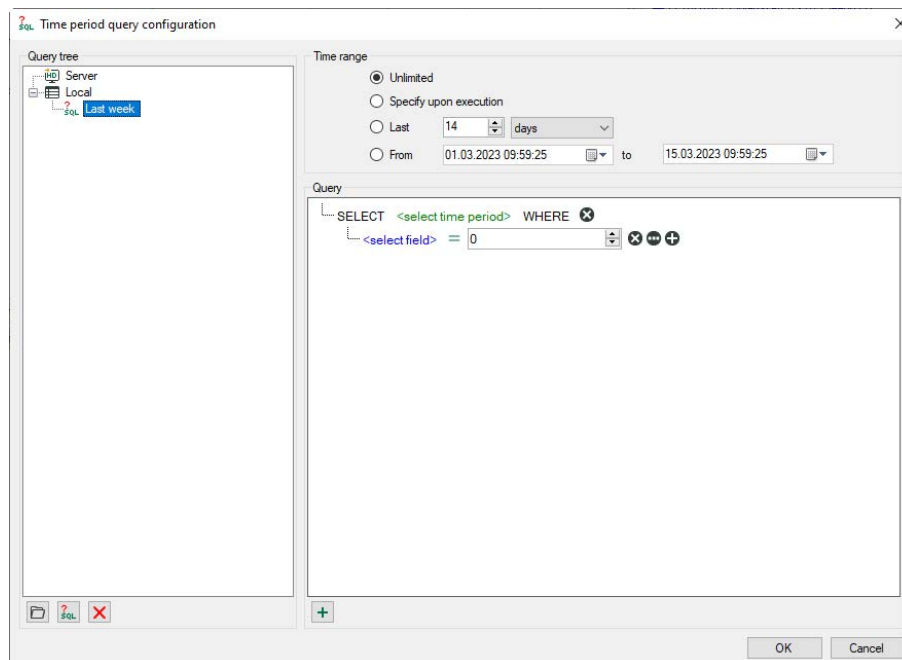


Check the configured users in the ibaHD Manager application and create an appropriate user oder set a new password, if required.

6.2 Configuration of queries




In the toolbar of the HD time periods table, you can filter the rows displayed by certain criteria. Enter a value or configure a query by clicking on the icon. If you remove a value, the filter is deleted.

Click on the  icon in the toolbar for the time periods table to configure queries.



Configuring queries

All existing queries are shown in the query tree. You can arrange queries by using a directory structure.

	Adds a new folder at the selected position
	Adds a new query at the selected position
	Deletes the selected node and eventually secondary nodes

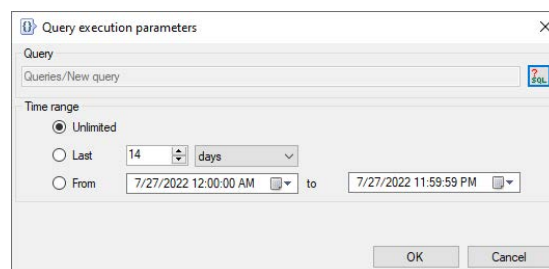
There are two types of queries:

- *Server queries* are saved on the *ibaHD-Server*, which means that all connected clients can use the same server queries.
Note: Only users who have the right *Edit ibaHD queries* can save or edit server queries.
- *Local queries* are saved in the layout. Other connected clients cannot see these.

Query range

Every query is executed over a defined time range. There are several options for restricting the query range.

- **Unrestricted:** The query covers the entire HD recording
- **Specify upon execution:** After clicking on the start button for the query, a dialog appears in which you have to specify the query range



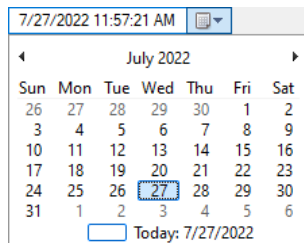
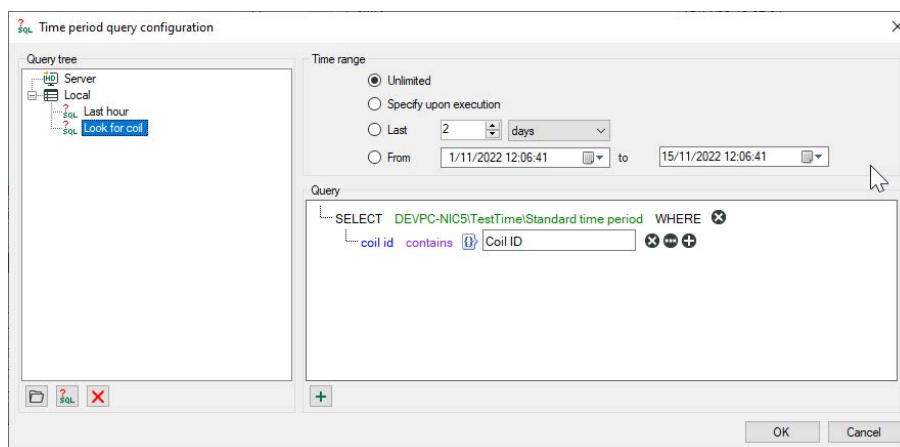
Clicking on <OK> executes the query.

- **The last xx minutes/hours/days/weeks/months:** The query only covers a fixed period of time from the start of the query, e.g. the last 8 hours for a shift overview.
- **From *Date* to *Date*:** The query covers the time period between the set days (inclusive).

Tip

To set the current date, open the Date & Time Picker by clicking on the down arrow.

Here you can select the entry under “Today...”.

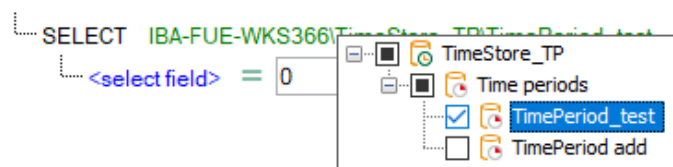
**Query details**

Here, you create the actual query using notation similar to SQL.

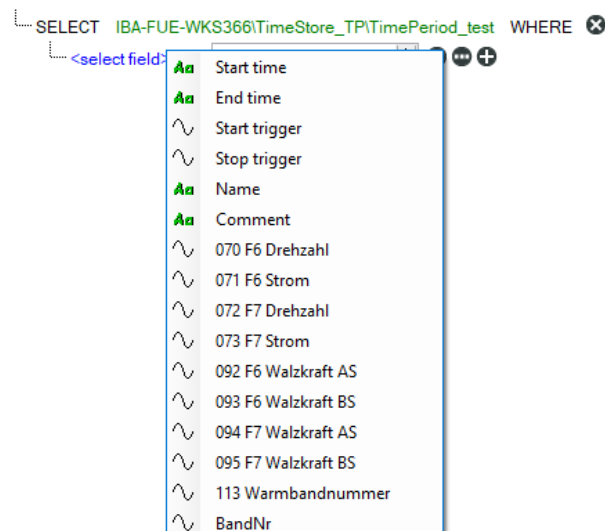
The query begins with a SELECT statement.

```
SELECT <select time period> WHERE
<select field> = 0
```

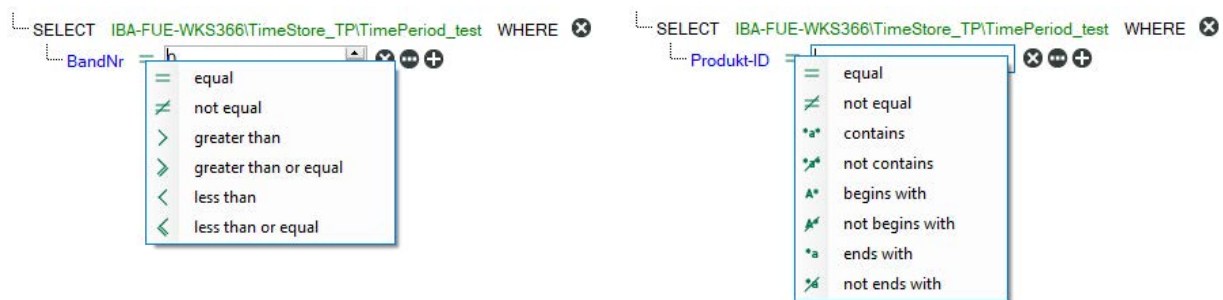
Clicking on the green text opens the tree structure for the configured time periods. Select the time period you want to search for.



Click on the blue text to select a column in the table or any existing information fields.



Depending on the selected field (numeric or text field), clicking on the green equals symbol displays a list of operators. Select the required operator.



Behind the elements in the query, there are different buttons which you can use to extend or modify the query:

: Deletes the expression

: Switches the right side of an expression between a fixed value and a field

: Adds the statement for a new expression

You can change the logical operator that is linked with a group of expressions by clicking on the operator.



Clicking on <OK> closes the editor and saves the created queries.

Executing the query

First select the relevant query from the list in the toolbar and then click on the button to the right.



When the query has been completed, the number of results found is displayed in the status bar on the right.

A screenshot of the query results window. The window title is 'HD time period table 1'. The toolbar shows the 'Run' button and a dropdown menu with 'Bandnr' selected. The table has five columns: 'Name', 'Start time', 'End time', 'InfoField...', and 'InfoField_Produ...'. The table contains 40 rows of data, each with a unique ID (e.g., pda1836_094360), a start time, an end time, and two info fields. At the bottom right, a status bar shows '40' and a 'more...' link.

Name	Start time	End time	InfoField...	InfoField_Produ...
pda1836_094360	2/22/23 12:09:13 PM	2/22/23 12:10:46 PM		094360
pda1837_027460	2/22/23 12:11:03 PM	2/22/23 12:12:00 PM		027460
pda1838_081800	2/22/23 12:12:18 PM	2/22/23 12:13:09 PM		081800
pda1839_013160	2/22/23 12:13:14 PM	2/22/23 12:14:49 PM		013160
pda1840_093120	2/22/23 12:14:59 PM	2/22/23 12:15:45 PM		093120
pda1841_094360	2/22/23 12:15:59 PM	2/22/23 12:17:33 PM		094360
pda1842_027460	2/22/23 12:17:49 PM	2/22/23 12:18:47 PM		027460
pda1843_081800	2/22/23 12:19:04 PM	2/22/23 12:19:56 PM		081800
pda1844_013160	2/22/23 12:20:01 PM	2/22/23 12:21:35 PM		013160
pda1845_093120	2/22/23 12:21:45 PM	2/22/23 12:22:31 PM		093120
pda1846_094360	2/22/23 12:22:45 PM	2/22/23 12:24:19 PM		094360
pda1847_027460	2/22/23 12:24:35 PM	2/22/23 12:25:33 PM		027460
pda1848_081800	2/22/23 12:25:50 PM	2/22/23 12:26:42 PM		081800
pda1849_013160	2/22/23 12:26:47 PM	2/22/23 12:28:22 PM		013160
pda1850_093120	2/22/23 12:28:31 PM	2/22/23 12:29:17 PM		093120
pda1851_094360	2/22/23 12:29:32 PM	2/22/23 12:31:05 PM		094360
pda1852_027460	2/22/23 12:31:22 PM	2/22/23 12:32:19 PM		027460
pda1853_081800	2/22/23 12:32:37 PM	2/22/23 12:33:28 PM		081800
pda1854_013160	2/22/23 12:33:33 PM	2/22/23 12:35:08 PM		013160
pda1855_093120	2/22/23 12:35:18 PM	2/22/23 12:36:04 PM		093120
pda1856_094360	2/22/23 12:36:18 PM	2/22/23 12:37:52 PM		094360
pda1857_027460	2/22/23 12:38:08 PM	2/22/23 12:39:06 PM		027460
pda1858_081800	2/22/23 12:39:23 PM	2/22/23 12:40:15 PM		081800
pda1859_013160	2/22/23 12:40:20 PM	2/22/23 12:41:54 PM		013160
pda1860_093120	2/22/23 12:42:04 PM	2/22/23 12:42:50 PM		093120
pda1861_094360	2/22/23 12:43:04 PM	2/22/23 12:44:38 PM		094360
pda1862_027460	2/22/23 12:44:44 PM	2/22/23 12:45:50 PM		027460

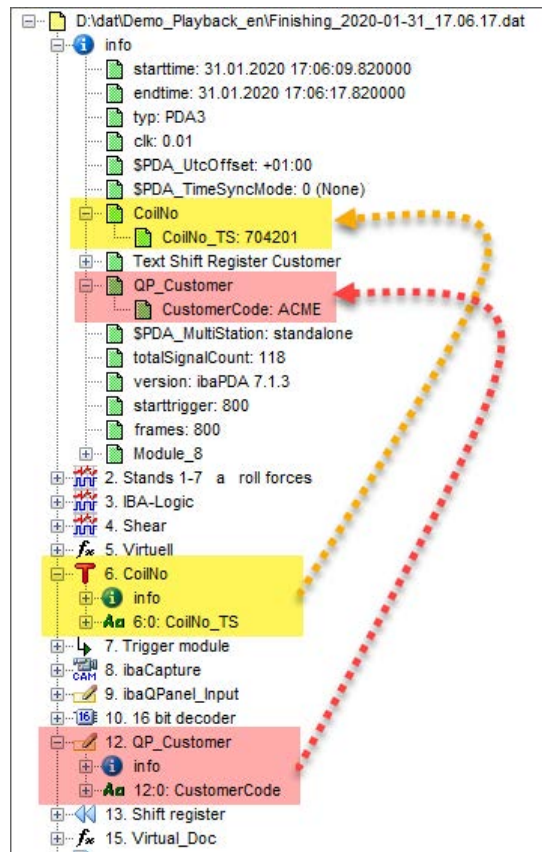
A query can return more results than the number of rows configured in the table. To display any other results, click on the blue “More...” link.

You can return to live mode by clicking on the  button.

6.3 Using infofields for directory names/filenames

Example

In a report task the names of both the subdirectories and the output files should be created in dependence of certain infofields.



The information for the infofields has derived from text signals:

- 'CoilNo_TS' from a text creator module
- 'CustomerCode' from a Qpanel input

Note



If the contents of text signals should be available as infofields in the data file, the text signals in question must be selected in the *ibaPDA* data store configuration, node Files - Infofields. Because an infofield can only have one value, the value of the text signal is written into the infofield either at beginning or at end of the data file.

The contents of the infofield `QP_Customer.CustomerCode` should be used for name of the subdirectory.

For the name of the output files, two infofields from the data file should be used:

- Customer code (`QP_Customer.CustomerCode`)
- Coil number (`CoilNo.CoilNo_TS`)

The two parts of information should be separated by an underscore in the final filename.

Sample: `CustomerCode_CoilNo.pdf`

Settings

In *ibaDatCoordinator*, you must enter the following:

E. g. entry for subdirectories

E. g. entry for output files

```
ConcatText ( InfoFieldText ( 0, "QP_Customer.CustomerCode" ),
"_" , InfoField ( 0, "CoilNo.CoilNo_TS" ) )
```

Tip



Use the expression builder in *ibaAnalyzer*, to create a correct expression.

1. Open *ibaAnalyzer* and load an appropriate data file.
2. Open the expression builder either in the signal table (add signal) or in the logical signals dialog (use a new, empty expression).
3. Enter the expression you want and click <OK>.
4. The final expression is displayed in the new signal line or the new logical signal respectively.
5. Mark the complete expression and copy it to the clipboard.
6. Return to *ibaDatCoordinator* and paste the expression from the clipboard into the entry field.

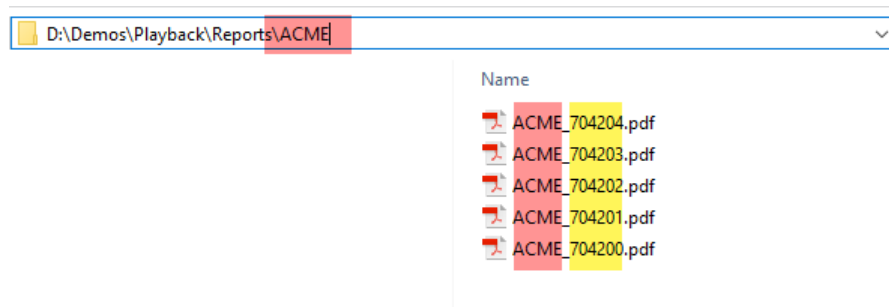
Result

Result in the file system after the job has been executed:

Subdirectories distinguished by customer name in the target folder for reports



Report output files with filenames derived from infofields



6.4 Configure ibaDatCoordinator service for other user

The following describes the action to run the *ibaDatCoordinator* service under a different local account, taking into account the interaction between *ibaDatCoordinator* and *ibaAnalyzer*.

The names used in the examples are exemplary. You can assign names according to your rules.

6.4.1 Server side

6.4.1.1 Create a managed service account

On the DC, the new service account must first be created.

This requires a PowerShell console with administrator privileges running the following.

```
Add-KdsRootKey -EffectiveTime ((get-date).addhours(-10)) -Verbose
```

```
New-ADServiceAccount svc_iba -DisplayName "iba Software Service" -DNSHostName svc_iba.excorp.local
```

```
Set-ADServiceAccount svc_iba -PrincipalsAllowedToRetrieveManagedPassword WKS1$
```

Example *ibaDatCoordinator* account:

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> Add-KdsRootKey -EffectiveTime ((get-date).addhours(-10)) -Verbose
VERBOSE: Performing the operation "Add-KdsRootKey" on target "W2K16-TD1.ibatest.local".

Guid
-----
74608809-

PS C:\Users\Administrator> New-ADServiceAccount svc_datco -DisplayName "ibaDatCoordinator Service" -DNSHostName svc_datco.ibatest.local
PS C:\Users\Administrator> Set-ADServiceAccount svc_datco -PrincipalsAllowedToRetrieveManagedPassword win10-td1$
PS C:\Users\Administrator>
```

This allows the new service account to be used on the WKS1 computer. If, in addition, it is to be used on computer WKS2, the last command must be repeated with **WKS2\$** instead of **WKS1\$**.

Command	Description
<code>Add-KdsRootKey</code>	Creates a new root key for the Microsoft Group Key Distribution Service (KdsSvc) and sets the date from which this key is valid to the current date minus 10 hours.

Command	Description
<code>New-ADServiceAccount</code>	Creates a new managed service account in the Active Directory named "svc_iba", sets the display name to a comprehensible value and defines the DNS entry for the service account to <service-name>.<domain-name>.local
<code>Set-ADServiceAccount</code>	Adds the system named "WKS1\$" to the members of the service account "svc_iba" and thus enables use of the account on the system.

In order to be able to assign permissions more granularly, it is recommended to create separate service accounts for each of the software products.

Examples for ibaDatCoordinator and ibaCapture:

- ibaDatCoordinator: svc_ibaDatCo
- ibaCapture: svc_ibaCapture

6.4.2 Client side

First install the required Software components

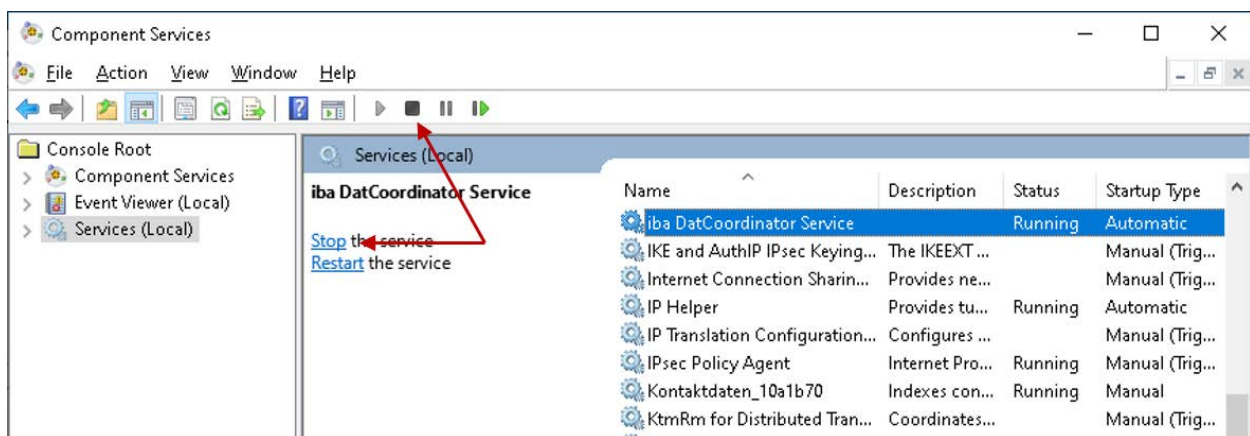
- *ibaDatCoordinator*
- *ibaAnalyzer (x64)*

6.4.2.1 Changing the service account

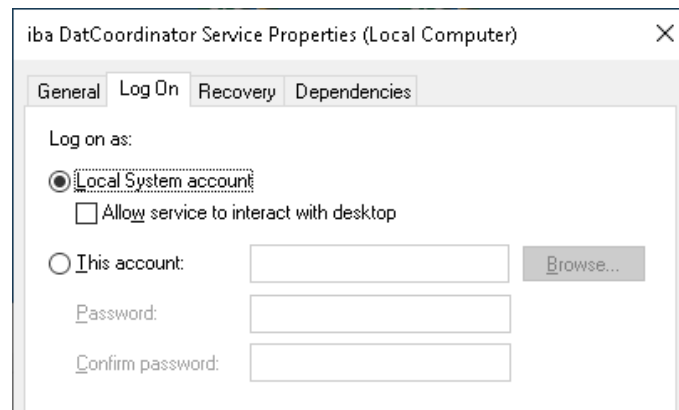
Open the start menu and enter the following command:

`dcomcnfg`

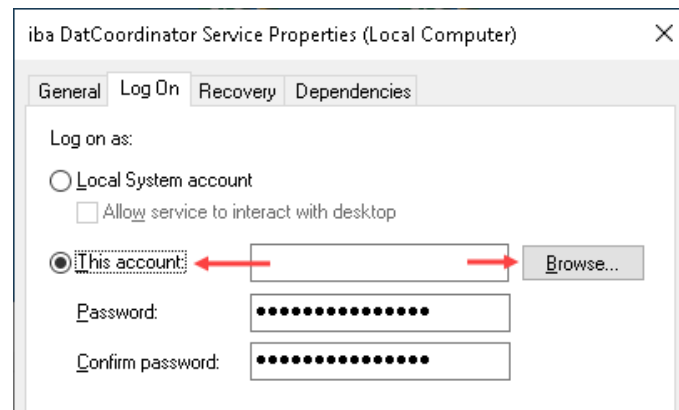
Switch to *Services* and select the service „iba DatCoordinator Service“ and stop it with a click on „Stop“.



Now open the properties of the service and select the tab *Log On*.



Select *This account* and click on <Browse...>.



In dialog *Select User or Service Account*, first click on <Advanced...> and then change the search path by clicking on <Paths> to „Entire Directory“ or the specific domain.

In addition, change the object type (click on <Object Types...>), so that only “Service Account” is selected.

Then click on <Find Now> and select the created service account „svc_datco“ from the search results.

Select User or Service Account

Select this object type:
Service Account Object Types...

From this location:
Entire Directory Locations...

Common Queries

Name: Starts with Columns...

Description: Starts with Find Now

☐ Disabled accounts

☐ Non expiring password

Days since last logon: Stop

Search results:

Name	In Folder
svc_datco	ibatest.local/Managed Service Accounts

OK Cancel

Select User or Service Account

Select this object type:
Service Account Object Types...

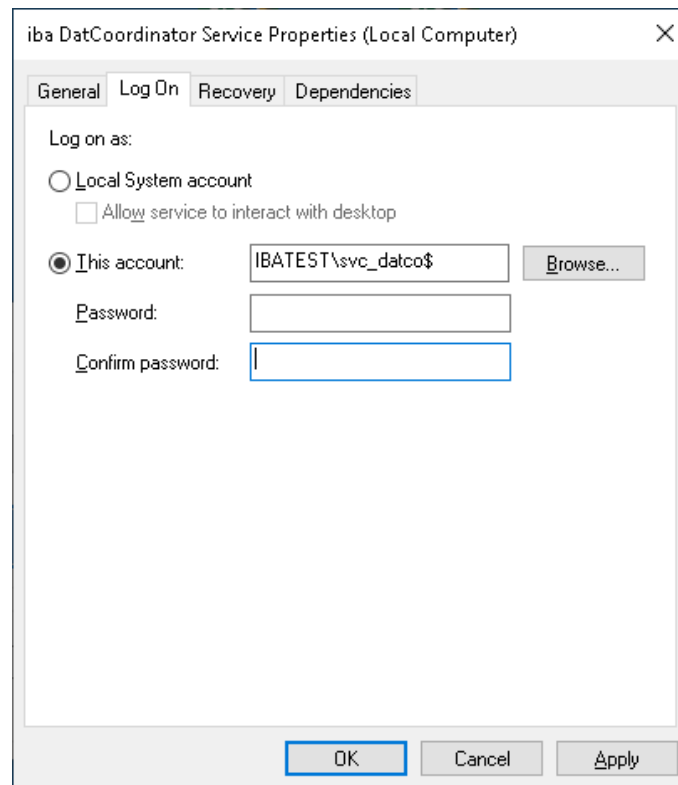
From this location:
Entire Directory Locations...

Enter the object name to select (examples):
svc_datco Check Names

Advanced... OK Cancel

Exit the dialogs *Select user or Service Account* with <OK> until you are back in the properties of the service.

Delete the contents of the preset password fields and exit the dialog with <OK>.



Now start "iba DatCoordinator Service".

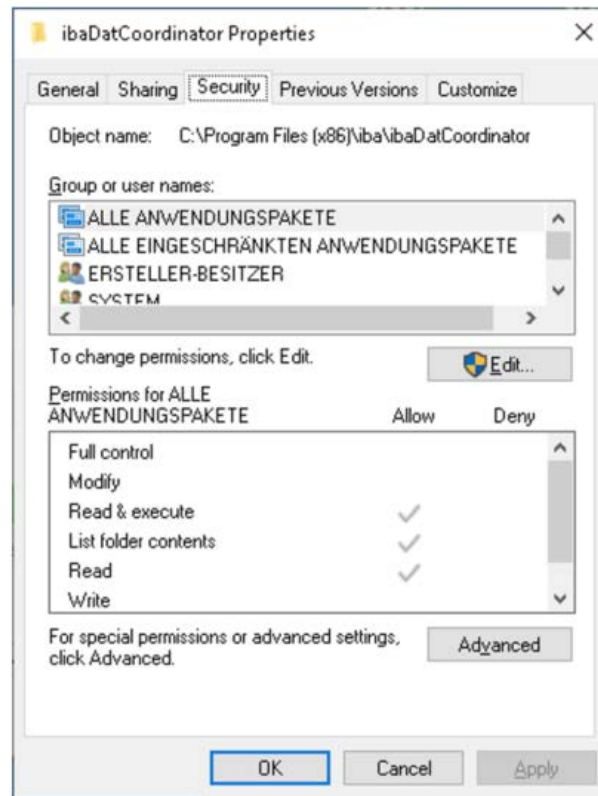
6.4.2.2 Setting permissions

6.4.2.2.1 ibaDatCoordinator directory

Since the „C:\Program Files (x86)“ and „C:\Program Files“ directories and their subdirectories are protected by the system, not every local account/service account can write to these directories.

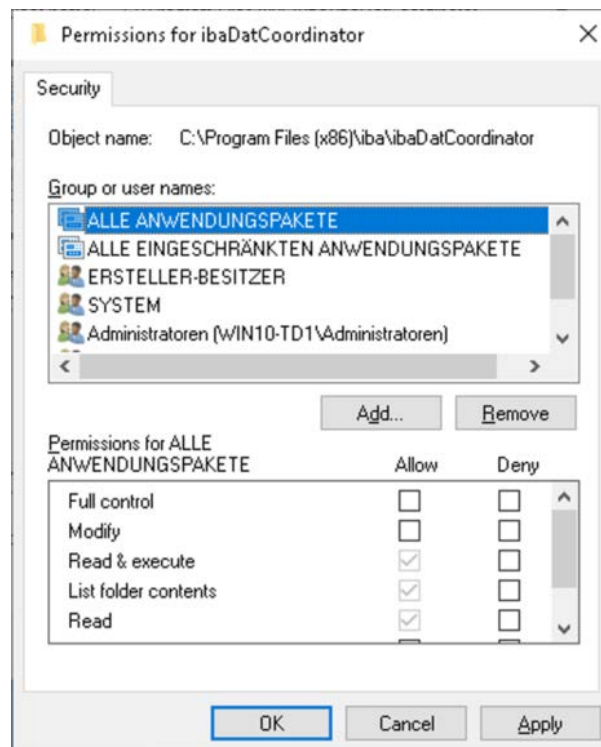
The newly created service account must be added to the directory and provided with the appropriate permissions in order to enable *ibaDatCoordinator* to cache the configurations in the installation directory.

To do this, open the properties of the directory „C:\Program Files (x86)\iba\ibaDatCoordinator“ and select the *Security* tab.



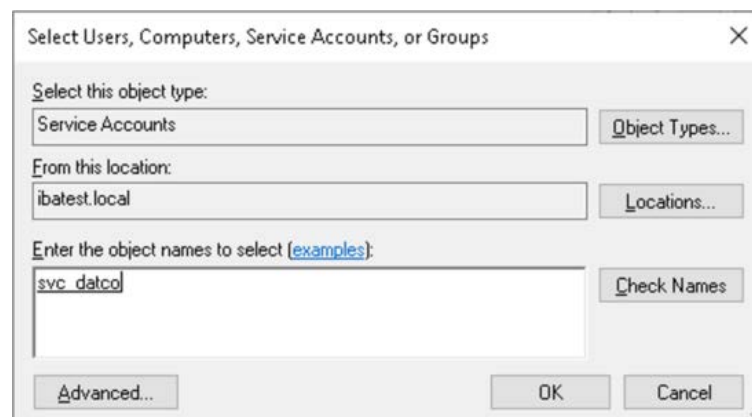
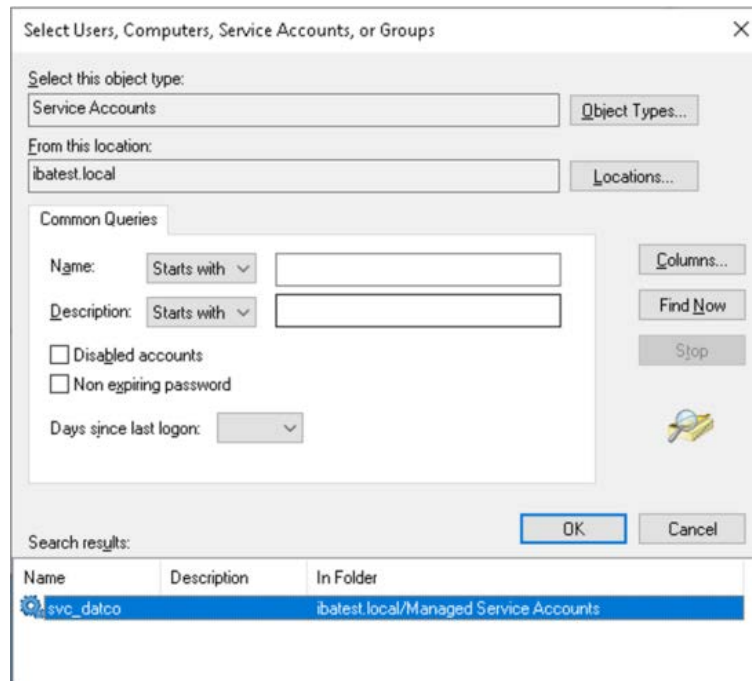
Click on <Edit...>.

In the dialog *Permissions for ibaDatCoordinator* click on <Add...>.



In the following dialog, click on <Advanced...> and change the object type so that only "Service Accounts" is selected.

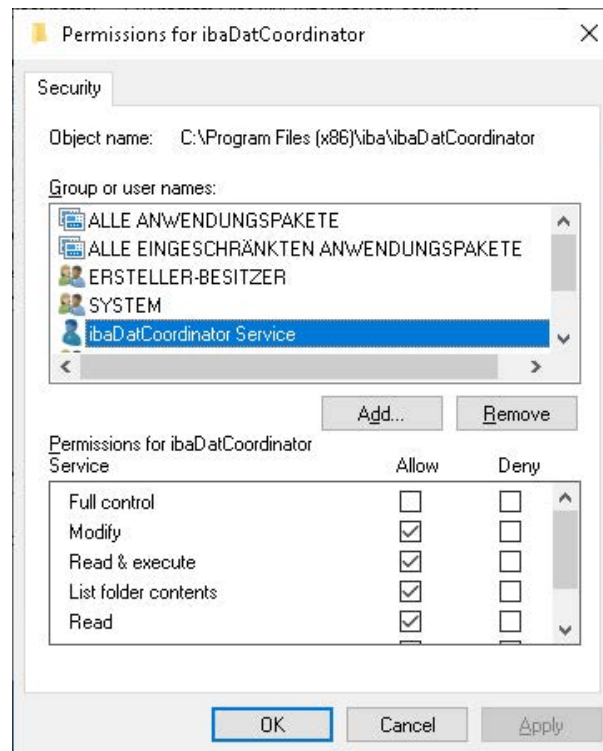
Then click on <Find now> and select the service account „svc_datco“ from the search results.



Exit the dialogs with <OK> until you are back in the dialog *Permissions for ibaDatCoordinator*.

Select the newly added account and set the following permissions

- Modify
- Read, execute
- List folder contents
- Read
- Write



Exit the dialogs *Permissions for ibaDatCoordinator* und *ibaDatCoordinator Properties* with <OK>.

6.4.2.2.2 DCOM permissions

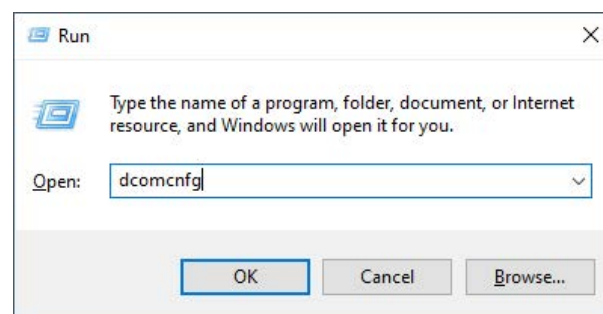
If *ibaDatCoordinator* is operated via a service account, this account lacks the necessary permission to start the *ibaAnalyzer* application.

This appears as the following error in the *ibaDatCoordinator* log:

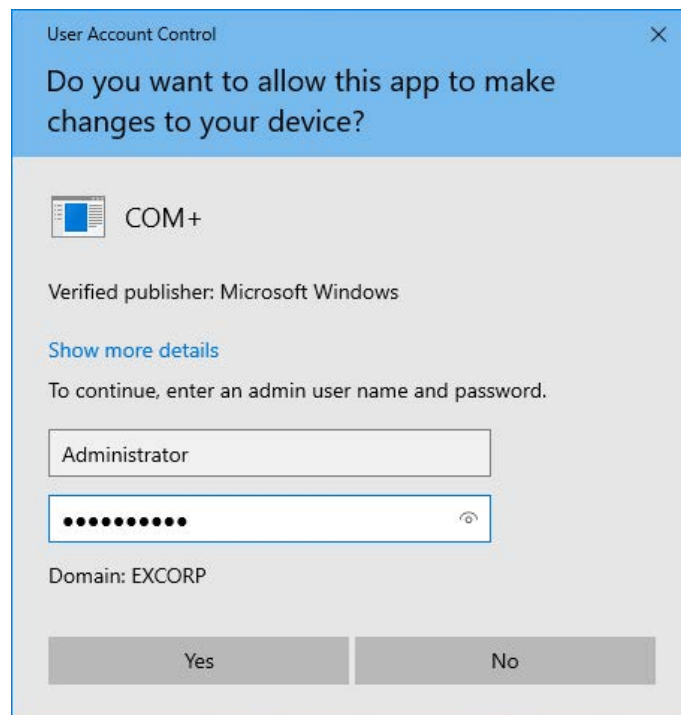
```
Failed to create an instance of ibaAnalyzer: Retrieving the COM class factory for component with CLSID {C4B00861-0324-11D3-A677-000000000000} failed due to the following error: 80070005 Access is denied. (Exception from HRESULT: 0x80070005 (E_ACCESSDENIED)).
```

To eliminate this error, the service account must be allowed to start *ibaAnalyzer* by means of the COM component. For this purpose, various authorizations must be made in the DCOM configuration. To do so, proceed as follows:

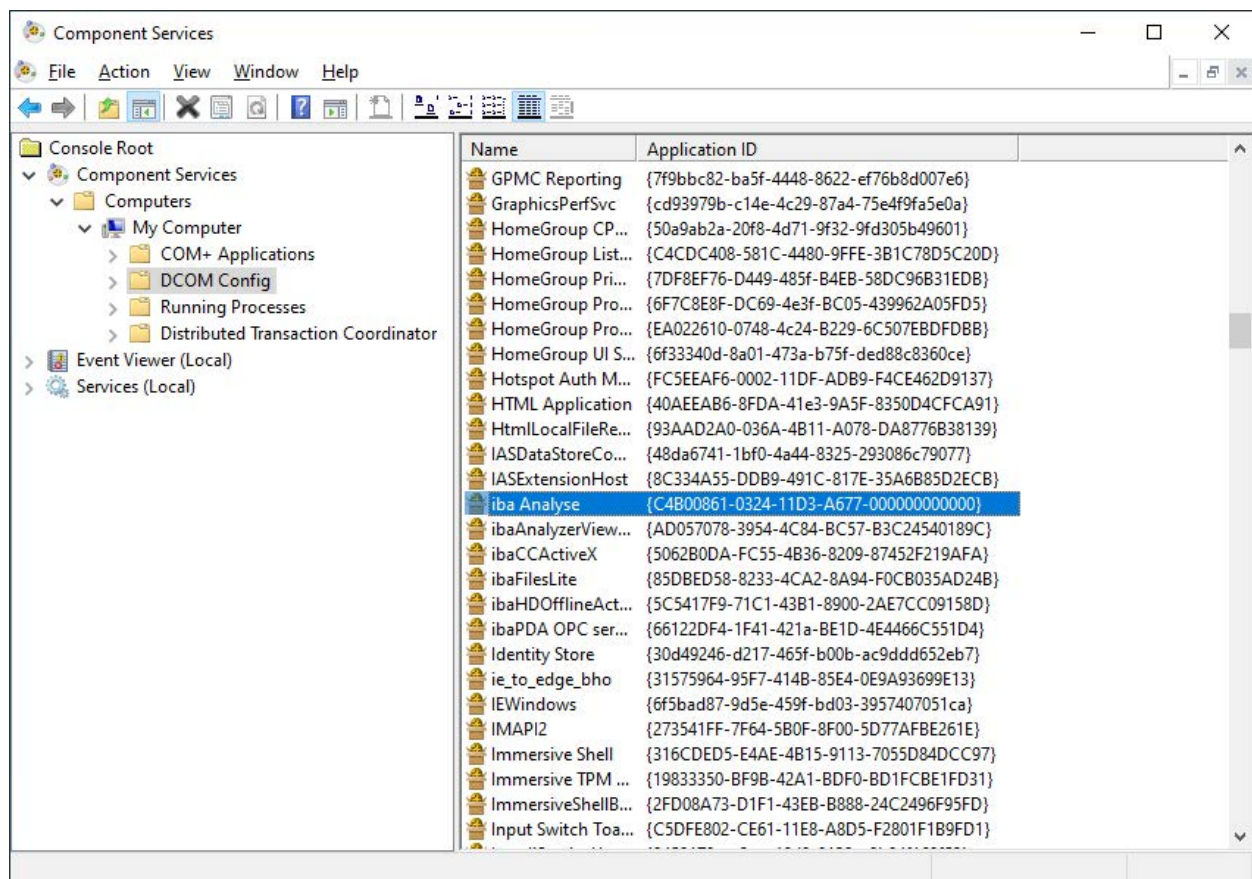
1. Open the component services by pressing <Windows>+<R>, typing "dcomcnfg" and selecting the DCOM configuration in the tree view.



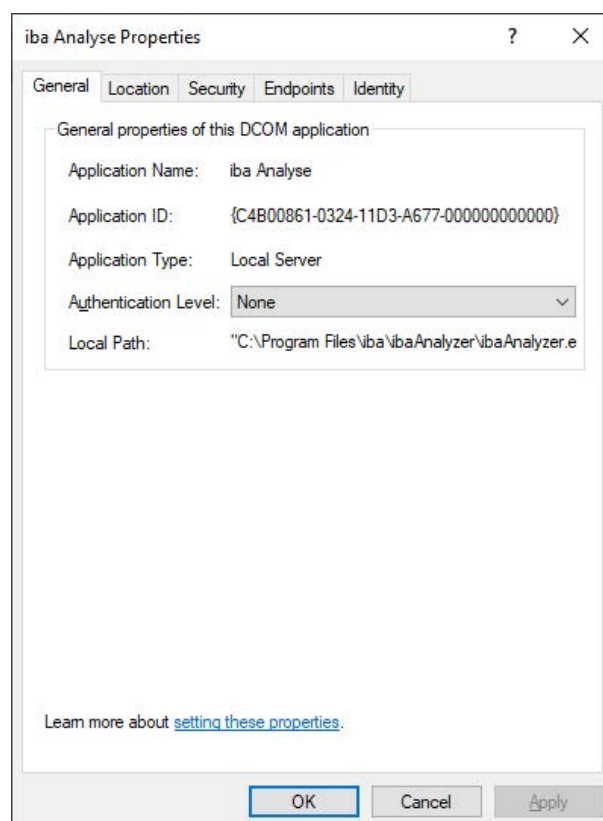
2. As a normal user, you will still need to initiate authorization in order to modify the settings.



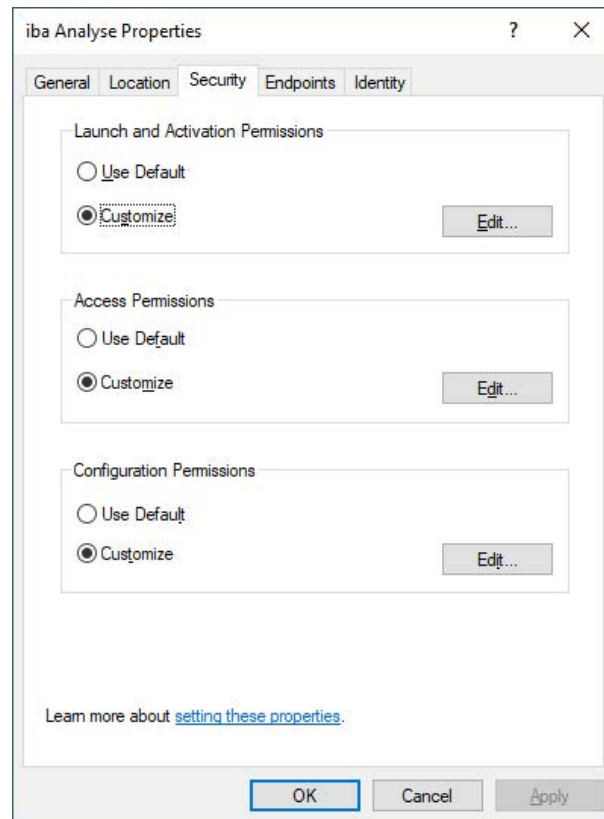
3. Switch to the detailed view.
4. Select the "iba Analyse" element and match the application ID with the CLSID from the error message.



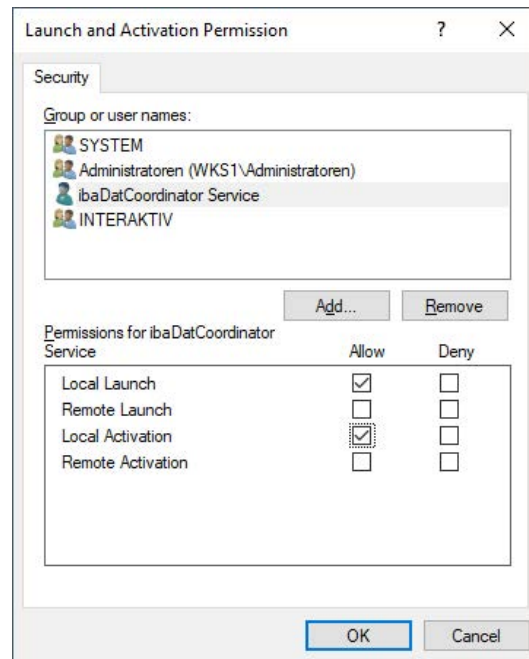
5. Open the properties for the component.
6. In the *General* tab, set the *Authentication level* from "Default" to "None".



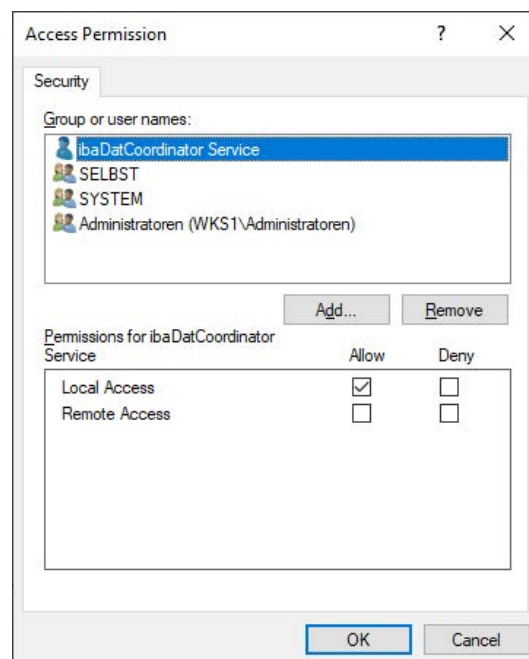
7. Switch to the *Security* tab.
8. Select the “Customize” option for *Launch and Activation Permissions* and *Access Permissions*.



9. Add the new service account to each of the two permission types via <Edit...> and grant it the following permissions:
 - Launch and Activation Permissions
 - Local Launch
 - Local Activation



- Access Permission
 - Local Access



6.4.2.2.3 Registry

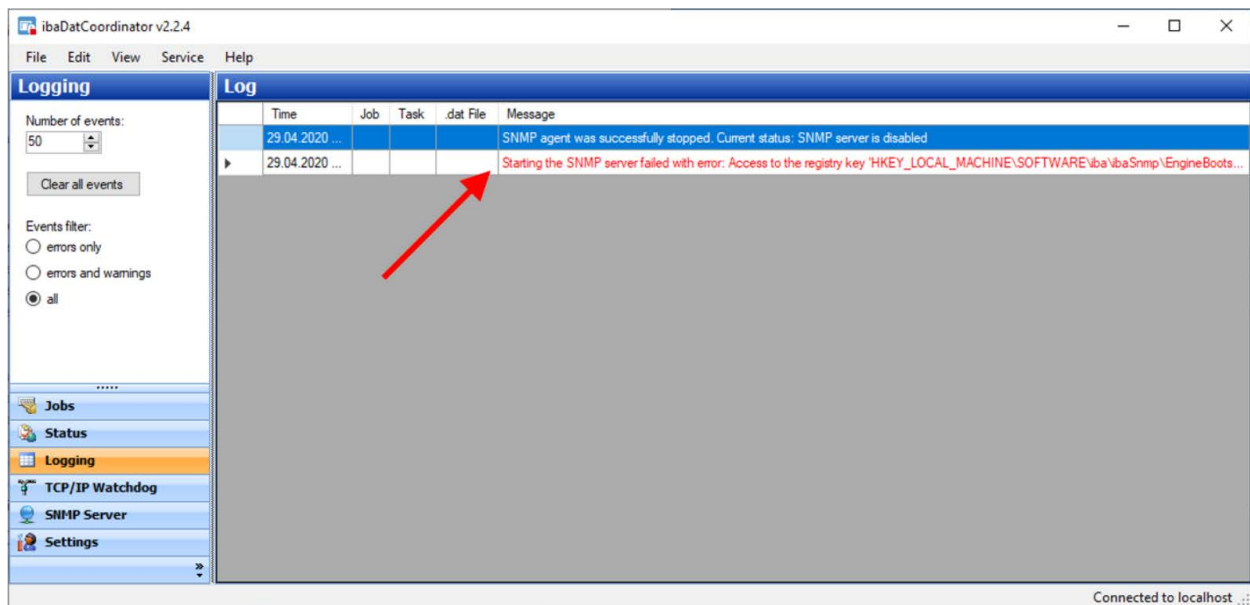
6.4.2.2.3.1 SNMP server

For the SNMP server in *ibaDatCoordinator* to work, it needs read/write access to certain paths in the registry.

HKEY_LOCAL_MACHINE\SOFTWARE\iba\ibaSnmp\EngineBoots

HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\iba\ibaSnmp\EngineBoots

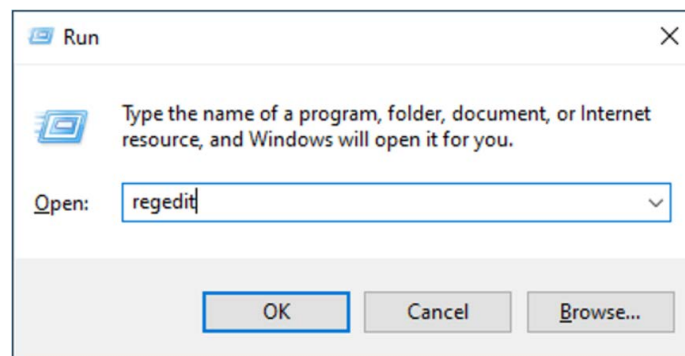
If the account used to start the service does not have access to the above-mentioned paths in the registry, the following message appears in the log of *ibaDatCoordinator*.



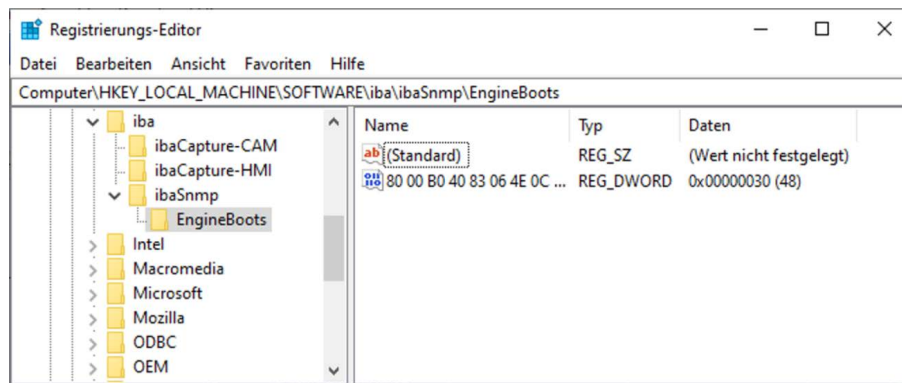
Starting the SNMP server failed with error: Access to the registry key 'HKEY_LOCAL_MACHINE\SOFTWARE\iba\ibaSnmp\EngineBoots' is denied.

To set the appropriate permissions, first open the registry editor.

<Windows>+<R> regedit

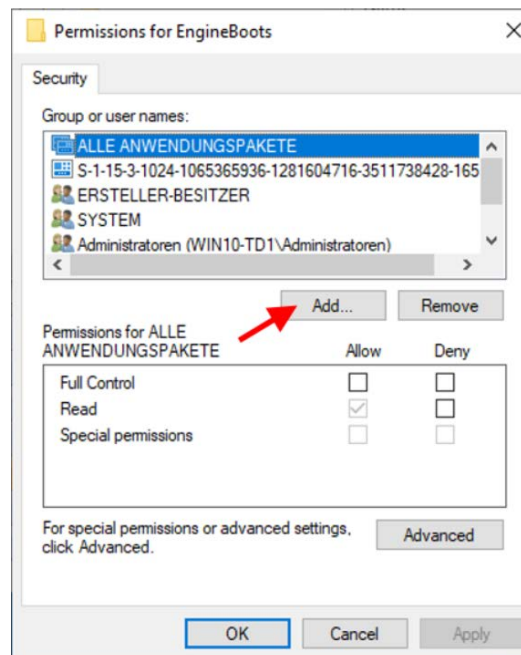


Then navigate to the registry keys mentioned above.

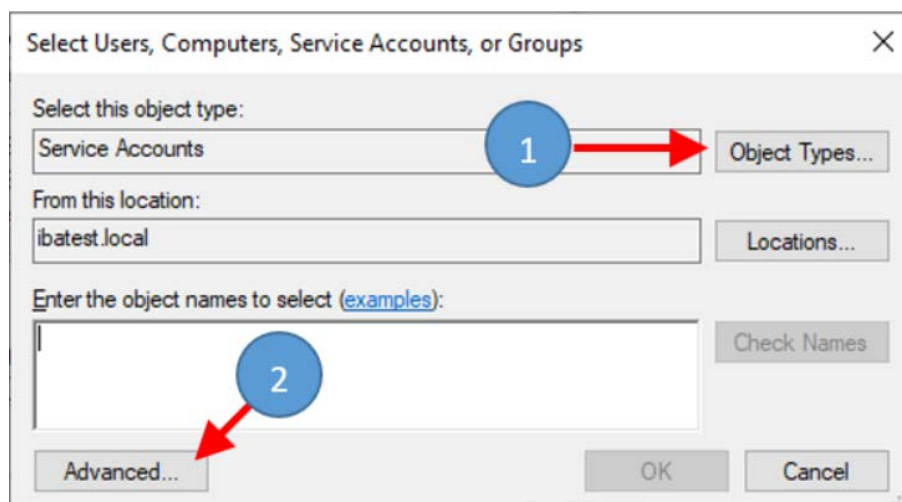


Open the context menu of the key *EngineBoots* and select the item „Permissions...“.

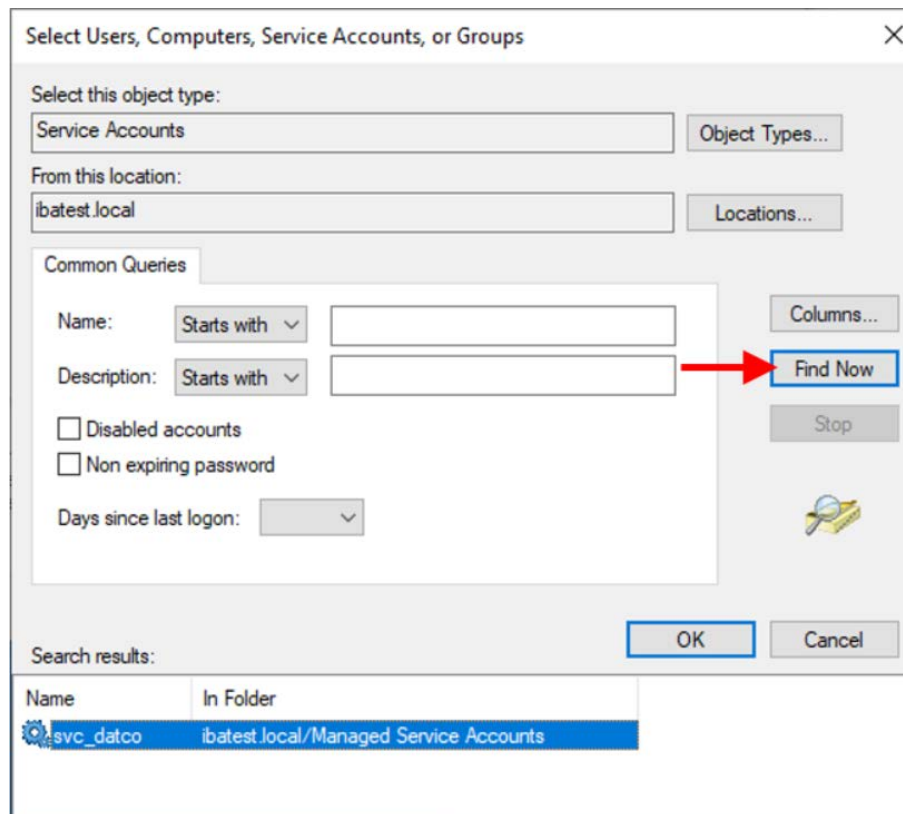
In the dialog *Permissions for EngineBoots* click on the button <Add>.



In the following dialog, first click on <Object Types...> and change the selection, so that only "Service Accounts" is selected. After exit of the dialog *Object Types* click on <Advanced...>.

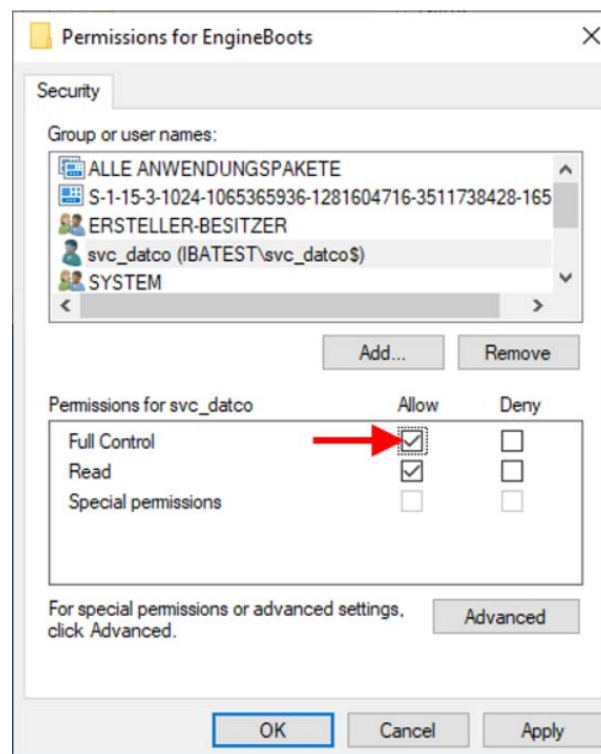


In the following dialog click on <Find now>.



Select the appropriate service account used for the *ibaDatCoordinator* service and exit the dialogs *Select Users, Computers, Service Accounts, or Groups* with <OK>.

Then grant the added account full control to the key.



Repeat this procedure for all keys mentioned above.

6.5 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)
<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.
Synchronization 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

7 Support and contact

Support

Phone: +49 911 97282-14

Email: support@iba-ag.com

Note



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

Contact

Headquarters

iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

Phone: +49 911 97282-0

Email: iba@iba-ag.com

Mailing address

iba AG
Postbox 1828
D-90708 Fuerth, Germany

Delivery address

iba AG
Gebhardtstrasse 10
90762 Fuerth, Germany

Regional and Worldwide

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

www.iba-ag.com