

ibaAnalyzer

Database interface

Manual Part 4

Issue 8.4

Measurement Systems for Industry and Energy

www.iba-ag.com

Manufacturer

iba AG

Gebhardtstrasse 10-20

90762 Fuerth

Germany

Contacts

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

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

© iba AG 2026, All rights reserved.

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

The current version is available for download on our website www.iba-ag.com and can be found in the iba help center docs.iba-ag.com.

Version	Date	Revision	Author	Version SW
8.4	06-2026	Changes to data extractor, database connection, new database schema check; MS Access no longer supported	mm	8.4.0

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

Contents

1 About this documentation 6

1.1 Target group 6

1.2 Notations 6

1.3 Used symbols 7

1.4 Documentation structure 8

2 About the database interface of ibaAnalyzer 9

2.1 Functions and applications 9

2.2 System requirements for database use 10

2.3 Installation 10

2.4 Preparation of the target database 11

3 Configuring the database connection 12

3.1 Connection to SQL server/Azure SQL 15

3.1.1 Basic settings for SQL Server 16

3.1.2 Advanced settings for SQL Server 17

3.2 Connection to Oracle 18

3.2.1 Default connection configuration 20

3.2.2 Advanced connection configuration 21

3.3 Connection to MySQL/MariaDB 22

3.4 Connection to PostgreSQL 24

3.5 Connection to IBM DB2 26

3.5.1 Installing and configuring the IBM data server driver 26

3.5.2 Connection to IBM DB2 in ibaAnalyzer 28

3.6 Connection to SQLite 29

3.7 Changing the connection to ODBC database 30

4 Data Extractor 32

4.1 Extractor output 33

4.1.1 Standard schema mode 33

4.1.2 Multi-column schema mode 35

4.1.3 Validation of the database schema 40

4.2 Archive profile assignment 42

4.3 Info columns 47
4.3.1 Editing the info columns 50
4.4 Computed columns 51
4.5 Diagnostic log 53
4.6 Notifications 54
4.7 Renumbering 55
5 Database format options 57
5.1 Database table "deFile" 57
5.2 Database table "deFileMeta" 59
5.3 "deChannel" database table 60
5.3.1 Using BLOBs 61
5.4 "deSegment" database table 61
5.4.1 Segment tables in standard format 61
5.4.2 Segment tables in multi-column format 63
5.5 Differences between the database providers 64
5.6 Customization and integration 65
6 Analysis of data from databases 67
6.1 Default queries 67
6.1.1 Query builder 67
6.1.2 SQL queries 70
6.1.3 Handling queried datasets 72
6.2 Trend queries 75
6.2.1 Trend query builder 76
6.2.2 SQL trend queries 80
6.3 Trend query results 82
6.3.1 Signal tree of trend query 82
6.3.2 Trend queries in Overview window 82
6.3.2.1 Results list and trend query chart 83
6.3.2.2 Markers 83
6.3.2.3 Overview options 85

7 General handling of databases 86

7.1 SQL statements for ibaAnalyzer 88

7.1.1 SELECT statements 88

7.1.2 AND, OR, NOT, BETWEEN, LIKE 89

7.1.3 Calculations 89

7.1.4 Advanced SELECT clauses (DISTINCT, TOP, CASE) 89

7.1.5 Dealing with NULL 91

7.1.6 JOIN clause 92

7.1.7 Aggregate functions 93

7.2 SQL scripts and scheduling 94

8 Extraction to database via the command line 95

9 Remote access to an SQL Server 96

9.1 Configuring remote access 96

9.2 Firewall settings for remote access 97

10 Support and contact 102

1 About this documentation

This documentation describes the function and application of the software

ibaAnalyzer.

1.1 Target group

This documentation addresses qualified professionals, who are familiar with handling electrical and electronic modules as well as communication and measurement technology. A person is regarded as trained specialist personnel if they are capable of assessing the work assigned to them and identifying possible risks based on their specialist training, knowledge, experience and awareness of the applicable regulations.

This documentation addresses in particular professionals who are in charge of analyzing measured data and process data. Because the data is supplied by other iba products the following knowledge is required or at least helpful when working with *ibaAnalyzer*:

- Operating system Windows
- *ibaPDA* (creation and structure of the measuring data files)

1.2 Notations

In this manual, the following notations are used:

Action	Notation
Menu command	Menu *Logic diagram*
Calling the menu command	*Step 1 – Step 2 – Step 3 – Step x* Example: Select the menu *Logic diagram – Add – New function block*.
Keys	<Key name> Example: <Alt>; <F1>
Press the keys simultaneously	<Key name> + <Key name> Example: <Alt> + <Ctrl>
Buttons	<Key name> Example: <OK>; <Cancel>
Filenames, paths	`Filename`, `Path` Example: `Test.docx`

1.3 Used symbols

If safety instructions or other notes are used in this manual, they mean:

Danger!



The non-observance of this safety information may result in an imminent risk of death or severe injury!

Observe the specified measures.

Warning!



The non-observance of this safety information may result in a potential risk of death or severe injury!

Observe the specified measures.

Caution!



The non-observance of this safety information may result in a potential risk of injury or material damage!

Observe the specified measures.

Note



A note specifies special requirements or actions to be observed.

Tip



Tip or example as a helpful note or insider tip to make the work a little bit easier.

Other documentation



Reference to additional documentation or further reading.

1.4 Documentation structure

This documentation describes the functionality of the *ibaAnalyzer* software in detail. It is designed both as a tutorial as well as a reference document.

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

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

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

The *ibaAnalyzer* documentation (PDF version) is divided into six separate parts. Each part has its own section and page numbering beginning at 1 and is updated independently.

Part	Title	Content
Part 1	Introduction and Installation	General notes, licenses and add-ons Installation and program start User interface
Part 2	Working with *ibaAnalyzer*	Working with measurement files and analyses, presentation features, macro configuration, filter design, preferences, printing, export, file extract, interfaces to *ibaHD-Server*, and *ibaCapture*
Part 3	Expression editor	Directory of all calculation functions in the expression builder, including explanation
Part 4	Database interface	Working with data from databases, connecting to the database, writing iba measurement data to databases, extracting the data from the database and analyzing the data.
Part 5	Application examples	*In preparation*

2 About the database interface of ibaAnalyzer

The *ibaAnalyzer* database interface is a default function that can be used interactively without an additional license. This interface provides ETL (Extract Transform Load) functions for commonly used database providers and thus enables trend and detail analyses based on databases. The *ibaAnalyzer* report generator can also process data from the database interface.

An additional license is only required if you want to automate the data exchange with the database. As the *ibaDatCoordinator* program is usually used in practice for the automated processing of measurement data, an *ibaDatCoordinator-DB* license is needed.

To read or query databases, the *ibaAnalyzer-DB-Read* license is needed.

2.1 Functions and applications

Database extraction is an integral function of *ibaAnalyzer* and is enabled during the installation of *ibaAnalyzer*. All data in iba measurement file (*.dat) format originating from *ibaPDA*, *ibaQDR*, *ibaLogic* or third-party applications that use the *ibaFiles* library can be easily extracted, transformed and loaded.

ibaAnalyzer can open iba measurement files (*.dat), which have a lossless and very efficiently compressed binary format. This makes it easy to handle large volumes of data. All the functions of *ibaAnalyzer* can be used to transform or generate the required (virtual) signals. Specifying storage profiles for database extraction allows re-sampling or aggregation of measurement data and virtual signals. This means that every data extraction can lead to a significant reduction of disk space by compressing information. *ibaAnalyzer* supports several databases: Microsoft SQL Server (local or in MS Azure Cloud), Oracle, DB2-UDB, MySQL/Maria-DB, PostgreSQL and SQLite. These databases may be located on the same computer or installed on a separate database server that has a network connection to the *ibaAnalyzer* computer. If you want to use other databases, please contact iba support.

ibaDaVIS, *ibaDatManager*, MES (Manufacturing Execution Systems), DWH applications (DataWareHouse) or e.g. other database-based cloud applications can easily integrate the data provided, as data handling can be performed using standard database tools and access methods.

Please note that this would be impossible in most cases with the very large raw datasets.

ibaAnalyzer treats the values extracted into a database as if they were values in DAT files. It is possible to query this data using a query builder or direct SQL commands. Data aggregated on a file basis can be used for long-term analyses (trend queries). In addition, these trend queries provide a navigation view for drill down analyses, which can be based on database values or the linked original files.

Alternatively, iba offers further extensions of *ibaAnalyzer* for extraction into new iba DAT files or ASCII files as well as into files of the COMTRADE, TDMS, MatLab or Parquet formats, see part 2 of the *ibaAnalyzer* manual, chapter *Data export and extraction*.

Summary of the most important functions:

- Data extraction from iba DAT files
- Data transformation or generation of new virtual signals
- Data loading into databases via ADO.NET interface
- Data detail analyses from database instead of DAT files
- Data trend analyses based on aggregated values
- Data drill down option from trend to detail

2.2 System requirements for database use

ibaAnalyzer supports the following databases:

- SQL Server/Azure SQL
- Oracle
- MySQL/MariaDB
- PostgreSQL
- IBM DB2-UDB
- SQLite

Notes for users of existing solutions with ibaAnalyzer-DB and USB-MARX dongle
Even after updating *ibaAnalyzer* to v8.x and *ibaDatCoordinator* v3.x it is possible to continue using USB-MARX dongles with the *ibaAnalyzer-DB* or *ibaAnalyzer-File-Extract* licenses as normal.

2.3 Installation

The database interface is automatically installed and activated together with *ibaAnalyzer*. During installation, no special steps are required.

Note



The distinction between single-column format and multi-column format known from previous versions no longer applies. This is set later when the database connection is configured.

2.4 Preparation of the target database

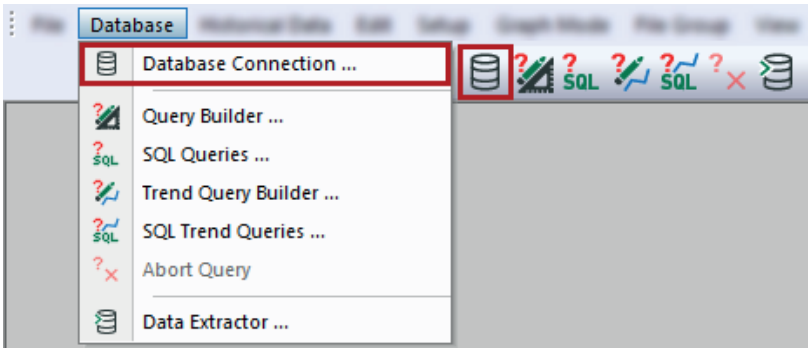
Depending on the target database and type of application, some IT requirements are necessary:

- Create or make available database
- Configure DB storage (database files, table spaces ...)
- Configure DB login or access (username and password)
- Configure network access (e.g. database listener or login via Active Directory)
- Configure firewall

The following section describes the connection settings and any characteristics of the various database providers, see ↗ *Configuring the database connection, page 12*.

3 Configuring the database connection

You can open the database connection either via the *Database – Database connection* menu or via the corresponding button in the toolbar.



If the database connection has not been previously configured, *ibaAnalyzer* shows the following default configuration. Otherwise, the previously selected configuration is shown.

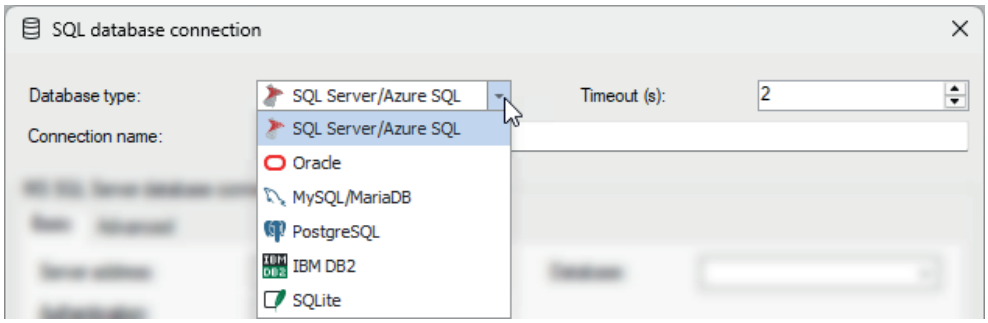


Setting up and editing database options

From the *Data source* drop-down list, select *Create database connection*. In the *Database connection* dialog box, select the required database provider under *Database type*.

To edit an existing connection, select the connection from the *Data source* drop-down list. Then click on the editing button.

Enter the other settings.



Information about the setup of the database connections can be found in the respective chapters:

- SQL Server/Azure SQL, see ↗ *Connection to SQL server/Azure SQL, page 15*
- Oracle, see ↗ *Connection to Oracle, page 18*
- MySQL/MariaDB, see ↗ *Connection to MySQL/MariaDB, page 22*
- PostgreSQL, see ↗ *Connection to PostgreSQL, page 24*
- DB2-UDB, see ↗ *Connection to IBM DB2, page 26*
- SQLite, see ↗ *Connection to SQLite, page 29*
- ODBC database only for changing older PDO files, see ↗ *Changing the connection to ODBC database, page 30*

Note

ibaAnalyzer v8.4.0 or later no longer supports connections to MS Access databases. Migrate existing MS Access configurations to a different database system.

Note

ibaAnalyzer v8.0.0 or higher no longer supports any database connections via ODBC. Databases that have been configured in older PDO files via ODBC must be converted to the actual database provider in *ibaAnalyzer* and directly connected to the database.

Testing the database connection in ibaAnalyzer

After set-up and before closing the *Database connection* window, test the database connection using the <Test connection> button.

Entering names of database tables

Enter the names of the database tables in the *File table*, *Channel table* and *Segment table* fields. The name of the file table also determines the name for the file metadata table, see ↗ *Database table "deFileMeta", page 59*. It is recommended to name these tables in accordance with the naming conventions of the database system you are using.

For test purposes, you can use the default names "deFile", "deChannel" and "deSegment".

Note

The names must comply with the naming conventions of the used database system. Avoid spaces, hyphens, special characters etc. and check the permitted length.

For example, if Oracle is used, the maximum length for the name of the segment table is 23 characters because 7 characters are required for the segment table suffixes "_CountT" and "_CountL".

Note



If the channel headers or segment tables are not needed, you can leave their names blank. The tables are then not created. This procedure is suitable if you are only working with trend data.

Selecting schema mode for database

Select the schema mode for the database.

- *Standard*, see ↗ *Standard schema mode, page 33* and ↗ *Segment tables in standard format, page 61*
- *Multi-column* format, see ↗ *Multi-column schema mode, page 35* and ↗ *Segment tables in multi-column format, page 63*

Saving configuration

Exit the *Database connection* window using <OK> to save the settings.

Save the configuration in an analysis file (PDO) afterwards.

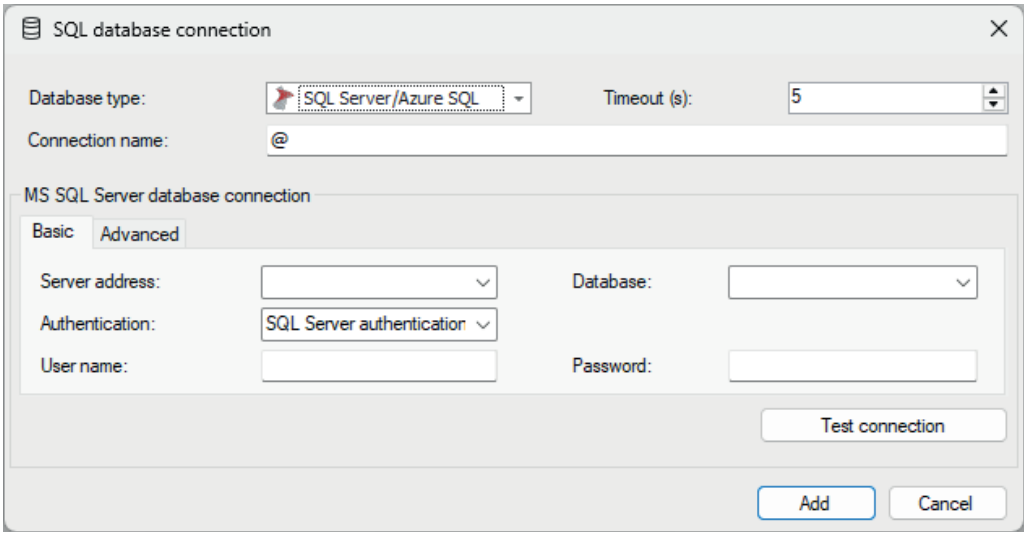
If you open *ibaAnalyzer* without an analysis file, *ibaAnalyzer* loads the latest configuration of the database connection in the same way as the other preferences.

3.1 Connection to SQL server/Azure SQL

To connect *ibaAnalyzer* to an SQL server, you do not need to make any special settings on the database side.

Note

ibaAnalyzer v8.0.0 or higher no longer supports any database connections via ODBC. Databases that have been configured in older PDO files via ODBC must be converted to the actual database provider in *ibaAnalyzer* and directly connected to the database.



From the *Data source* drop-down list, select *Create database connection*. In the *Database connection* dialog box, select the required database provider under *Database type*.

To edit an existing connection, select the connection from the *Data source* drop-down list. Then click on the editing button.

Enter the other basic settings, see ↗ *Basic settings for SQL Server, page 16*.

You can also use the advanced settings to configure a failover partner, see ↗ *Advanced settings for SQL Server, page 17*.

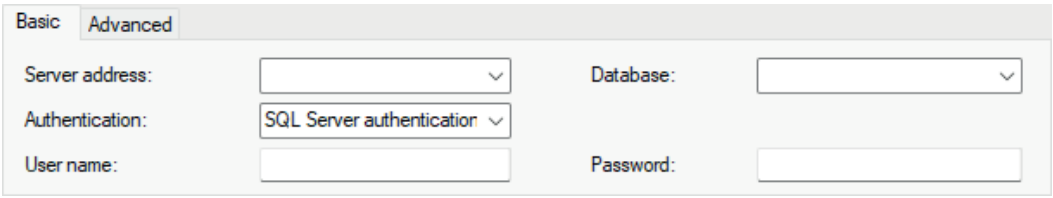
Connection name
This line contains the name of the database connection. The name is automatically formed and entered according to the scheme *username@database* as soon as you have filled in the fields below. Alternatively, you can enter and overwrite the connection name manually.

Timeout
Here you can specify a value for the timeout in seconds for establishing the connection. If the time set here is exceeded, *ibaAnalyzer* aborts the connection attempt and displays an error message.

Note that once the connection is made, each command sent to the database has an additional timeout, which can be set by the application that makes use of it. There is no generic configuration for the command timeout.

<Test connection>
ibaAnalyzer checks the access to the selected database on the set DB server. If the connection is successful, the message includes the version of the database.

3.1.1 Basic settings for SQL Server



Server address
Enter the IP address, the host name or a combination of the host name and instance name of the SQL Server database server. You can enter the server address or choose from the drop-down list. The list shows the database instances that *ibaAnalyzer* can reach. Only the registered SQL Server instances are automatically shown on the computer on which *ibaAnalyzer* is installed. To search for remote database servers on the network, use the *<Discover network servers>* entry at the end of the drop-down list.

For the server address, you can also specify a port. Separate the port from the server name with a comma. If no port is specified for the server address, the default port 1433 is used.

Example: To connect to port 1488 on server sqlserver.domain, use "sqlserver.domain,1488".

Authentication type
Select the appropriate authentication method here to obtain access to the database. Please consult with your DB administrator to find out which method is the right one and what the log-in information is.

- *Windows authentication*:
  To connect to the SQL Server instance, the user account under which the SQL Server is running is used.

Note

For automated use with *ibaDatCoordinator-DB* license:

When using the local system account, you must grant additional authorizations to the corresponding login *NT AUTHORITY\SYSTEM* in the SQL Server that are not set up by default. The rights to connect to the database as well as other rights are required to be able to execute the planned SQL statements, such as read rights for queries. Write rights are only required if you use output modules for commands (e.g. UPDATE, INSERT). Clarify with your DB administrator in advance whether the extension of rights is possible or whether a different login or authentication should be selected.

- *SQL Server authentication*:
  Authentication that is stored in the database. A username and password are required for this purpose.
- *Azure Active Directory - Universal with MFA*:
  Multi-factor authentication, as configured on Azure.

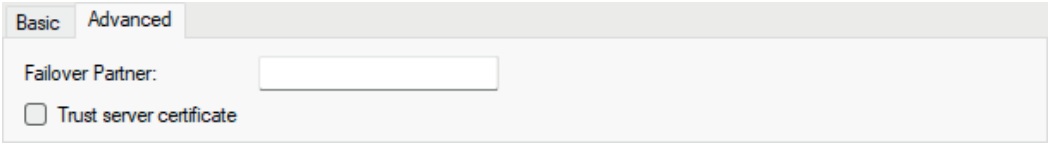
- *Active Directory - Password or Integrated*:
  The authentication methods can be used for members of a user group in the active directory, which has access to the database. A relationship of trust must be established between the user and their device and domain. If you select *Active Directory - Password,* you must enter username and password.

Username/Password
Enter here the required login data for the database. If uncertain, ask your DB administrator.

Depending on the authentication method selected, these fields are available or not available.

Database
Enter the database here. When you have entered the server address, authentication or username and password. The drop-down list shows all available databases for the connection and you can select one.

3.1.2 Advanced settings for SQL Server



Failover partner
You can set up a failover partner for an SQL Server instance. For this, a failover cluster must be configured on the SQL Server instances that are part of the cluster.

In order to connect to a failover cluster, it is sufficient to enter something in this text field. If the correct address of the failover partner is entered, failover can already take place upon the initial connection to the database. Failover is not used if you leave this field blank.

More information about failover can be found here: https://learn.microsoft.com/en-us/sql/sql-server/failover-clusters/windows/always-on-failover-cluster-instances-sql-server?view=sql-server-ver16

Trust server certificate
Use this option only for testing purposes or as an emergency solution! The SQL Server authenticates itself with a TLS certificate. You can use this option to trust the server certificate even if it cannot be verified. Preferably, ask your administrator to set up authentication properly.

3.2 Connection to Oracle

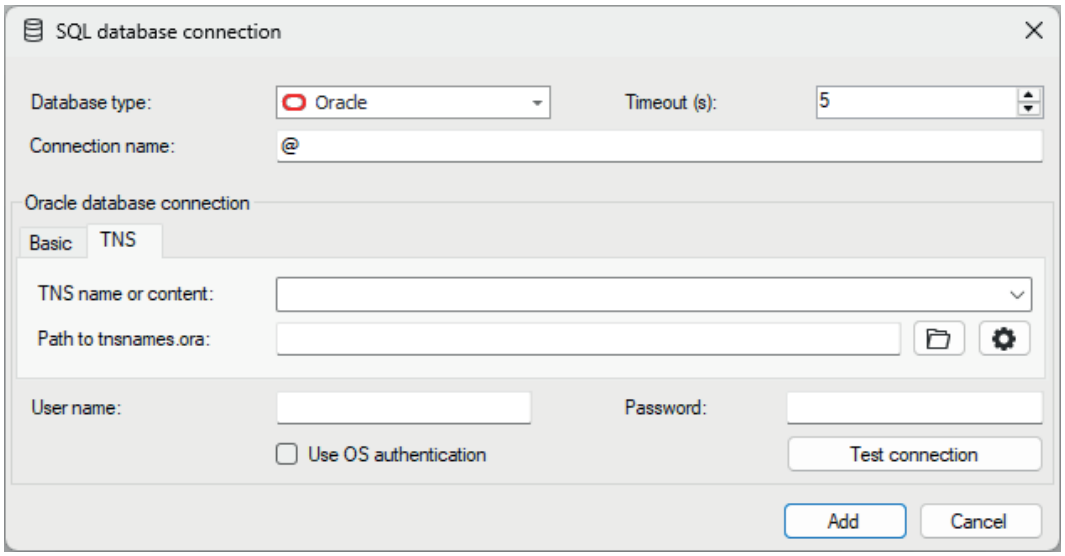
In a centralized Oracle multiuser DB environment, it is recommended to create a dedicated user, scheme and disk space (tablespace). Give this user the necessary rights for the loading process in *ibaAnalyzer* ("create session", "create table" ...).

For interactive database queries from different people, it is recommended to create a single shared or multiple dedicated database users with corresponding rights ("grant select on" ...). This applies to large installations where the iba data is not to be mixed with data from existing applications. Configure the database in such a way that the user has read and/or write rights thus preventing, for example, the deletion of existing data using the *Create default tables* option of *ibaAnalyzer*.

An Oracle client is not needed to establish a connection from *ibaAnalyzer* to an Oracle database.

Note

ibaAnalyzer v8.0.0 or higher no longer supports any database connections via ODBC. Databases that have been configured in older PDO files via ODBC must be converted to the actual database provider in *ibaAnalyzer* and directly connected to the database.



From the *Data source* drop-down list, select *Create database connection*. In the *Database connection* dialog box, select the required database provider under *Database type*.

To edit an existing connection, select the connection from the *Data source* drop-down list. Then click on the editing button.

It is recommended to configure the connection via the *TNS* tab, see ↗ *Default connection configuration, page 20*. Configuring the database connection via the *Basis* tab is recommended for DB administrators and developers, see ↗ *Advanced connection configuration, page 21*.

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. Alternatively, you can enter and overwrite the connection name manually.

Timeout
Here you can specify a value for the timeout in seconds for establishing the connection. If the time set here is exceeded, *ibaAnalyzer* aborts the connection attempt and displays an error message.

Note that once the connection is made, each command sent to the database has an additional timeout, which can be set by the application that makes use of it. There is no generic configuration for the command timeout.

Username/Password
Enter here the required login data for the database. If uncertain, ask your DB administrator.

These fields are only available if the option *OS-Authentication* is disabled.

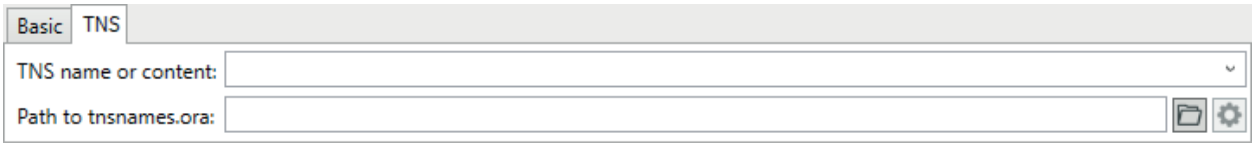
Use OS authentication
If you enable this option, the user account is used under which *ibaAnalyzer* runs. Usually, this is the system account. Contact your DB administrator to find out if this setting is suitable.

<Test connection>
ibaAnalyzer checks the access to the selected database on the set DB server. If the connection is successful, the message includes the version of the database.

3.2.1 Default connection configuration

It is recommended to configure the connection via the *TNS* tab.

The configuration with TNS also enables the use of a failover cluster instead of a single server configuration for the Oracle database. Please contact your DB administrator about this topic.



TNS name or content
For TNS configuration, use one of the following possibilities:

- Enter the name of a connection that is available in the "tnsnames.ora" file in the Oracle client directory.
- Enter a complete section like it can be found in a "tnsnames.ora" file.

Example:

```
(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)
(HOST=192.168.99.101)(PORT=1521)))(CONNECT_DATA=
(SERVER=DEDICATED)(SERVICE_NAME=xe)))
```

Path to tnsnames.ora
If an Oracle client is already installed on the computer, the appropriate `tnsnames.ora` file can be searched for. You can also specify the path to the file or select it using the browser button.

If the file does not exist, a suitable file must be created or made available by the DB administrator. Save the file on your computer and enter the path.

If you specify the complete contents of the `tnsnames.ora` file in the first line (*TNS name or content*), the *Path* field can remain empty. If you specify only the TNS name, you must specify the path to the complete file.

As soon as you have specified a path, you can use the gear button to set this path as the default path.

Automatic search
The search is first done in the registry key TNS_ADMIN. There you can enter the reference to `tnsnames.ora` as admin.

If the file is found there, then the search is done in the default directory for an Oracle client. If the file is also found there, then the *ibaAnalyzer* server directory is searched.

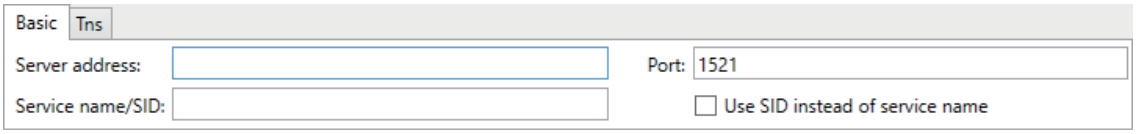
If `tnsnames.ora` is found, then the path is also shown in the *Path* field.

If you overwrite the path manually, then the entered path is valid.

If no `tnsnames.ora` file is found via the automatic search, then an error message appears with a dialog, in which you can select the file.

3.2.2 Advanced connection configuration

Configuring the database connection via the *Basis* tab is recommended for DB administrators. You cannot configure a failover cluster with these settings. Use the *TNS* tab for this, see ↗ *Default connection configuration, page 20*.



Server address
Enter the IP address or the host name of the database server.

Port
The DB server communicates via this port. Usually, you can keep the default port 1521. If a different port is required, you can enter this port here.

Service name/SID
The name of the Oracle service as it is configured in the Oracle server.

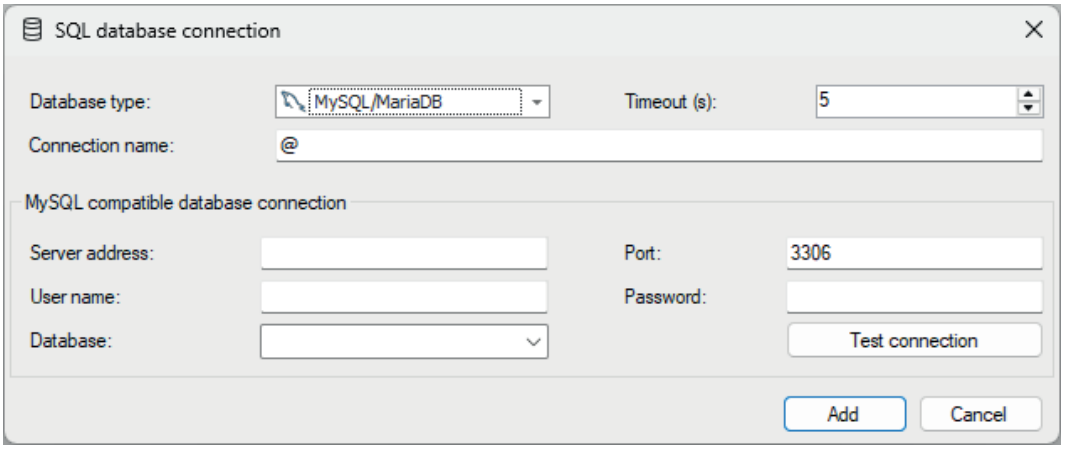
Use SID instead of service name
In some rare cases, the SID must be used. Refer to the Oracle documentation for the difference between SID and service name.

3.3 Connection to MySQL/MariaDB

To connect *ibaAnalyzer* to a MySQL database or MariaDB, you do not need to make any special settings on the database side.

Note

ibaAnalyzer v8.0.0 or higher no longer supports any database connections via ODBC. Databases that have been configured in older PDO files via ODBC must be converted to the actual database provider in *ibaAnalyzer* and directly connected to the database.



From the *Data source* drop-down list, select *Create database connection*. In the *Database connection* dialog box, select the required database provider under *Database type*.

To edit an existing connection, select the connection from the *Data source* drop-down list. Then click on the editing button.

Enter the other settings.

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. Alternatively, you can enter and overwrite the connection name manually.

Timeout
Here you can specify a value for the timeout in seconds for establishing the connection. If the time set here is exceeded, *ibaAnalyzer* aborts the connection attempt and displays an error message.

Note that once the connection is made, each command sent to the database has an additional timeout, which can be set by the application that makes use of it. There is no generic configuration for the command timeout.

Server address
Enter the IP address or the host name of the database server.

Port
The DB server communicates via this port. Usually, you can keep the default port 3306. If a different port is required, you can enter this port here.

Username/Password
Enter here the required login data for the database. If uncertain, ask your DB administrator.

Database
Enter the database here. When you have entered the server address, authentication or username and password. The drop-down list shows all available databases for the connection and you can select one.

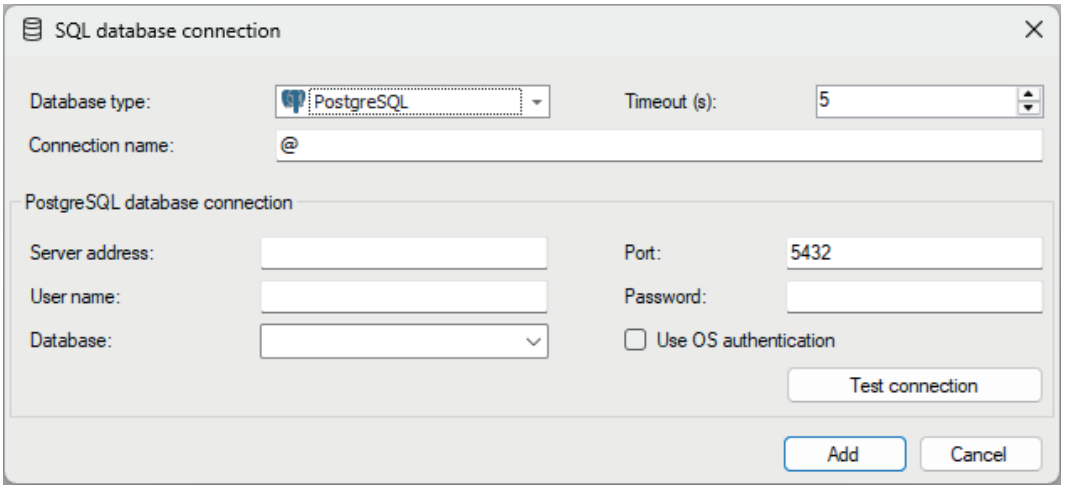
<Test connection>
ibaAnalyzer checks the access to the selected database on the set DB server. If the connection is successful, the message includes the version of the database.

3.4 Connection to PostgreSQL

To connect *ibaAnalyzer* to a PostgreSQL database, you do not need to make any special settings on the database side.

Note

ibaAnalyzer v8.0.0 or higher no longer supports any database connections via ODBC. Databases that have been configured in older PDO files via ODBC must be converted to the actual database provider in *ibaAnalyzer* and directly connected to the database.



From the *Data source* drop-down list, select *Create database connection*. In the *Database connection* dialog box, select the required database provider under *Database type*.

To edit an existing connection, select the connection from the *Data source* drop-down list. Then click on the editing button.

Enter the other settings.

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. Alternatively, you can enter and overwrite the connection name manually.

Timeout
Here you can specify a value for the timeout in seconds for establishing the connection. If the time set here is exceeded, *ibaAnalyzer* aborts the connection attempt and displays an error message.

Note that once the connection is made, each command sent to the database has an additional timeout, which can be set by the application that makes use of it. There is no generic configuration for the command timeout.

Server address
Enter the IP address or the host name of the database server.

Port
The DB server communicates via this port. Usually, you can keep the default port 5432. If a different port is required, you can enter this port here.

Username/Password
Enter here the required login data for the database. If uncertain, ask your DB administrator.

These fields are only available if the option *OS-Authentication* is disabled.

Use OS authentication
If you enable this option, the user account is used under which *ibaAnalyzer* runs. Usually, this is the system account. Contact your DB administrator to find out if this setting is suitable.

Database
Enter the database here. When you have entered the server address, authentication or username and password. The drop-down list shows all available databases for the connection and you can select one.

<Test connection>
ibaAnalyzer checks the access to the selected database on the set DB server. If the connection is successful, the message includes the version of the database.

3.5 Connection to IBM DB2

Note

ibaAnalyzer v8.0.0 or higher no longer supports any database connections via ODBC. Databases that have been configured in older PDO files via ODBC must be converted to the actual database provider in *ibaAnalyzer* and directly connected to the database.

To connect *ibaAnalyzer* to an IBM DB2 database, you have to install the IBM data server driver software. The version 11.5.7 of IBM data server driver has been tested successfully with *ibaAnalyzer*.

ibaAnalyzer does not support connections to mainframes (z/OS) and no authentication via Kerberos (active directory).

3.5.1 Installing and configuring the IBM data server driver

Installing the IBM data server driver

1. Download the IBM data server driver as a fix pack from the IBM website, e.g. here: https://www.ibm.com/support/pages/download-fix-packs-version-ibm-data-server-client-packages

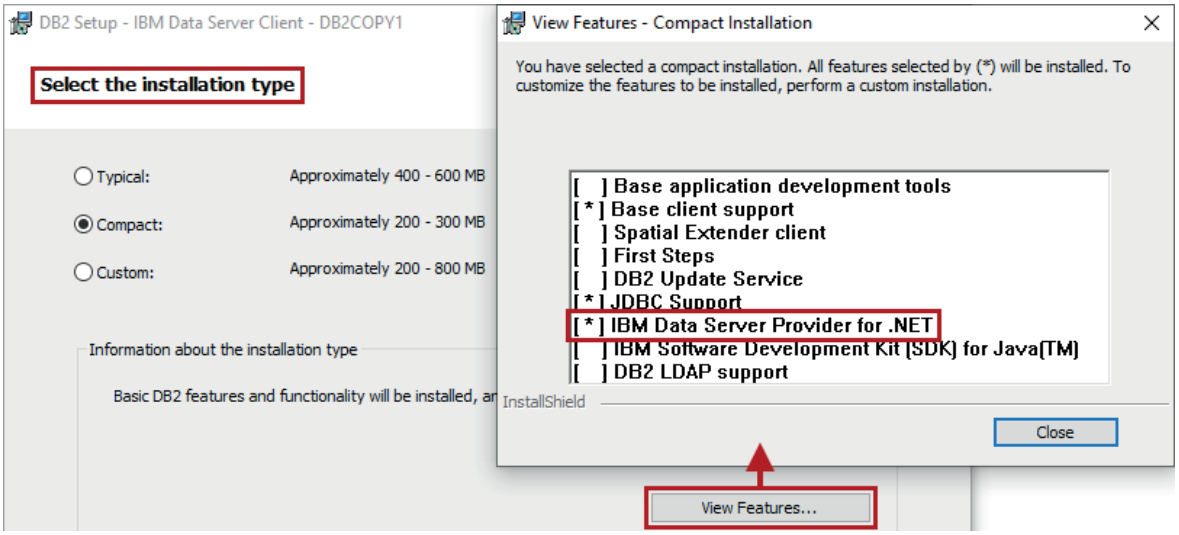
   Create a free IBM ID for this.

   Select the latest fix pack that corresponds to the bit version of your Windows version.

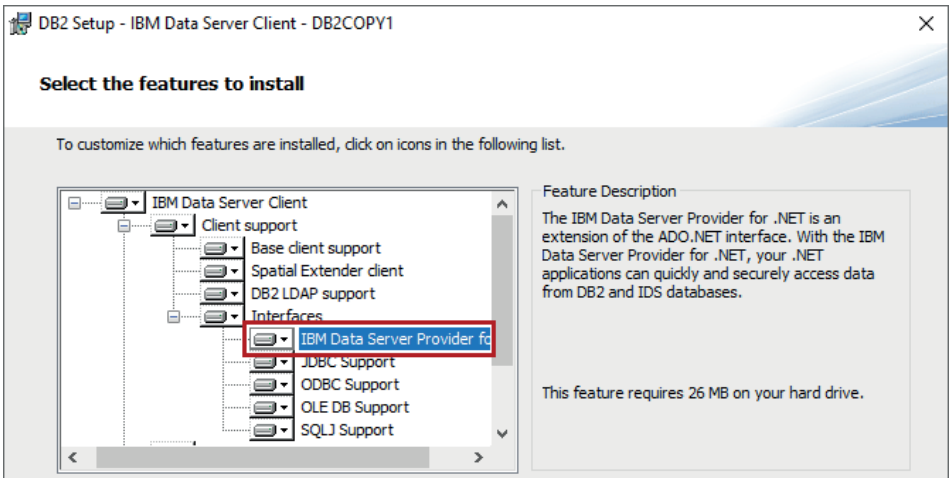
2. Execute the file `v11.5.Y_ntZ_client.exe`

→ The installer will be extracted to your chosen directory.

3. Execute the `setup.exe` file in the "CLIENT" folder in the selected directory.
4. Select *Install product* and install the IBM Data Server Client using the <New installation> button.
5. Follow the instructions of the installation wizard.
6. During the *Select installation type* step, ensure that the *IBM Data Server Provider for .NET* is included. However, you can select any installation type.



If you select the *Custom* installation type, you can choose the individual components later.



7. Continue the installation with the desired settings and install the driver.

   If an alert about missing user rights appears, it can be ignored.

Configuring the database name via IBM DB2 Command Window

To configure the database name, it is necessary to execute a few commands in the DB2 Command Window.

1. Open the DB2 command window, e.g. via the start menu: *Start – IBM – DB2 Command Window*
2. Add a node by entering the following command in the Command Window:

   `db2 catalog tcpip node <nodename> remote <ip> server <port>`
3. Connect the database to the node by entering the following command:

   `db2 catalog database <dbname> at node <nodename>`
4. Transfer the changes to the client by entering the following command:

   `db2 terminate`

→ The following example shows the configuration for a "testdb" node and a "testdb" DB2 server running on 192.168.82.30 on port 50000.

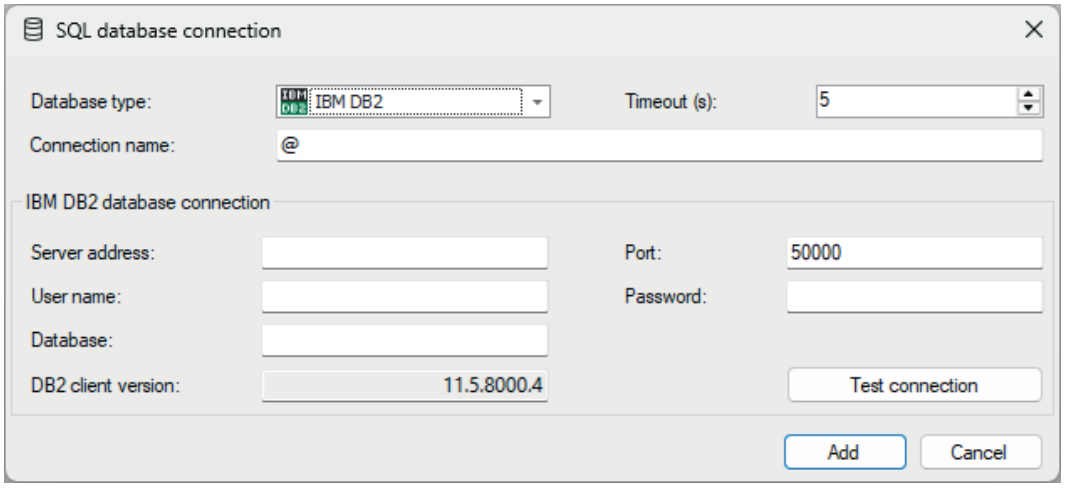
```
DB2 CLP - DB2COPY2
C:\Program Files\IBM\SQLLIB\BIN>db2 catalog tcpip node testdb remote 192.168.82.30 server 50000
DB20000I  The CATALOG TCPIP NODE command completed successfully.
DB21056W  Directory changes may not be effective until the directory cache is
refreshed.

C:\Program Files\IBM\SQLLIB\BIN>db2 catalog database testdb at node testdb
DB20000I  The CATALOG DATABASE command completed successfully.
DB21056W  Directory changes may not be effective until the directory cache is
refreshed.

C:\Program Files\IBM\SQLLIB\BIN>db2 terminate
DB20000I  The TERMINATE command completed successfully.

C:\Program Files\IBM\SQLLIB\BIN>db2 catalog tcpip node testdb remote 192.168.82.30 server 50000
```

3.5.2 Connection to IBM DB2 in ibaAnalyzer



From the *Data source* drop-down list, select *Create database connection*. In the *Database connection* dialog box, select the required database provider under *Database type*.

To edit an existing connection, select the connection from the *Data source* drop-down list. Then click on the editing button.

Enter the other settings.

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. Alternatively, you can enter and overwrite the connection name manually.

Timeout
Here you can specify a value for the timeout in seconds for establishing the connection. If the time set here is exceeded, *ibaAnalyzer* aborts the connection attempt and displays an error message.

Note that once the connection is made, each command sent to the database has an additional timeout, which can be set by the application that makes use of it. There is no generic configuration for the command timeout.

Server address
Enter the IP address or the host name of the database server.

Port
The DB server communicates via this port. Usually, you can keep the default port 50000. If a different port is required, you can enter this port here.

Username/Password
Enter here the required login data for the database. If uncertain, ask your DB administrator.

Database
Enter the database here. When you have entered the server address, authentication or username and password. The drop-down list shows all available databases for the connection and you can select one.

DB2 Client Version
ibaAnalyzer automatically displays the installed version of the DB2 client.

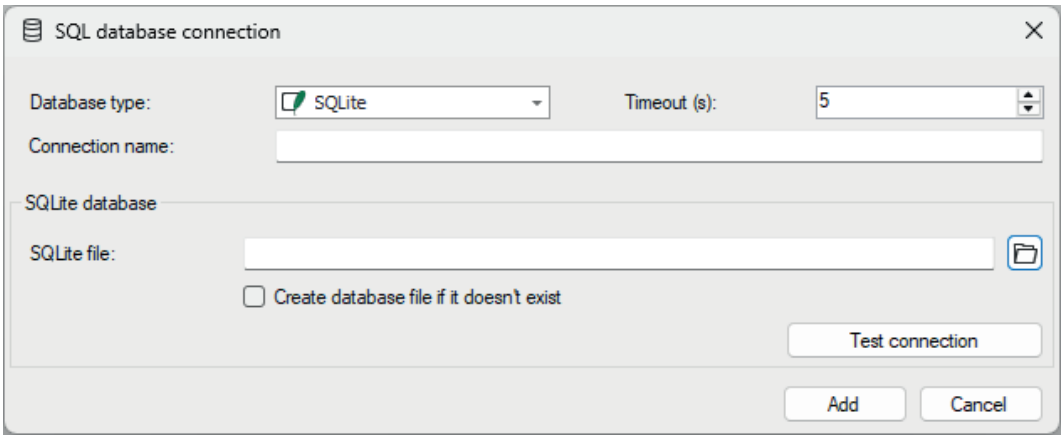
<Test connection>
ibaAnalyzer checks the access to the selected database on the set DB server. If the connection is successful, the message includes the version of the database.

3.6 Connection to SQLite

To connect *ibaAnalyzer* to an SQLite database, you do not need to make any special settings on the database side.

Note

ibaAnalyzer v8.0.0 or higher no longer supports any database connections via ODBC. Databases that have been configured in older PDO files via ODBC must be converted to the actual database provider in *ibaAnalyzer* and directly connected to the database.



From the *Data source* drop-down list, select *Create database connection*. In the *Database connection* dialog box, select the required database provider under *Database type*.

To edit an existing connection, select the connection from the *Data source* drop-down list. Then click on the editing button.

Enter the other settings.

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. Alternatively, you can enter and overwrite the connection name manually.

SQLite file
Enter the path to the database in this field or select the database using the browser button.

Create DB file if it is missing
If no SQLite database exists, enable this option. *ibaAnalyzer* then creates a corresponding database.

<Test connection>
ibaAnalyzer checks the access to the selected database on the set DB server. If the connection is successful, the message includes the version of the database.

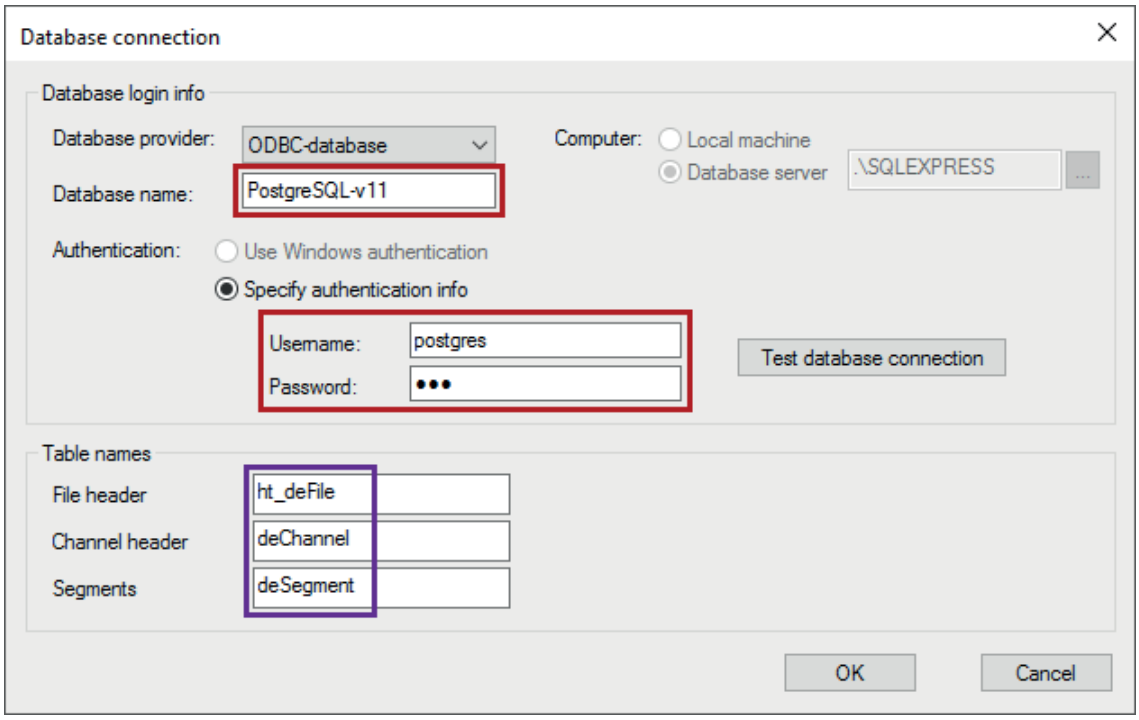
3.7 Changing the connection to ODBC database

Note

ibaAnalyzer v8.0.0 or higher no longer supports any database connections via ODBC. Databases that have been configured in older PDO files via ODBC must be converted to the actual database provider in *ibaAnalyzer* and directly connected to the database.

If you open an older PDO file of *ibaAnalyzer-v7*, in which an ODBC database connection has been configured, a message is shown that this connection is no longer supported. Therefore, you have to change the database connection in the PDO file. You make these changes for each PDO file with an ODBC database connection.

Database settings via ODBC in *ibaAnalyzer-v7* for comparison.

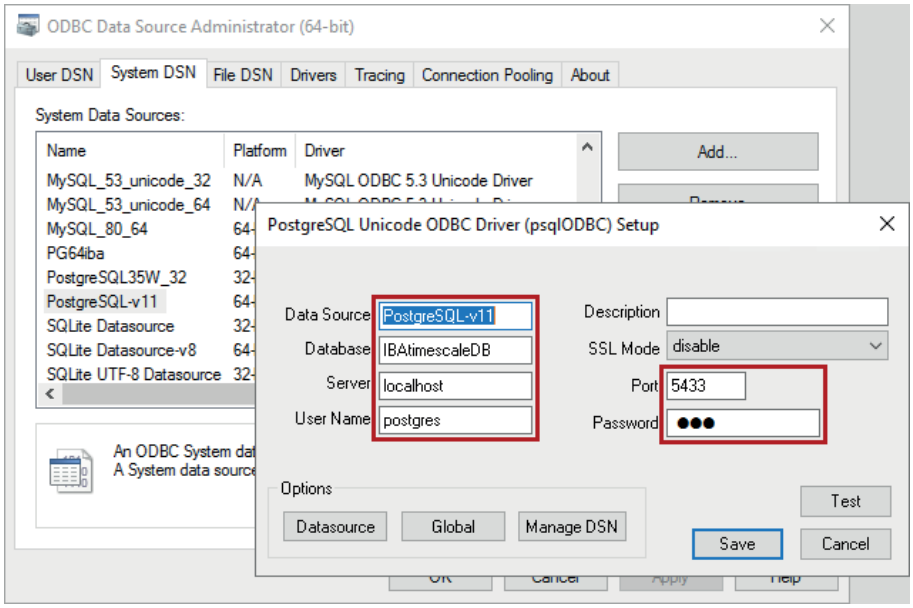


Customizing the database connection of PDO files with ODBC database connections

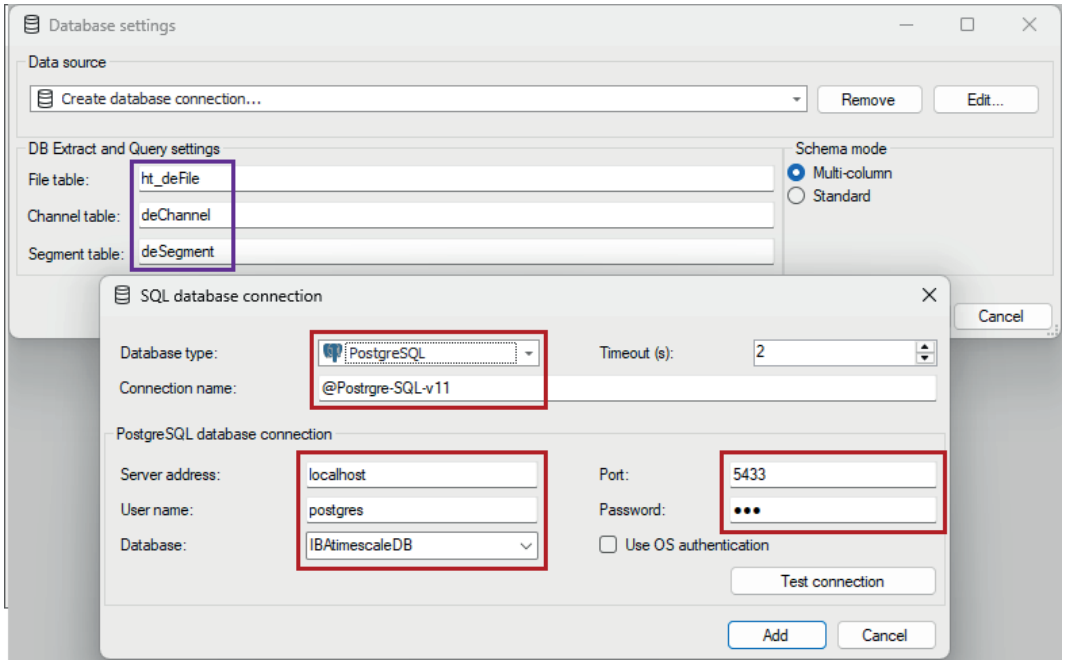
1. Open the PDO file from *ibaAnalyzer-v7* in *ibaAnalyzer* v8.1.0 or higher.
2. Open the *Database settings*.
3. From the *Data source* drop-down list, select *Create new connection*.

4. Configure the new connection with the settings for the original database.

→ To view the specific settings of the connection, you can open the ODBC settings under Windows.



→ *ibaAnalyzer* automatically applies the table names.



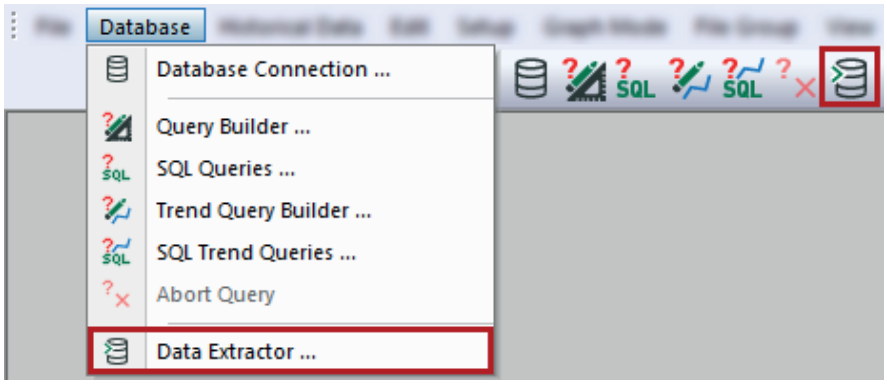
5. Apply the database settings and save the PDO file.

Further information on the settings for the specific databases can be found in the respective chapters:

- MySQL/MariaDB, see ↗ *Connection to MySQL/MariaDB, page 22*
- PostgreSQL, see ↗ *Connection to PostgreSQL, page 24*
- SQLite, see ↗ *Connection to SQLite, page 29*

4 Data Extractor

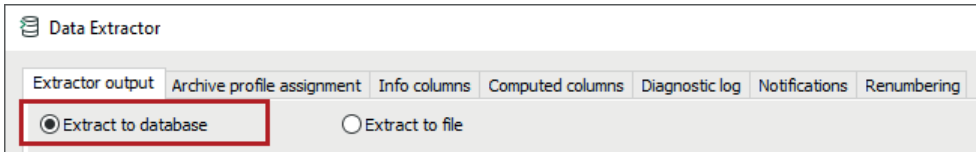
You can open the data extractor either via the *Database – Data Extractor* menu or via the corresponding button in the toolbar.



The *Extractor output* tab appears automatically corresponding to the database provider.

Settings for data extraction in the database

In the *Extractor output* tab, select the *Extract to database* option.



→ All tabs then adapt the respective settings to the data extraction into a database.

→ *ibaAnalyzer* automatically connects to the configured database.

If no database connection has been configured yet, configure a database connection using the <Database connection> button, see ↗ *Configuring the database connection*, page 12.

The following chapters describe the individual tabs of the data extractor with the *Extract to database option* selected.

Buttons on each tab

All tabs have the following buttons:

- <Extract now>
  ibaAnalyzer fills the database tables based on the current settings in the data extractor.
- <OK>
  ibaAnalyzer saves all current settings in the data extractor window and closes the window without filling the database tables.
- <Cancel>
  ibaAnalyzer discards all changed settings and closes the window.

4.1 Extractor output

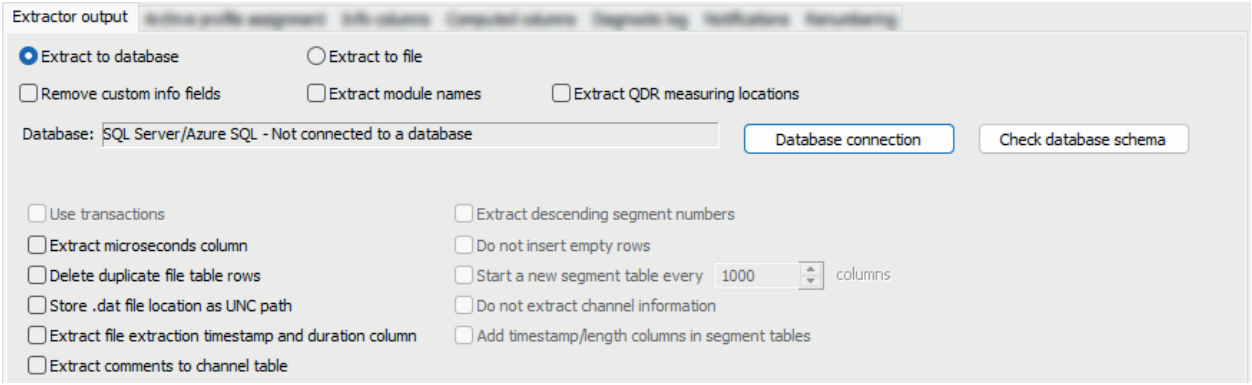
The *Extractor output* is automatically opened first when you open the data extractor. Depending on the selected schema mode, you have different options:

↗ *Standard schema mode, page 33*

↗ *Multi-column schema mode, page 35*

4.1.1 Standard schema mode

If the *Standard* schema mode is selected when configuring the database connection, the following settings and options are available.



Remove custom info fields
This feature is not supported for extraction into a database, but only for extraction into files. During extraction to databases, all desired info fields must be explicitly configured.

Extract module names
If you enable this option, the module names are extracted to the file table. The columns are named after the module numbers used in the DAT file (e.g., *Module_name_0*). *ibaAnalyzer* writes the name of the respective module in each column.
By default, this option is disabled to keep the number of columns in the file table to a minimum.

Extract QDR measuring locations
This option is only relevant when extracting QDR data. If you enable this option, *ibaAnalyzer* creates additional columns in the file table, which are numbered according to the numbering of the measuring locations.
Two columns are generated for each measuring location: Name of the measuring location and type of measuring location (e.g., *ML0.name* and *ML0.type*). In addition, *ibaAnalyzer* adds the *_MeasuringLocation* column to the channel table. This column lists the number of the corresponding measuring location for each channel.

Database
Displays the connected database provider.

<Database connection>
Use this button to open the *Database connection* window, see ↗ *Configuring the database connection*, page 12.

<Check database schema>
Use this button to compare the tables configured in the database connection with the actual database schema. In a dialog, you can check for each table which action is required (e.g., create, modify, or delete) and apply it as needed.

For more information about the dialog and the settings, see ↗ *Validation of the database schema, page 40*.

Note



You must follow the naming conventions of the connected database, e.g., avoiding special characters.

Note



Check the user rights. In environments with multiple users, it is recommended that only the application administrator is the object owner. Other users then have other database logins with appropriate restrictions (e.g. SELECT rights only).

In large, centralized multi-user DB environments (Oracle, DB2-UDB), the default DDL commands without storage options and simple indices may not be sufficient. In addition, the creation of database tables by <Create default tables> using *ibaAnalyzer* is not permitted. On such systems, the database administrator can create the default objects in a test environment and use reverse DDL script generators to create and customize their own DDL scripts. Depending on the integration scenario it might also be necessary to create further indices on info columns.

Use transactions – function not yet available
All SQL INSERT operations take place within one transaction. If an operation within the transaction fails, the whole transaction is aborted.

Extract microseconds column
If you enable this option, a *_TimeStampMicroSecs* column is created, which permits correct time stamp referencing on a microsecond basis. See also the item *_TimeStampMicroSecs* at ↗ *Database table "deFile", page 57*.

Delete duplicate file table rows
Enable this option to delete all previously saved files that have the same name as the file currently being extracted, see also the item *_FileName* at ↗ *Database table "deFile", page 57*.

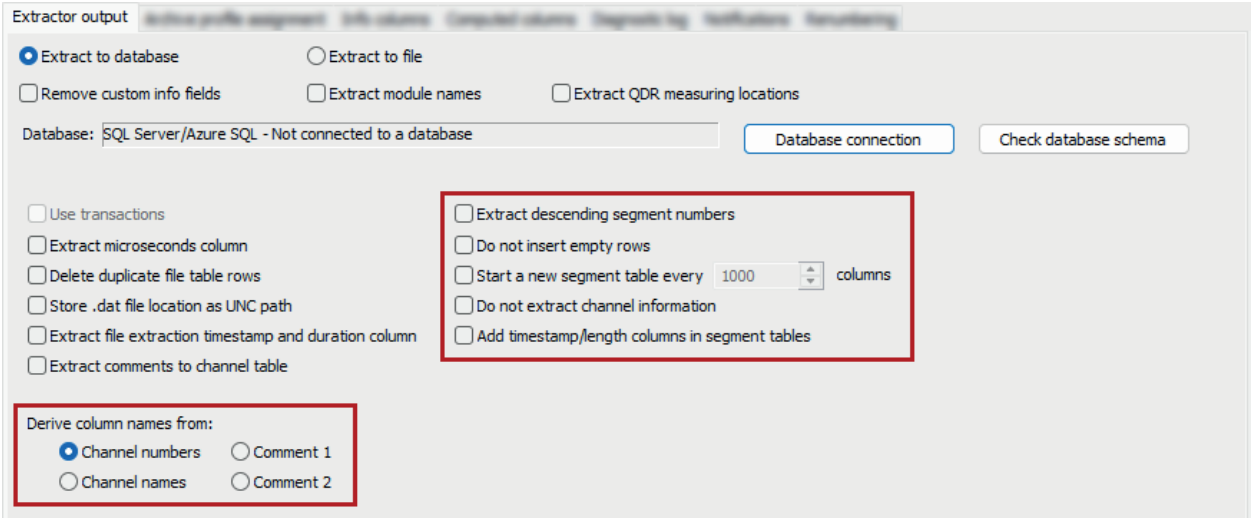
Save .dat-file path in UNC notation
Enable this option if DAT files are not stored locally.

Extract file extraction timestamp and duration column
If you enable this option, the time of the extraction and the duration in seconds are extracted into separate new columns.

Extract comments to the channel table
If you enable this option, the comments from *ibaPDA* or the expressions and logical signal definitions from *ibaAnalyzer* are extracted into separate new columns in the channel header.

4.1.2 Multi-column schema mode

If the *Multi-column* schema mode is selected when configuring the database connection, further options are available in addition to the settings for the *Standard* schema mode.



Remove custom info fields
This feature is not supported for extraction into a database, but only for extraction into files. During extraction to databases, all desired info fields must be explicitly configured.

Extract module names
If you enable this option, the module names are extracted to the file table. The columns are named after the module numbers used in the DAT file (e.g., *Module_name_0*). *ibaAnalyzer* writes the name of the respective module in each column.
By default, this option is disabled to keep the number of columns in the file table to a minimum.

Extract QDR measuring locations
This option is only relevant when extracting QDR data. If you enable this option, *ibaAnalyzer* creates additional columns in the file table, which are numbered according to the numbering of the measuring locations.
Two columns are generated for each measuring location: Name of the measuring location and type of measuring location (e.g., *ML0.name* and *ML0.type*). In addition, *ibaAnalyzer* adds the *_MeasuringLocation* column to the channel table. This column lists the number of the corresponding measuring location for each channel.

Database
Displays the connected database provider.

<Database connection>
Use this button to open the *Database connection* window, see ↗ *Configuring the database connection*, page 12.

<Check database schema>
Use this button to compare the tables configured in the database connection with the actual database schema. In a dialog, you can check for each table which action is required (e.g., create, modify, or delete) and apply it as needed.

For more information about the dialog and the settings, see ↗ *Validation of the database schema, page 40*.

Note

You must follow the naming conventions of the connected database, e.g., avoiding special characters.

Note

Check the user rights. In environments with multiple users, it is recommended that only the application administrator is the object owner. Other users then have other database logins with appropriate restrictions (e.g. SELECT rights only).

In large, centralized multi-user DB environments (Oracle, DB2-UDB), the default DDL commands without storage options and simple indices may not be sufficient. In addition, the creation of database tables by <Check database schema> using *ibaAnalyzer* is not permitted. On such systems, the database administrator can create the default objects in a test environment and use reverse DDL script generators to create and customize their own DDL scripts. Depending on the integration scenario it might also be necessary to create further indices on info columns.

Use transactions – function not yet available
All SQL INSERT operations take place within one transaction. If an operation within the transaction fails, the whole transaction is aborted.

Extract microseconds column
If you enable this option, a *_TimeStampMicroSecs* column is created, which permits correct time stamp referencing on a microsecond basis. See also the item *_TimeStampMicroSecs* at ↗ *Database table "deFile", page 57*.

Delete duplicate file table rows
Enable this option to delete all previously saved files that have the same name as the file currently being extracted, see also the item *_FileName* at ↗ *Database table "deFile", page 57*.

Save .dat-file path in UNC notation
Enable this option if DAT files are not stored locally.

Extract file extraction timestamp and duration column
If you enable this option, the time of the extraction and the duration in seconds are extracted into separate new columns.

Extract comments to the channel table
If you enable this option, the comments from *ibaPDA* or the expressions and logical signal definitions from *ibaAnalyzer* are extracted into separate new columns in the channel header.

Extract descending segment numbers
This option adds a new column *_ReverseSegmentNr* to the segment table, which numbers the segments in reverse order.

	_FileId	_SegmentNr	_ReverseSegmentNr	C33155	C33157
1	-374350715	0	5999	217.2715	190.2763
2	-374350715	1	5998	238.0183	196.4841
3	-374350715	2	5997	219.696	182.4169
4	-374350715	3	5996	226.9127	182.002
5	-374350715	4	5995	241.7527	185.6307
6	-374350715	5	5994	223.1701	192.2534

Do not insert empty rows
If you enable this option, segments containing only "NULL" are ignored during extraction.

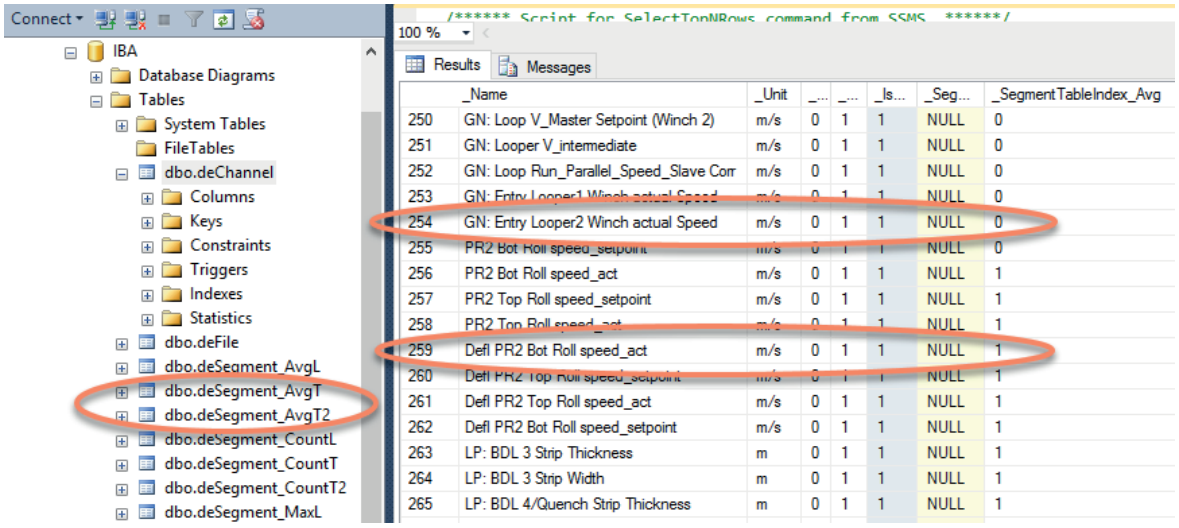
Disabled

	_FileId	_SegmentNr	C536870913
1	-1193539364	0	NULL
2	-1193539364	1	NULL
3	-1193539364	2	NULL
4	-1193539364	3	27.74264
5	-1193539364	4	27.73261

Enabled

	_FileId	_SegmentNr	C536870913
1	-1193363690	3	27.74264
2	-1193363690	4	27.73261

Start a new segment table every x columns
In multi-column format, database restrictions may make additional segment tables necessary. For example, in Oracle the maximum number of columns is limited to 1000. If more signals are to be extracted, further segment tables need to be created. If the specified number (x >99) of columns is reached, the extractor automatically creates new segment tables. This option also extends the channel table by new reference columns. In the example below, "0" in the column *SegmentTableIndex_Avg* indicates that channel 254 can be found in table "dba.deSegment_AvgT". "1" indicates that channel 259 can be found in table "dba.deSegment_AvgT2".



	_Name	_Unit	_...	_...	_Is...	_Seg...	_SegmentTableIndex_Avg
250	GN: Loop V_Master Setpoint (Winch 2)	m/s	0	1	1	NULL	0
251	GN: Looper V_intermediate	m/s	0	1	1	NULL	0
252	GN: Loop Run_Parallel_Speed_Slave Corr	m/s	0	1	1	NULL	0
253	GN: Entry Looper1 Winch actual Speed	m/s	0	1	1	NULL	0
254	GN: Entry Looper2 Winch actual Speed	m/s	0	1	1	NULL	0
255	PR2 Bot Roll speed_setpoint	m/s	0	1	1	NULL	0
256	PR2 Bot Roll speed_act	m/s	0	1	1	NULL	1
257	PR2 Top Roll speed_setpoint	m/s	0	1	1	NULL	1
258	PR2 Top Roll speed_act	m/s	0	1	1	NULL	1
259	Defl PR2 Bot Roll speed_act	m/s	0	1	1	NULL	1
260	Defl PR2 Top Roll speed_setpoint	m/s	0	1	1	NULL	1
261	Defl PR2 Top Roll speed_act	m/s	0	1	1	NULL	1
262	Defl PR2 Bot Roll speed_setpoint	m/s	0	1	1	NULL	1
263	LP: BDL 3 Strip Thickness	m	0	1	1	NULL	1
264	LP: BDL 3 Strip Width	m	0	1	1	NULL	1
265	LP: BDL 4/Quench Strip Thickness	m	0	1	1	NULL	1

Note

A value of 0 (zero) in the segment table index denotes the reference to a segment table without counter at the end of its name. Values x > 0 represent a reference to segment tables with the number x+1 at the end of their names.

Note

Additional segment tables are created dynamically during the extraction, similar to the creation of new columns in segment tables.

For more information on segment tables, see ↗ *Segment tables in multi-column format, page 63*.

Do not extract additional information channel in DB
If you enable this option, no data is added to the channel table. This option is useful if the data is only used for providing other internal applications or if you want to save disk space in the database. In this case, however, it is no longer possible to view the data in its entirety using the standard queries in *ibaAnalyzer*.

Add time stamp and length columns to segment tables
If you enable this option, the time stamp or the length information for the respective measured value dataset is extracted into separate new columns in the segment table.

Derive column names from
A column is created for each channel in the segment tables with the channel number as the column name. Select here whether the column name should instead be formed from the channel name or one of the comments defined in *ibaPDA*.

Note



If you change the selection, additional channels will be created. When the database schema is checked, the existing channels are listed as optional changes and are not automatically deleted unless you use the <Start fresh> function.

If *Comment 1* is selected and a signal does not have a first comment, the channel name is used instead. If *Comment 2* is selected for a signal where "Comment 2" is not available, "Comment 1" is used. If "Comment 1" is also missing, the channel name is used.

Note

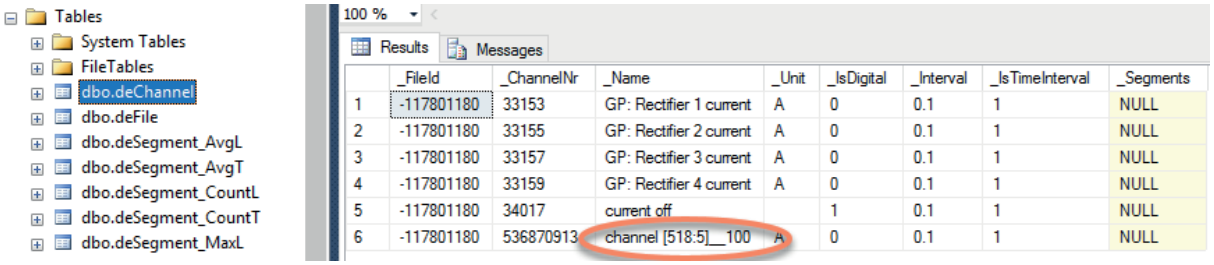


Note that comments are not stored separately in the database. If the column names are generated from one of the comments, the original channel name (signal name) is not transferred to the database and is lost.

Example: Derive column name from

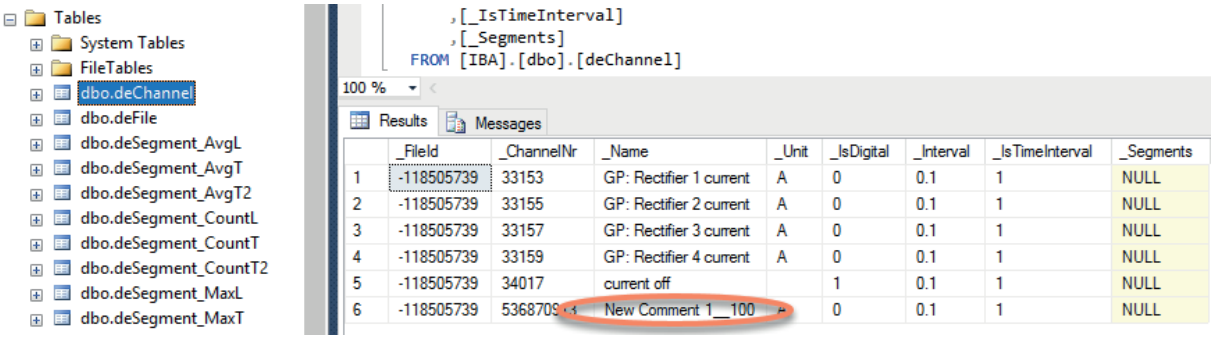
	Show	SignalName	Expression	Comment 1
1	☑	channel [518:5]	fx [18:7]	New Comment 1

Selection of *Channel number* or *Channel name*:



	_FileId	_ChannelNr	_Name	_Unit	_IsDigital	_Interval	_IsTimeInterval	_Segments
1	-117801180	33153	GP: Rectifier 1 current	A	0	0.1	1	NULL
2	-117801180	33155	GP: Rectifier 2 current	A	0	0.1	1	NULL
3	-117801180	33157	GP: Rectifier 3 current	A	0	0.1	1	NULL
4	-117801180	33159	GP: Rectifier 4 current	A	0	0.1	1	NULL
5	-117801180	34017	current off		1	0.1	1	NULL
6	-117801180	536870913	channel [518:5]__100	A	0	0.1	1	NULL

Selection of *Comment 1* or *Comment 2*:

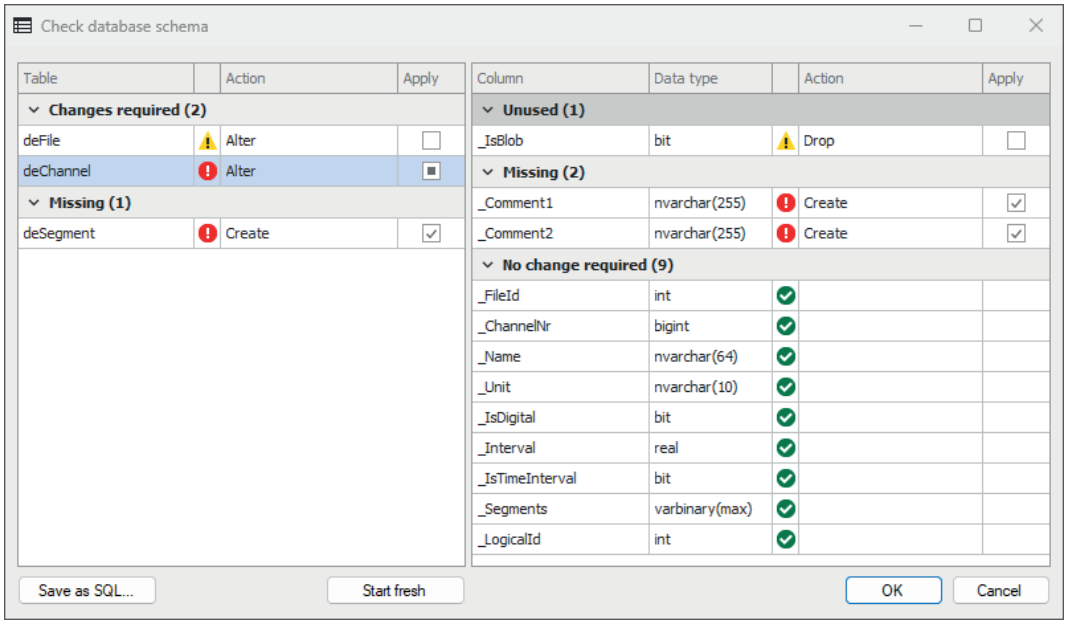


	_FileId	_ChannelNr	_Name	_Unit	_IsDigital	_Interval	_IsTimeInterval	_Segments
1	-118505739	33153	GP: Rectifier 1 current	A	0	0.1	1	NULL
2	-118505739	33155	GP: Rectifier 2 current	A	0	0.1	1	NULL
3	-118505739	33157	GP: Rectifier 3 current	A	0	0.1	1	NULL
4	-118505739	33159	GP: Rectifier 4 current	A	0	0.1	1	NULL
5	-118505739	34017	current off		1	0.1	1	NULL
6	-118505739	536870914	New Comment 1__100	A	0	0.1	1	NULL

4.1.3 Validation of the database schema

In the *Data Extractor*, you can also check the database schema on the *Extractor output* tab. To do this, click the <Check Database Schema> button. *ibaAnalyzer* compares the tables configured in the database connection with the actual database schema.

The required tables and columns are determined by your settings in *ibaAnalyzer*, that is, by the extracted channels and their archive profiles. During the check, *ibaAnalyzer* compares this required schema with the schema of the existing database and displays the differences in a dialog. There, you select the changes to apply, and *ibaAnalyzer* updates the database accordingly.



User interface of the Check database schema dialog

The *Check database schema* dialog displays all the tables in the database connection on the left and indicates which changes are required for each table (e.g., create, modify, or delete) to align the tables with the configuration in *ibaAnalyzer*.

The right side of the dialog shows the columns of the selected table and the necessary changes for each one.

Columns

- *Table*: Table names
- *Column*: Column names in the selected table
- Column with symbols: Required action for each change

✓	Green	no change
⚠	yellow	optional change
!	red	required change

- *Data type*: Data type of the column
- *Action*: Type of required change

- *Apply*: Tick the box to apply the change.
  If changes are required, the checkbox is always selected and cannot be deselected.
  For optional changes, the checkbox is not selected by default. You must actively decide whether you want to apply the change.

Buttons

At the bottom of the dialog box, you can find the following buttons:

- <Save as SQL>: Saves the active changes as an SQL file. This file can be used independently of the data extractor, for example, in an SQL editor. There, you can also add instructions that are not possible in *ibaAnalyzer*, such as custom calculated columns. This feature is also useful if you want to pass the changes on to the database administrator.
- <Start fresh>: Deletes all tables and recreates them. This will result in the loss of all data in the tables. The action is not executed immediately; instead, it first updates only the actions in the left area. It is not applied until you click <OK>.

Caution!



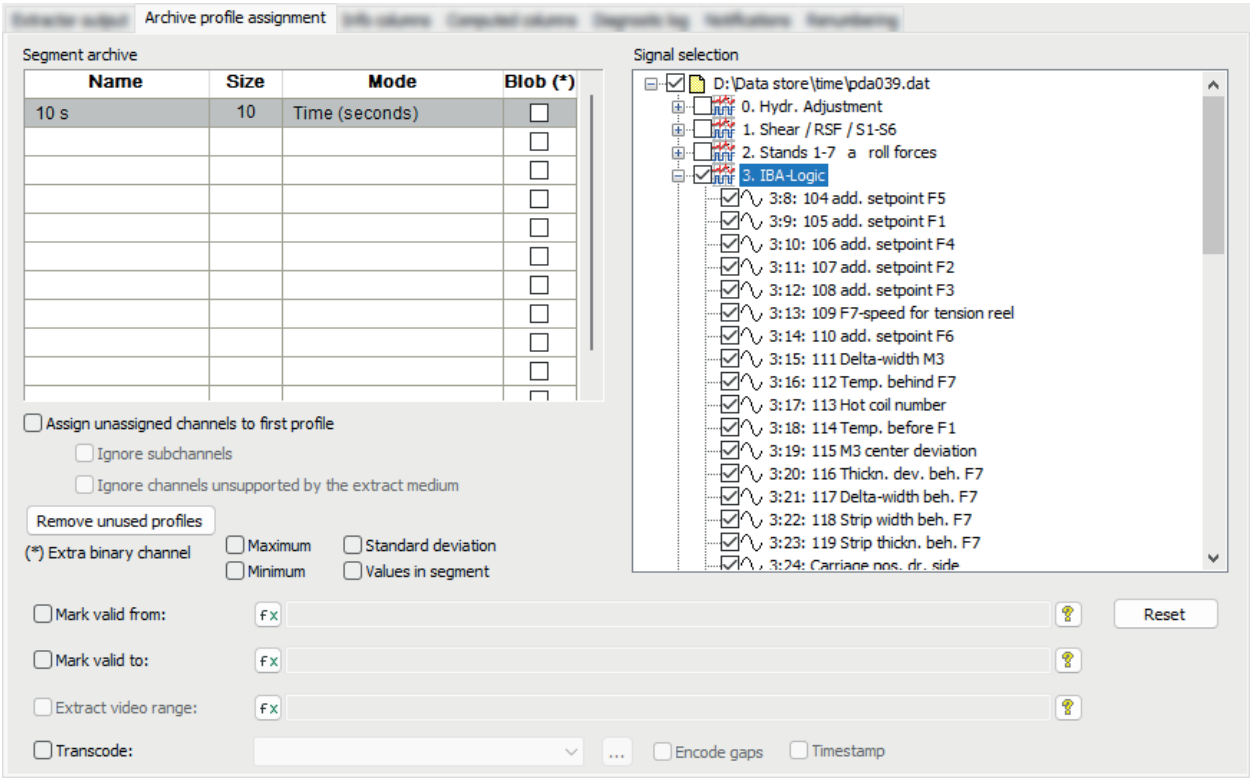
If you use this function, all existing data in the tables will be permanently deleted. Create a backup beforehand if necessary.

- <OK>: Applies all selected changes to the database tables.
- <Cancel:> *ibaAnalyzer* discards all changed settings and closes the window.

4.2 Archive profile assignment

In the *Archive profile assignment* tab, you can configure archive profiles that define the storage interval (timebased or lengthbased). You can then assign the profile to the signals for extraction. You can configure more than one profile, but you can only assign one profile to a signal.

To create an archive profile, at least one DAT file must be open for extraction.



Segment archive

Name column

Select a cell in the *Name* column and enter a profile name.

It is recommended to use a name which reflects the chosen storage interval for the extracted file, e.g. "10 s" = size (10) × mode (time (seconds)).

Mode and Size columns

Enter a multiplier in the *Size* column to determine the storage interval for the extracted data e.g. 10 s = "Size (10)" × "Mode (time (seconds))".

Click in the *Mode* column and select an extraction mode:

- For the *Time* and *Length* modes it is possible to enter a value < 1 to obtain fractions of the unit under *Mode*.
  When entering the fraction factor for values < 1, make sure a point is entered instead of a comma as the decimal separator (e.g. "0.5" instead of "0,5").
- The following modes use the original base of the signal or a multiple of it, so fractions are not useful here. For example, a temperature that was measured every minute does not need to be extracted every 100 ms.

*_signal default base_: original base, either time or length, depending on the signal type

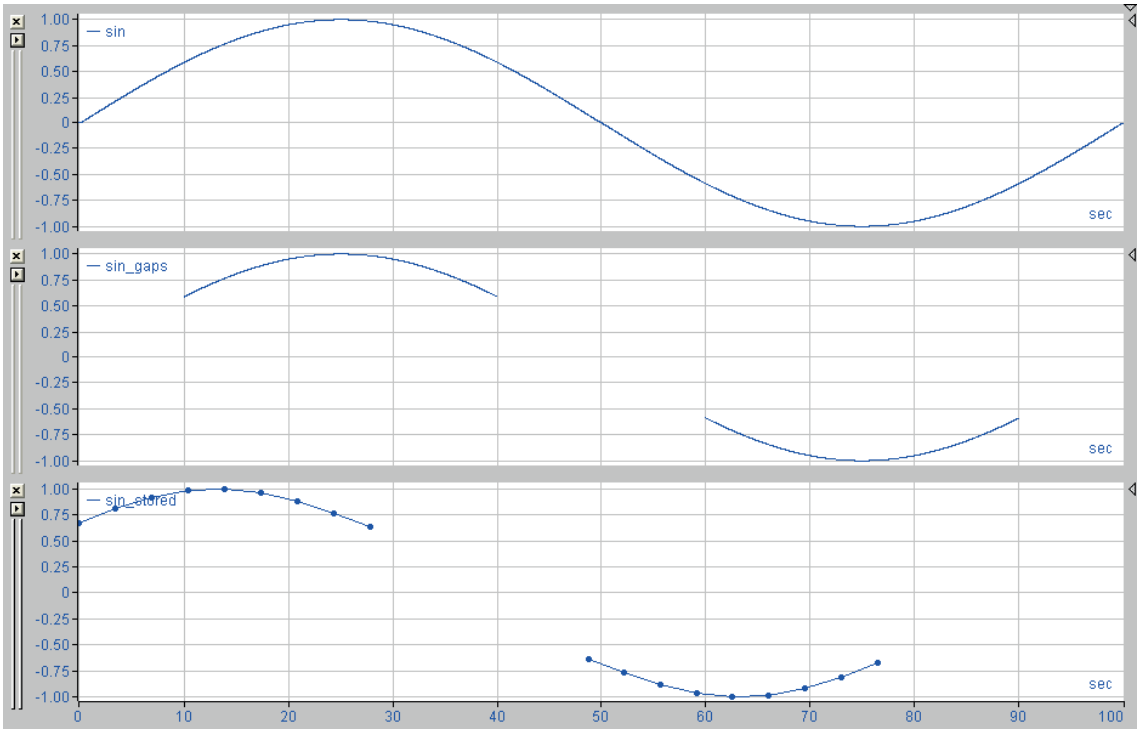
*_signal timebase_: original timebase

*_signal lengthbase_: original lengthbase

*_signal time- and lengthbase_: original time- and lengthbase; supported only for multi-column format, because at least two segment tables are written.

If the timebase of the recording signal is 1 ms, the 100 ms profile could also be defined as follows: *Size* = 100 and *Mode* = *_Signal timebase_*

- By using the *Time segments* or *Length segments* modes it is possible to define a fixed number of equidistant time or length segments. For example, *Size* = 1 and *Mode* = *Time segments* means that only 1 segment value is stored for the whole data channel.
  If parts of the source signal are invalid, the extracted segments containing the invalid parts are left blank. If the empty segments are at the beginning or end of the signal, they are not transferred to the extracted data. If the empty segments are between valid segments, the extracted data contains corresponding gaps with invalid values.



- The *Time, non-equidistant* mode is not supported for extraction into a database and leads to the following error message. This option only exists for extraction into text files.

Note

If you assign signals to a profile that do not match the selected mode, e.g., time signals for the "Length" mode, these signals will not be extracted into the database.

Blob column
If you enable this option, the data is saved as a Binary Large OBject (BLOB). For additional information, please see ↗ *Using BLOBs, page 61*.

Note

If BLOBs are used, the statistical values are only accessible by a user-defined decompression program and not by *ibaAnalyzer*!

The *Values in segment* option is also not available if BLOBs are used.

Assign unassigned channels to the first profile

If you enable this option, all signals that have not been manually assigned to a profile are automatically extracted with the first profile. However, this automatic assignment is not displayed in the signal assignment. This helps prevent errors in the extraction process if the file structure is modified across different files, e.g., when the DAT file is expanded in *ibaPDA*.

Ignore subchannels
If you enable this option, subchannels from an HD query are not extracted, e.g. the channels min/max.

Ignore channels unsupported by the extract medium
If you enable this option, channels that the destination file format does not support are not extracted, e.g. *ibaCapture* synchronization channels for non-iba formats.

<Remove unused profiles>

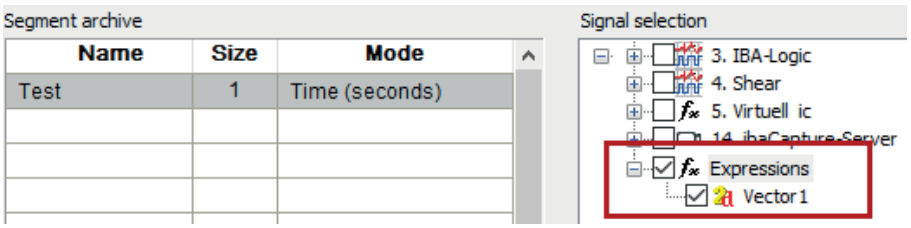
Use this option to automatically delete profiles that have no signals assigned to them.

Input signals

You have to assign the signals to the created profiles. Select the signals to be included in the extracted data. The extracted signal values are the averages of the source signal values within the selected storage intervals.

When a module node is enabled or disabled, all signals that belong to this module are also enabled or disabled. Individual signals can be enabled or disabled in the open module node. The selected signals are always assigned to the highlighted profile (gray background color). When changing the profile, the associated selected signals also change. You can only assign one profile to a signal.

You can extract and query also multidimensional expressions (virtual signals, logical expressions and vectors from a DAT file) from there. A new column *_LogicalId* (*i_LogicalId* for Oracle and IBM DB2 databases) then appears in the channel header of the database to show which subchannels belong to the same multidimensional channel.

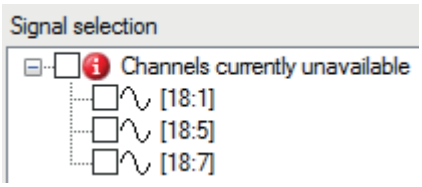


Note

Assigning a length mode to a timebased signal or a time mode to a lengthbased signal will result in an error during extraction.

Note

If the archive profile specifies signals that are not present in the current analysis file, these are listed under *Channels currently unavailable*.



Additional channel information: Maximum, minimum or standard deviation

The options for *Maximum*, *Minimum*, *Standard deviation* and *Values in segment* are only available for the *Multi-column* schema mode, see ↗ *Multi-column schema mode, page 35*. In *Standard* schema mode these values are automatically extracted into the segment table.

If you enable this option, subchannels are generated, each representing the maximum, minimum, or standard deviation of the source signal values within the selected storage intervals. This option can be helpful for newly recorded signals.



Note

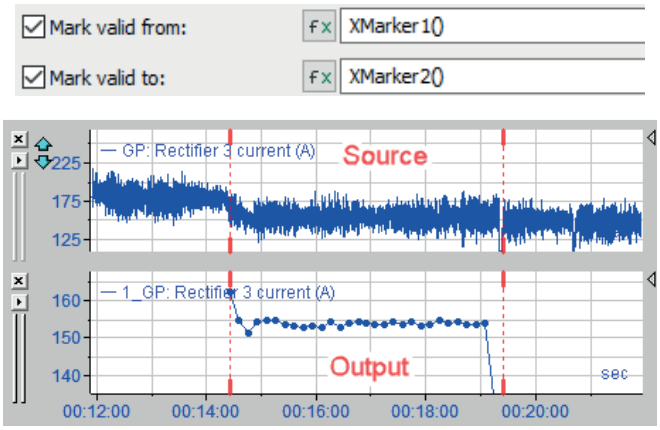
If the selected storage interval of the profile is identical to that of the source signals, these values are irrelevant.

Additional channel information: Values in segment

If you enable this option, the *_ValuesInSegment* column is created in the database. This column contains the number of data points that are compressed in each segment in the database. This option is only available for the *Multi-column* schema mode and if the profile is not extracted as BLOB, see above.

Mark valid from/Mark valid to

If you do not want to extract the signal over the full duration, you can define the start and end of the profile range here. To calculate the range from your data, you can use expressions.



If neither of the two expressions is defined despite the options being enabled, the signal is extracted in full.

If the specified expressions cannot be evaluated, an error message appears during the extraction.

Extract video range/Transcode

These functions are not supported for extraction into a database but only for extraction into files.

<Reset>

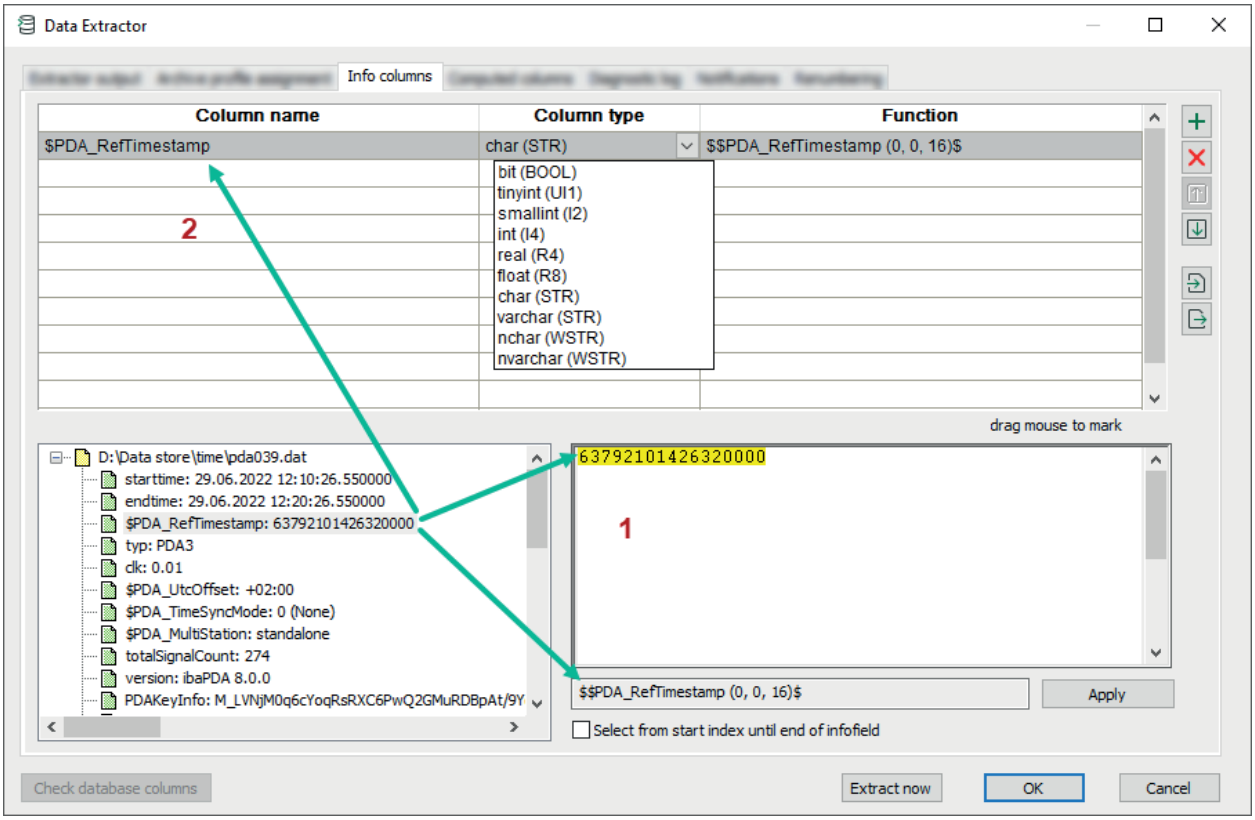
Delete all signals from the selected profile.

4.3 Info columns

Use the *Info columns* tab to create info columns in the database table. You can extract additional information from the DAT files into these columns. The data is located in the "deFile" file table in the database.

Using info fields

You have two options for adding info fields to the table so you can extract them.



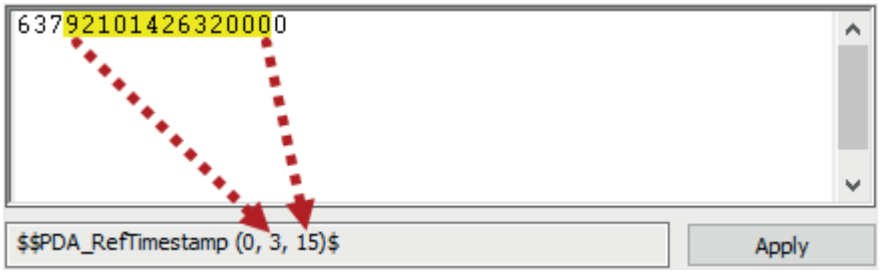
First select a line for the field information in the table (2).

- Click on the source field. The field information is transferred to the processing field (1).
  Select the range of characters with the mouse and click on <Apply>.
  Your selection is transferred to the table (2) and the processing field. The *Function* column shows the character selection.
- Double-click on the source field.
  The field information is completely transferred to the table (2) and the processing field (1).

Selecting a character range

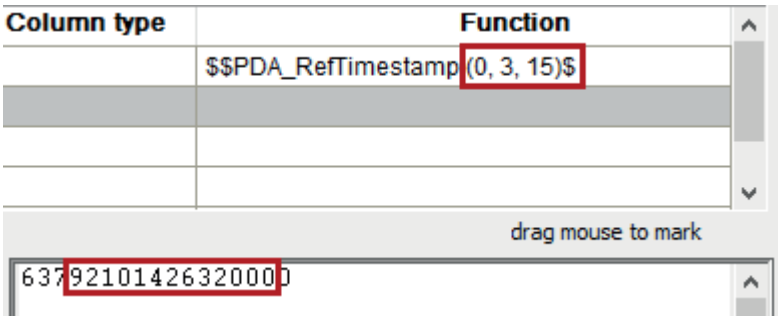
You can also select a range of characters from the contents of the info field.

1. Click the first required character, drag the selection to the last required character.



2. Click on <Apply>.

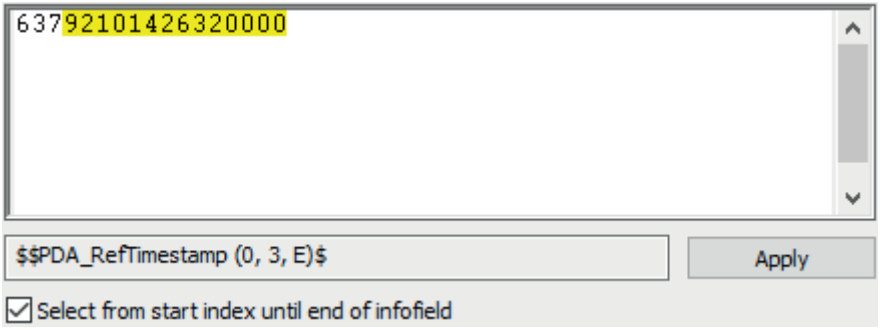
→ The excerpt from the source string is transferred to the column field.



Alternatively, you can also enter the position of the first and last character manually in the *Function* column.

Select from start index to end of info field

If you enable this option, all characters from the selected start point to the end of the info field are selected.



Customizing the info fields in the table

In the table, you can manually rename the *Column name*. Please note the restrictions for the column names of the database.

CAUTION

The names must comply with the naming conventions of the used database system. Avoid spaces, hyphens, special characters etc. and check the permitted length.
When highlighting the characters in the processing field, make sure that the correct row is selected in the table above. The last highlighted excerpt before exiting the row or closing the window is valid and will be saved.

The *Column type* can be selected from the drop-down list, which contains the available data types of the currently active database provider.

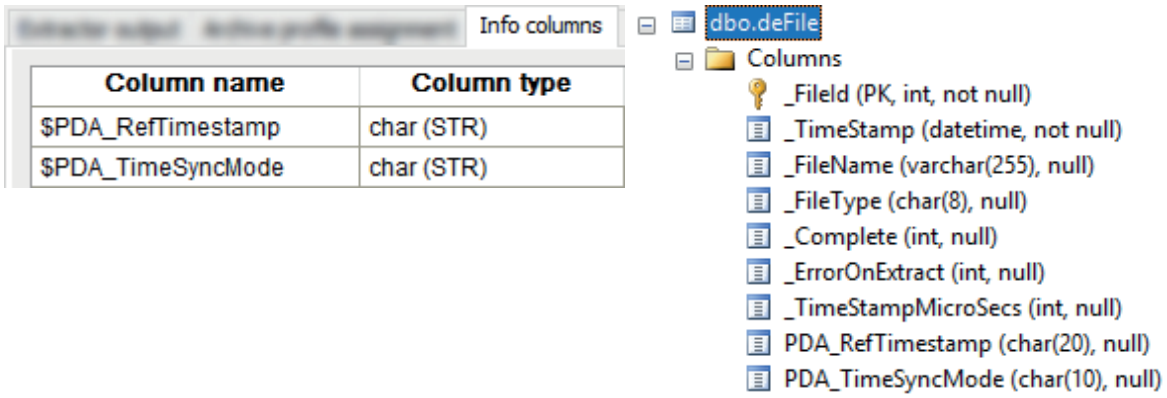
The *Function* characters can be changed in the processing window.

For further adjustments, see ↗ *Editing the info columns, page 50*.

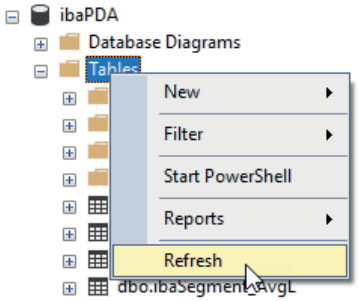
<Check database columns>

This button opens the dialog for checking the database schema, see ↗ *Validation of the database schema, page 40*. After the info columns have been configured, it is recommended to synchronize these definitions with the columns in the file table. If the desired column does not exist, you have to confirm its creation.

Location of the info columns in the database



Remember to refresh the tables in the database for display e.g. in SQL Server.



4.3.1 Editing the info columns

Using the buttons on the side, you can change the order of info columns or import or export information about the columns.

	Add line	Add a new empty line above the currently selected line.
	Delete line	Delete the currently selected line.
	Move line up/down	Move currently selected line up/down.
	Import info file	Import column information from text file. The *Open text file* dialog appears. **Alternative:** Right-click on the table header and select *Import*. **Attention:** An import always overwrites all current columns.
	Export info file	Export column information to text file. The *Save text file* dialog appears. **Alternative:** Right-click on the table header and select *Export*.

Tip

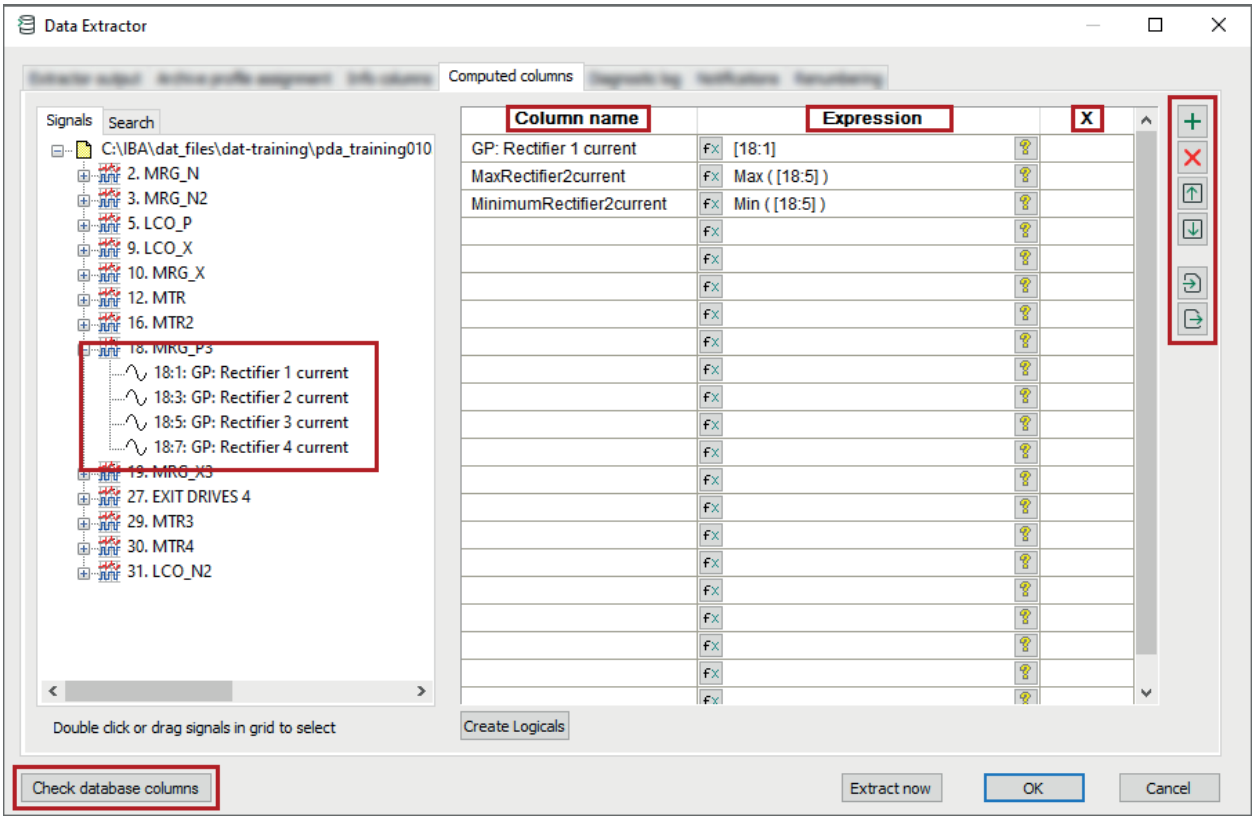


The import-export-function is very helpful if you want to use the same expressions in different places, e.g. for the database extraction or when handling large numbers of computed columns.

The exported text file is easy to edit using an editor or MS Excel.

4.4 Computed columns

The *Computed columns* tab is used to specify operations for the individual signals. These operations create one value for the associated signal (e.g. the average, maximum or minimum of all data of one signal). These values are defined as Computed Columns and are saved in the database file table, see ↗ *Database table "deFile", page 57*. Therefore, you can use the values as part of a query condition or for long-term trend analyses.



Selecting and preparing signals

Add the desired signal to the table either by double-clicking on the signal or using drag & drop. You can also select multiple signals in the signal tree by the standard Windows method of a mouse click + <Shift> or <Ctrl>, and drag them into the table with the mouse.

The *Column name* column initially contains the source signal's name, which you can change manually.

The *Expression* column initially contains the channel number of the source signal. You can either enter an expression or use the <fx> button to open the expression builder. The result of the expression should be a signal value.
If you do not use an operation or group function for *Expression*, then *ibaAnalyzer* automatically extracts the average (AVG) from the signal data.

The *X* column automatically shows the X-axis mode of the source signal (time-based or length-based).

You can edit the computed columns in the same way as the info columns, see ↗ *Editing the info columns, page 50*.

<Create Logicals>
At different places in *ibaAnalyzer*, you can define and manage logical expressions. This button creates a group of logical expressions from the defined computed columns. You can then manage the created logical expressions via the *Logical signals* dialog. When renaming or moving the expressions, all references to them are automatically adapted.

If you have already used the same expressions or the same names elsewhere in *ibaAnalyzer* and they are not yet grouped, you can select how you want to proceed with these expressions: ignore, overwrite other expression or replace this expression with the other one.

<Check database columns>
This button opens the dialog for checking the database schema, see ↗ *Validation of the database schema, page 40*. After the info columns have been configured, it is recommended to synchronize these definitions with the columns in the file table. If the desired column does not exist, you have to confirm its creation.

Special feature: Text signals

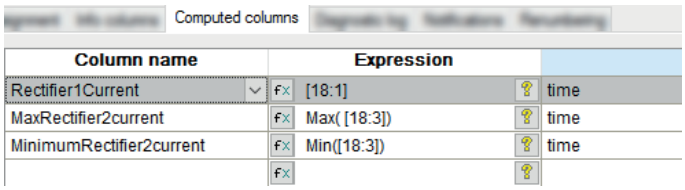
Text signals can also be used in computed columns. A column with the name of the text signal and its first value is then created in the file table.

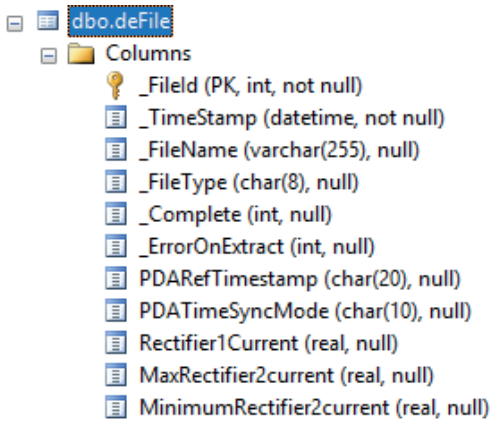
If you want to extract a specific value from a text signal with several values in the DAT file, you have to select this value explicitly using a suitable signal expression.

Example: Selection of the third value (counting starts at 0)

```
XCutRange([text signal], XFirst([text signal],2),
XFirst([text signal],2) + 0.01)
```

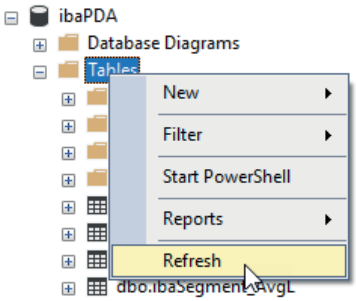
Location of the computed columns in the database





Additionally, for each computed column, a row is created in the file metadata table containing the name, comments, expression, etc., see ↗ *Database table "deFileMeta", page 59*.

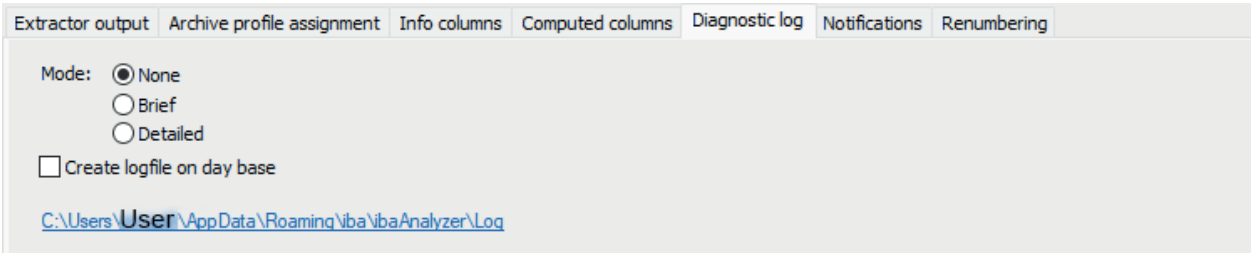
Remember to refresh the tables in the database for display e.g. in SQL Server.



4.5 Diagnostic log

The diagnostic log is useful for checking progress when automatic extraction is operating (e.g. triggered by *ibaDatCoordinator*). If you select a log mode, *ibaAnalyzer* creates a log file and updates it with each extraction.

The linked path takes you to the directory containing the log files.



Mode

- *None*: *ibaAnalyzer* does not create log files of the extraction.
- *Brief*: The log file only contains the filenames of the extracted files.

```
24-May-17 13:45:54: Start extract file 'D:\IBA\dat files\dat-training\pda_training021.dat'
24-May-17 13:46:57: Start extract file 'D:\IBA\dat files\dat-training\pda_training021.dat'
```

- *Detailed*: The log file contains the filenames of the extracted files and individual channels.

```
24-May-17 13:16:10: Start extract file 'D:\IBA\dat files\dat-training\pda_training021.dat'
24-May-17 13:16:10: Start extraction to file 'd:\dat\pda_training021_de.dat'
24-May-17 13:16:10: Start extract channel 18:1 GP: Rectifier 1 current with profile 100ms
24-May-17 13:16:10: Start extract channel 18:3 GP: Rectifier 2 current with profile 100ms
24-May-17 13:16:10: Start extract channel 18:5 GP: Rectifier 3 current with profile 100ms
24-May-17 13:16:10: Start extract channel 18:7 GP: Rectifier 4 current with profile 100ms
24-May-17 13:16:10: Extract completed
24-May-17 13:16:35: Start extract file 'D:\IBA\dat files\dat-training\pda_training021.dat'
24-May-17 13:16:35: Start extraction to file 'd:\dat\pda_training021_de_00.dat'
24-May-17 13:16:35: Start extract channel 18:1 GP: Rectifier 1 current with profile 100ms
24-May-17 13:16:35: Start extract channel 18:3 GP: Rectifier 2 current with profile 100ms
24-May-17 13:16:35: Start extract channel 18:5 GP: Rectifier 3 current with profile 100ms
24-May-17 13:16:35: Start extract channel 18:7 GP: Rectifier 4 current with profile 100ms
24-May-17 13:16:35: Extract completed
```

Create logfile on day base

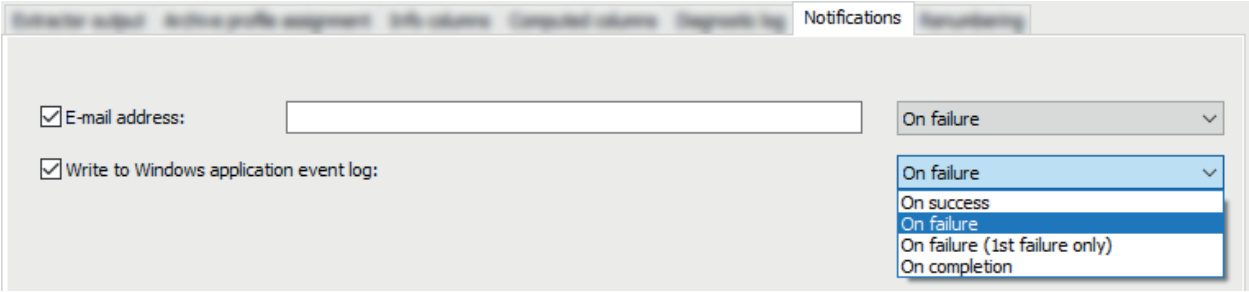
ibaAnalyzer creates a log file for each day, e.g. `DataExtractorLog_23_02_2026.txt`.

Note

If logging is activated permanently, you have to implement a cleanup strategy for the log files externally (not part of *ibaAnalyzer*).

4.6 Notifications

On the *Notifications* tab, you can choose between two communication methods how to receive notification or record events.

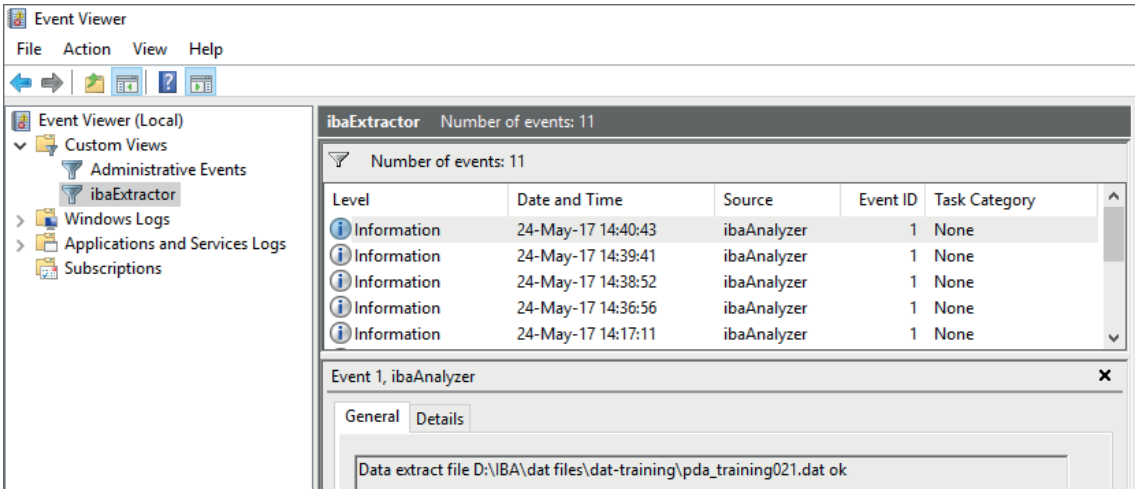


E-mail address
Enter the address and select the send mode.

Note

To send e-mails, a standard e-mail client (e.g. MS Outlook, Outlook Express, etc.) must be installed on the computer.

Write to Windows application event log
Events are shown in the Windows Event Viewer. Select the send mode.



Send modes for each communication method

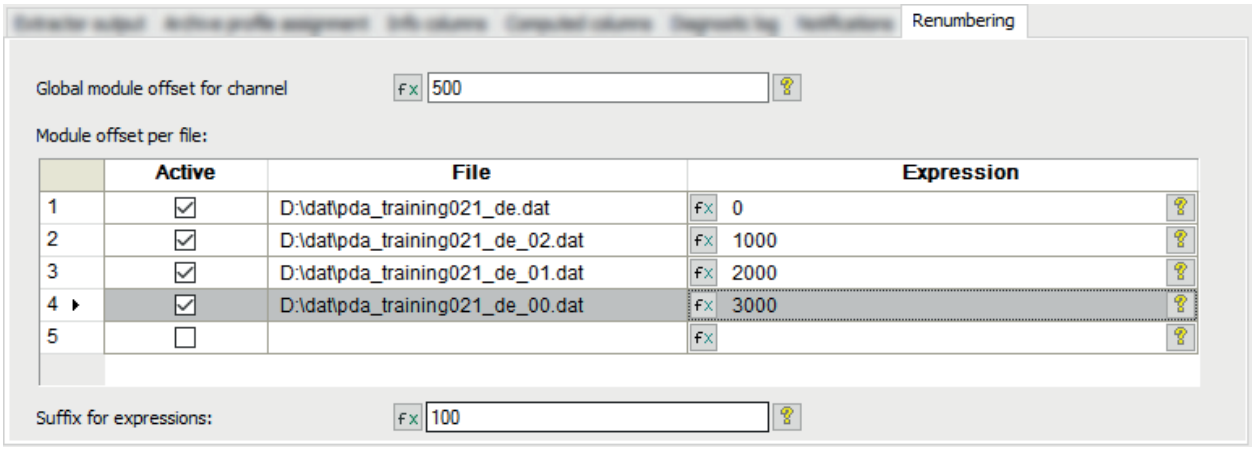
- *On success*
  Notification when extraction is successful.
- *On failure*
  Notification when any extraction has failed.
- *On failure (1st failure only)*
  Notification when the extraction has failed for the first time.
- *On completion*
  Notification when the extraction has been completed, success or failure is irrelevant.

4.7 Renumbering

If multiple files are opened and you want to extract several signals from several files, it is likely that some channels have the same ID number. The channels therefore need to be renumbered, otherwise it is not possible to uniquely identify them in the exported media. *ibaAnalyzer* can perform this renumbering automatically, but issues an alert before extraction if this is necessary.

The channel IDs consist of a module number and a position within the module. *ibaAnalyzer* uses an internal algorithm to create the channel IDs. For this reason, the values are also not directly recognizable during renumbering in the database column.

The *Renumbering* tab allows the specification of an offset to the module numbers for each file, hence enabling the user to prevent ID collisions and avoid automatic renumbering.



Global module offset for channel numbering
This number is added to DAT file channel numbers.

Active column
Only active files obtain a channel offset. Extraction of channels in inactive files results in a notification.

File column
The open DAT files are automatically listed in this column.

Expression column
The offset for each DAT file is entered in this field.

Note

The original and offset channel numbers are not visible in the database. They can only be seen in *ibaAnalyzer* after database queries have been carried out, see ↗ *Analysis of data from databases, page 67*.

Suffix for expressions
This applies only to the expressions, names and comments created in *ibaAnalyzer*. The specified suffix is appended to the expression names.

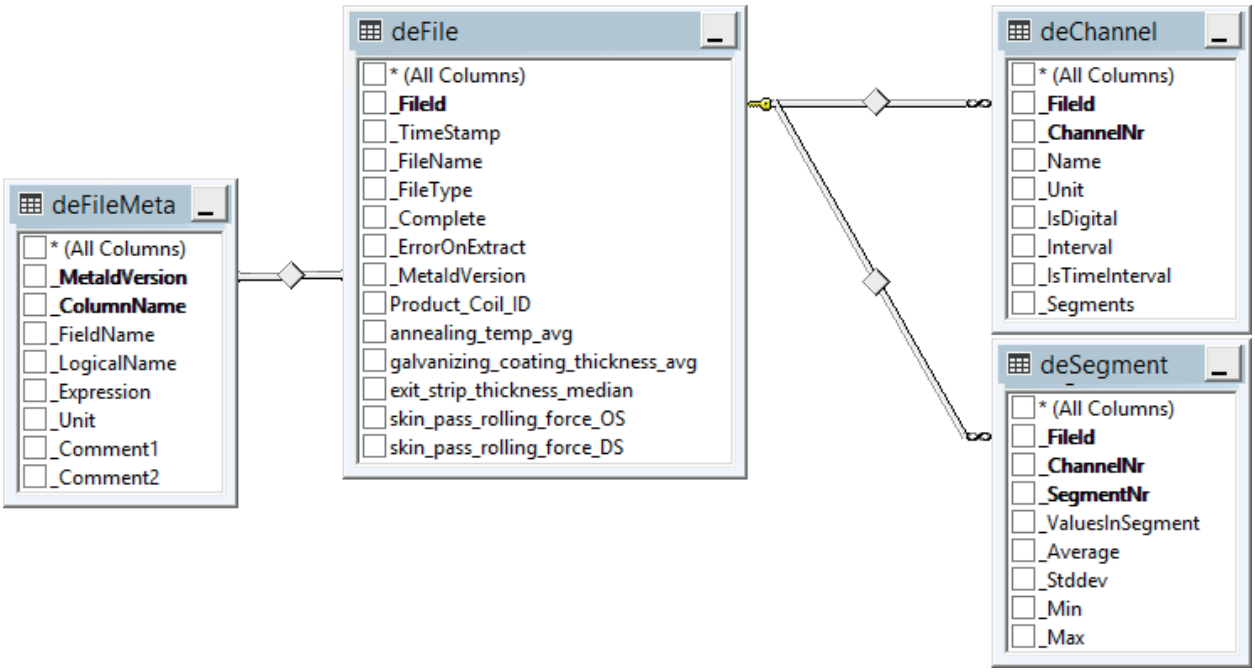
Example of extracted data with renumbering

Extracted data with renumbering (based on the settings above) is displayed in the database as shown in the following figure. *ibaAnalyzer* uses an internal algorithm to create the channel IDs. For this reason, the values are also not directly recognizable during renumbering in the database column.

	_FileId	_ChannelNr	_Name	_Unit	_IsDigital	_Interval	_IsTimeInterval	_Segments
1	-1049613619	4060	GP: Rectifier 1 current	A	0	0.1	1	NULL
2	-1049613619	4061	GP: Rectifier 2 current	A	0	0.1	1	NULL
3	-1049613619	4062	GP: Rectifier 3 current	A	0	0.1	1	NULL
4	-1049613619	4063	GP: Rectifier 4 current	A	0	0.1	1	NULL
5	-1049613619	33153	GP: Rectifier 1 current	A	0	0.1	1	NULL
6	-1049613619	33155	GP: Rectifier 2 current	A	0	0.1	1	NULL
7	-1049613619	33157	GP: Rectifier 3 current	A	0	0.1	1	NULL
8	-1049613619	33159	GP: Rectifier 4 current	A	0	0.1	1	NULL
9	-1049613619	161153	GP: Rectifier 1 current	A	0	0.1	1	NULL
10	-1049613619	161155	GP: Rectifier 2 current	A	0	0.1	1	NULL
11	-1049613619	161157	GP: Rectifier 3 current	A	0	0.1	1	NULL
12	-1049613619	161159	GP: Rectifier 4 current	A	0	0.1	1	NULL
13	-1049613619	225153	GP: Rectifier 1 current	A	0	0.1	1	NULL
14	-1049613619	225155	GP: Rectifier 2 current	A	0	0.1	1	NULL
15	-1049613619	225157	GP: Rectifier 3 current	A	0	0.1	1	NULL
16	-1049613619	225159	GP: Rectifier 4 current	A	0	0.1	1	NULL
17	-1049613619	536870913	_GP: Max Rectifier 1 current__100		0	0.1	1	NULL
18	-1049613619	536870914	_GP: Min Rectifier 2 current__100		0	0.1	1	NULL
19	-1049613619	536870915	_GP: Max Rectifier 3 current__100		0	0.1	1	NULL
20	-1049613619	536870916	_GP: Max Rectifier 4 current__100		0	0.1	1	NULL

5 Database format options

The following database structure is automatically generated when default tables are created.



5.1 Database table "deFile"

The file table "deFile" contains the following header information for each extracted dataset.

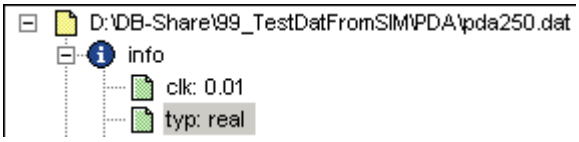
A new dataset is created for each extraction. The handling of identical source files is described in the *Delete duplicate file rows* option, see ↗ *Standard schema mode*, page 33.

_FileID
This is a unique internal ID assigned to each dataset.

_TimeStamp
The second column contains the date and start time of the acquisition.

_FileName
This column refers to the file name of the DAT file. This is used for information or as a 'path-name' to display datasets in *ibaAnalyzer*.

_FileType
This column specifies the file type and refers to the recording system.
Examples: "real" → *ibaPDA* record, "ibaQDR" → *ibaQDR* record



_Complete
This column shows the status of the extraction process. The first dataset of a file extraction is always inserted into the file table ("deFile") with status "_Complete = 0". After finishing all other inserts in the channel headers and segment tables, the flag is updated to "_Complete = 1".

When querying the "deFile" table, it is recommended that the WHERE condition checks this complete flag to see whether the corresponding extraction was successful.

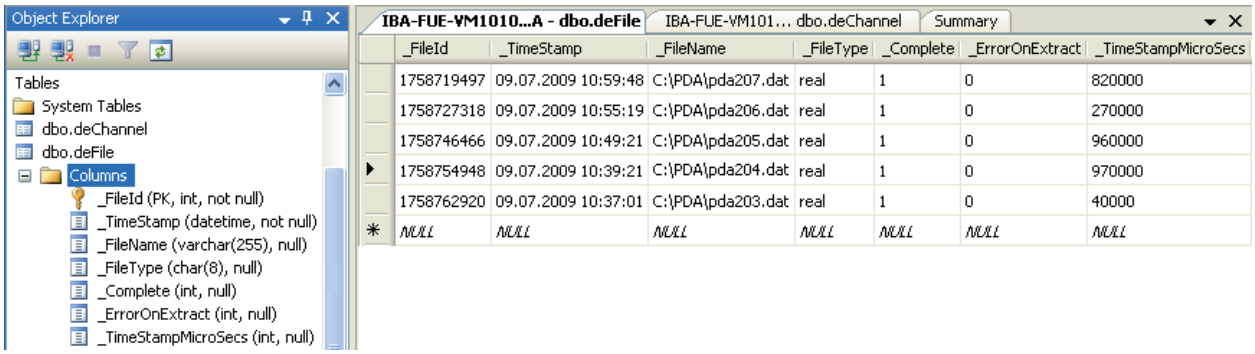
_ErrorOnExtract
This flag is set if an extraction error occurs.

_TimeStampMicroSecs
This column is created by enabling the *Extract microseconds column* in the *Extractor output* tab, see ↗ *Standard schema mode*, page 33.

Note

In many database systems there are data types that can process microsecond resolution. A separate column exists for the purpose of maintaining backward compatibility.

The "deFile" is displayed in the database as shown in the following figure.

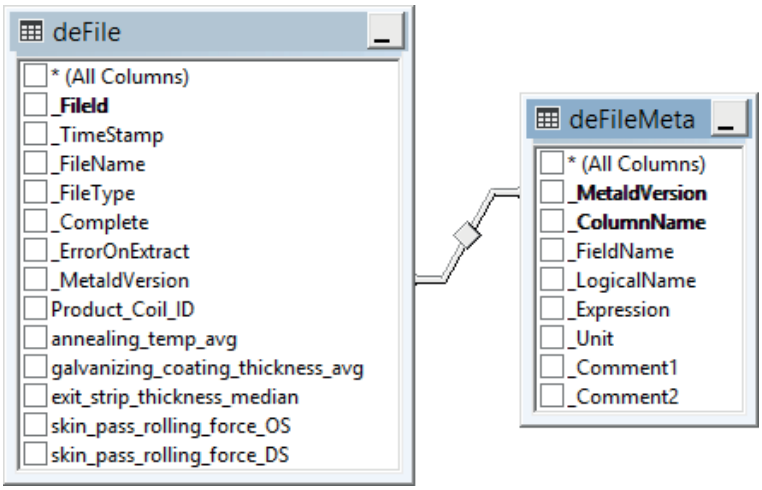


5.2 Database table "deFileMeta"

The file metadata table contains additional information (metadata) about values from computed columns in the file table. The file metadata table is created together with the default tables. It is also created or refreshed if the calculated fields are checked.

The name of the file metadata table is derived from the name of the file table and the "meta" attachment. For example, if the name of the file table is "deFile", the file metadata table is automatically called "deFileMeta".

An additional column *_MetaIdVersion* is added to the file table to refer to the metadata version for each file.



The file metadata table has the following columns:

_MetaIdVersion
The version to which the datasets in the file table refer. The current version is always the largest value of this column in the file metadata table.

_ColumnName
The column name in the file table to which this metadata set refers. This column name can be directly used in an SQL statement.

_FieldName
A readable version of the column name that you can display instead of the column name (_ColumnName).

_LogicalName
The name of the logic used to calculate the value of the column in the referenced column in the file table. The value in *_LogicalName* can be "NULL" (empty) if no logic but rather a simple expression was used for the calculation.

_Expression
The expression that was used to calculate the value in the column of the referring file table.

_Unit
The unit that was added to the logical expression used to calculate the value of the relevant column in the file table.

_Comment1 and _Comment2
Comments that have been added to the logic.

5.3 "deChannel" database table

The "deChannel" channel header contains the following information on the extracted channels. One row is assigned to each channel.

_FileID
Unique internal ID assigned to each dataset

_ChannelNr
Unique internal channel ID assigned to the extracted signal

_Name
Name of the extracted signal

_Unit
Physical unit of the extracted signal

_IsDigital
This flag is set when the signal is "logical".

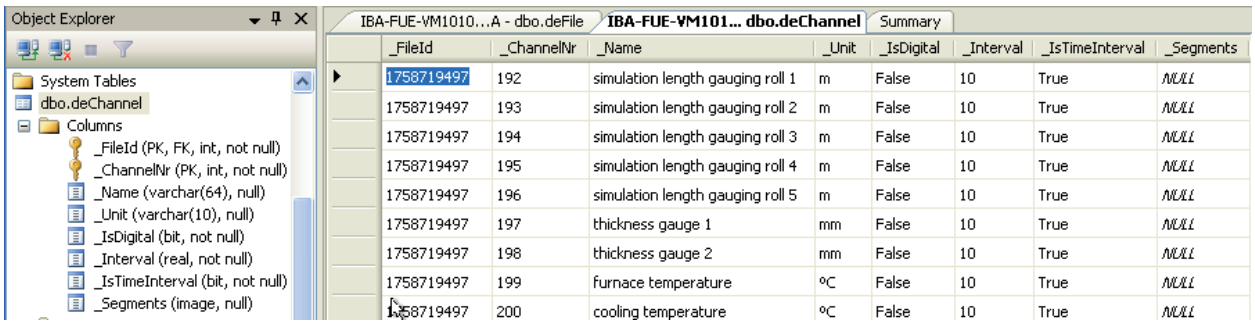
_Interval
Storage interval of the extracted data, see ↗ *Archive profile assignment*, page 42

_IsTimeInterval
This flag is set for timebased datasets.

_Segments
This column contains compressed data, which is extracted as BLOBs (binary large objects) if the corresponding option is selected for extraction, see ↗ *Using BLOBs, page 61*.

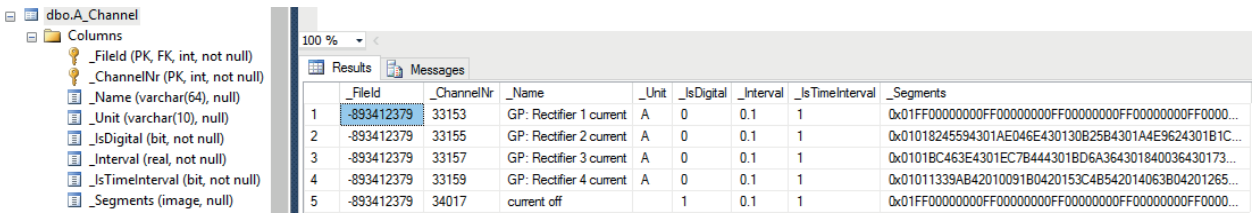
The "deChannel" is displayed in the database as shown in the following figures.

Without BLOBs:



_FileId	_ChannelNr	_Name	_Unit	_IsDigital	_Interval	_IsTimeInterval	_Segments
1758719497	192	simulation length gauging roll 1	m	False	10	True	NULL
1758719497	193	simulation length gauging roll 2	m	False	10	True	NULL
1758719497	194	simulation length gauging roll 3	m	False	10	True	NULL
1758719497	195	simulation length gauging roll 4	m	False	10	True	NULL
1758719497	196	simulation length gauging roll 5	m	False	10	True	NULL
1758719497	197	thickness gauge 1	mm	False	10	True	NULL
1758719497	198	thickness gauge 2	mm	False	10	True	NULL
1758719497	199	furnace temperature	°C	False	10	True	NULL
1758719497	200	cooling temperature	°C	False	10	True	NULL

With BLOBs:



	_FileId	_ChannelNr	_Name	_Unit	_IsDigital	_Interval	_IsTimeInterval	_Segments
1	-893412379	33153	GP: Rectifier 1 current	A	0	0.1	1	0x01FF00000000FF00000000FF00000000FF00000000FF0000...
2	-893412379	33155	GP: Rectifier 2 current	A	0	0.1	1	0x01018245594301AE046E430130B25B4301A4E9624301B1C...
3	-893412379	33157	GP: Rectifier 3 current	A	0	0.1	1	0x0101BC463E4301EC7B444301BD6A364301840036430173...
4	-893412379	33159	GP: Rectifier 4 current	A	0	0.1	1	0x01011339AB42010091B0420153C4B542014063B04201265...
5	-893412379	34017	current off		1	0.1	1	0x01FF00000000FF00000000FF00000000FF00000000FF0000...

For vector signals, the *_LogicalID* column is created additionally.

5.3.1 Using BLOBs

Instead of saving the measurement data in readable number columns in segment tables, it is also possible to keep the binary format when extracting DAT files into the database. Depending on the signal characteristics, lossless compression (only saving changes) saves a large amount of disk space and thus enables database processing of high resolution measurement data. To access this compressed data with third party applications, a decompression function is required, which has to be individually implemented.

Channel tables have a *_IsBlob* column, which states whether the corresponding *_Segments* column contains binary data. This column is created as soon as at least one profile is extracted as a BLOB regardless of the database provider used.

If you use *ibaAnalyzer* as an analysis tool, implementing BLOBs can be a very efficient storage solution. You can enable this option in the *Archive profile assignment* tab, see ↗ *Archive profile assignment*, page 42.

5.4 "deSegment" database table

5.4.1 Segment tables in standard format

The segment table "deSegment" contains the values of all signals saved in the database. There is only one table for all values in the standard format.

Note



This table is only filled if the *Blob* option is disabled.

_FileID
This is a unique internal ID assigned to each dataset.

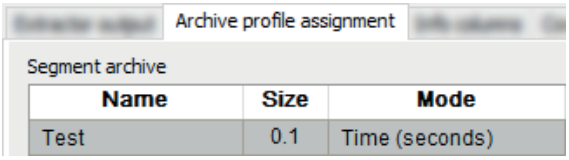
_ChannelNr
This is a unique internal channel ID assigned to the extracted signal.

_SegmentNr
Each compressed sample is assigned a segment number. The segment number is automatically incremented for each channel, starting from 0 until the end of the dataset.

_ValuesInSegment
The number of values in the raw DAT file that are contained in each stored segment (storage interval) in the database is displayed here.

Example: _ValuesInSegment
Raw data in the DAT file: 1 value every 10 ms.

Data extract to database: 1 value every 100 ms (= storage interval). Here, *Size* is set to 0.1 based on a mode of *Time (seconds)* which is set in the *Archive profile assignment* tab.



Values in every segment: 100 ms / 10 ms = 10.

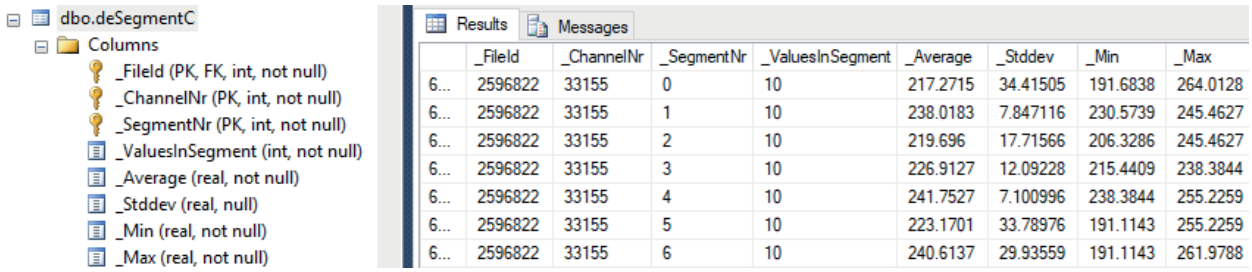
_Average
Average value of signal values within each segment

_Stddev
Standard deviation of signal values within each segment

_Min
Minimum value of signal values within each segment

_Max
Maximum value of signal values within each segment

The "deSegment" table is displayed in the database as shown in the following figure.



	_FileId	_ChannelNr	_SegmentNr	_ValuesInSegment	_Average	_Stddev	_Min	_Max
6...	2596822	33155	0	10	217.2715	34.41505	191.6838	264.0128
6...	2596822	33155	1	10	238.0183	7.847116	230.5739	245.4627
6...	2596822	33155	2	10	219.696	17.71566	206.3286	245.4627
6...	2596822	33155	3	10	226.9127	12.09228	215.4409	238.3844
6...	2596822	33155	4	10	241.7527	7.100996	238.3844	255.2259
6...	2596822	33155	5	10	223.1701	33.78976	191.1143	255.2259
6...	2596822	33155	6	10	240.6137	29.93559	191.1143	261.9788

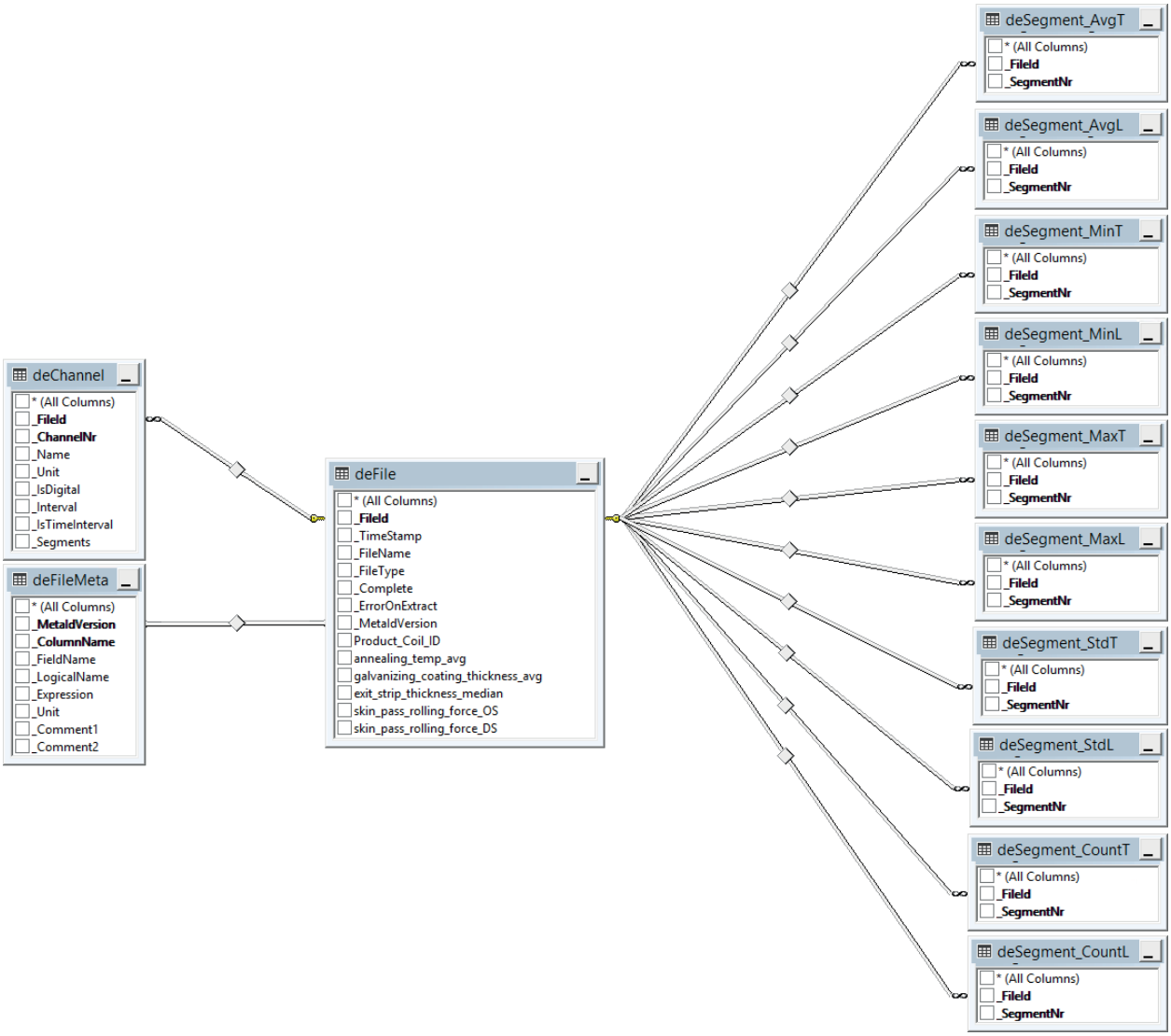
5.4.2 Segment tables in multi-column format

The multi-column format (MC) provides 5 identically structured segment tables for timebased (T) and lengthbased (L) extracted data.

- Average (Table deSegment_AverageT/L)
- Minimum (Table deSegment_MinT/L)
- Maximum (Table deSegment_MaxT/L)
- Standard deviation (Table deSegment_StdT/L)
- Values in segment (table deSegment_CountT/L)

Which of these tables are actually created depends on the settings in the archive profiles: The table for the average is always created. The tables for minimum, maximum, standard deviation, and values in a segment are created as soon as at least one profile has enabled the respective calculation.

In contrast to the standard format, each channel has its own column in the respective segment table.



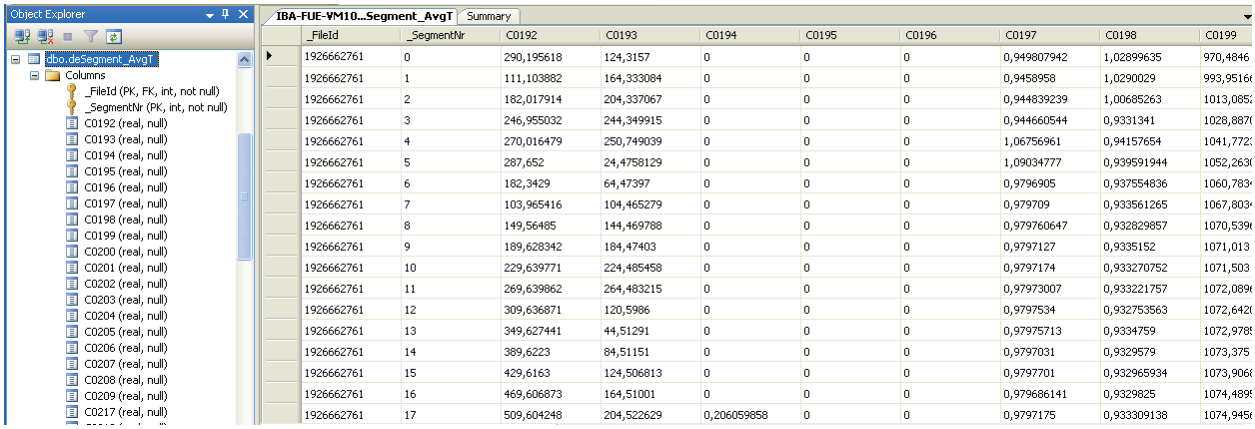
Depending on the option *Start a new segment table every x columns* additional segment tables may be created automatically, see ↗ *Multi-column schema mode*, page 35. You can use this option to bypass restrictions on the number of columns in database tables.

Before inserting the segment records, *ibaAnalyzer* checks if columns for all the channels exist. If not, the segment tables are modified. By default, the column names for the channels are derived from the channel number.

Note

These tables are only filled if the *Blob* option is disabled.

The table "deSegment_AvgT" is displayed in a multi-column database in MSSQL-format as shown in the following figure.



_FileId	_SegmentNr	C0192	C0193	C0194	C0195	C0196	C0197	C0198	C0199
1926662761	0	290,195618	124,3157	0	0	0	0,949807942	1,02899635	970,4846
1926662761	1	111,103882	164,333084	0	0	0	0,9458958	1,0290029	993,9516
1926662761	2	182,017914	204,337067	0	0	0	0,944839239	1,00685263	1013,085
1926662761	3	246,955032	244,349915	0	0	0	0,944660544	0,9331341	1028,887
1926662761	4	270,016479	250,749039	0	0	0	1,06756961	0,94157654	1041,772
1926662761	5	287,652	24,4758129	0	0	0	1,09034777	0,939591944	1052,263
1926662761	6	182,3429	64,47397	0	0	0	0,9796905	0,937554836	1060,783
1926662761	7	103,965416	104,465279	0	0	0	0,979709	0,933561265	1067,803
1926662761	8	149,56485	144,469788	0	0	0	0,979760647	0,932829857	1070,539
1926662761	9	189,628342	184,47403	0	0	0	0,9797127	0,9335152	1071,013
1926662761	10	229,639771	224,485458	0	0	0	0,9797174	0,933270752	1071,503
1926662761	11	269,639862	264,483215	0	0	0	0,97973007	0,933221757	1072,089
1926662761	12	309,636871	120,5986	0	0	0	0,9797534	0,932753563	1072,642
1926662761	13	349,627441	44,51291	0	0	0	0,97975713	0,9334759	1072,978
1926662761	14	389,6223	84,51151	0	0	0	0,9797031	0,9329579	1073,375
1926662761	15	429,6163	124,506813	0	0	0	0,9797701	0,932965934	1073,906
1926662761	16	469,606873	164,51001	0	0	0	0,979686141	0,9329825	1074,489
1926662761	17	509,604248	204,522629	0,206059858	0	0	0,9797175	0,933309138	1074,945

5.5 Differences between the database providers

There are some differences between the databases supported by *ibaAnalyzer*. For example, the column names of *ibaAnalyzer* for Oracle and MSSQL are not identical.

The restrictions are particularly severe for object and column naming in large database providers based on mainframe, VMS or UNIX platforms and architectures (DB2-UDB, Oracle). Thus, some functions of *ibaAnalyzer* cannot work with all database providers e.g. BLOBs or cascading constraints, see ↗ *Customization and integration, page 65*.

5.6 Customization and integration

ibaAnalyzer provides a basic framework for database integration, i.e. it creates default tables, constraints and indices with a mouse click. This may be sufficient in small standalone environments, however even in environments like this, some implementations are recommended, such as data deletion or archiving, to ensure availability.

If you load high-resolution data into a database, archiving functions and, in particular, deletion functions are required. You must derive these deletion functions from the individual data management strategy.

In MySQL, DB2-UDB, MSSQL, PostgreSQL and Oracle databases, the cascading constraints ensure that when a row in the file table is deleted, the corresponding rows in the segment tables and channel headers are also deleted (referential integrity).

Example of a simple deletion strategy
Every night at 02:00, an automated job is executed that deletes all data older than 100 days.

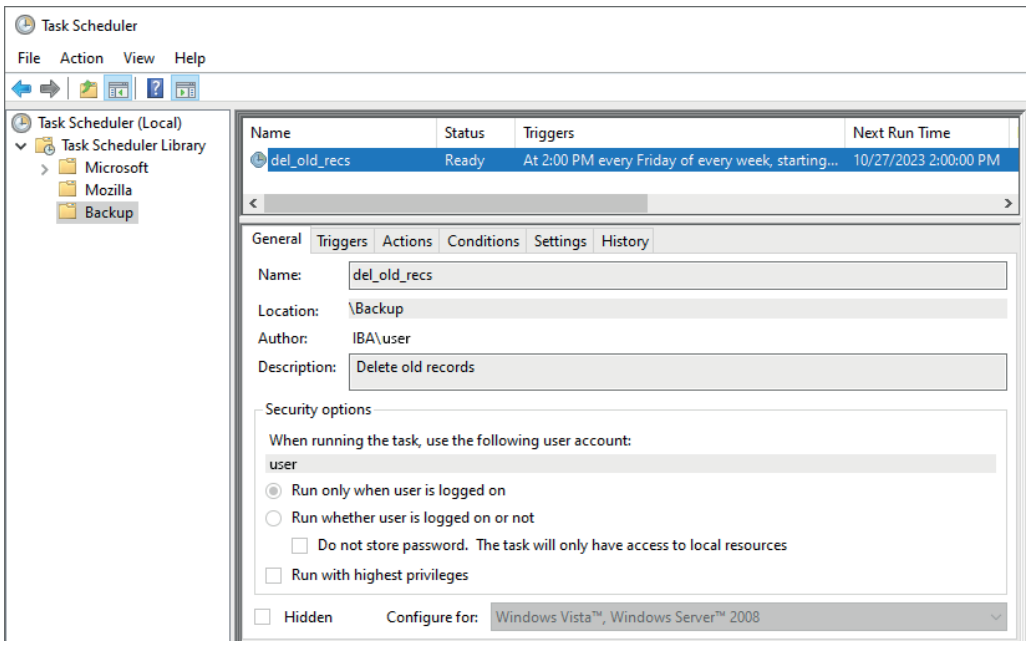
For this purpose, a simple SQL command (MSSQL syntax) is processed every night, which can be stored in a short SQL script (`del_old_recs.sql`):

```
1 USE IBADB
2 GO
3 delete deFile where _TimeStamp < (GetDate() - 100);
4 GO
5 COMMIT
6 GO
```

A batch file (`del_old_recs.bat`) can be created in a Windows environment.

```
@echo off
SQLCMD -S dbhost\SQLEXPRESS -i C:\Backup\del_old_recs.sql
```

In the control panel, add a scheduled task "del_old_recs".



Note



Deleting data sets does not always lead to an increase in free disk space. Depending on the database provider, it may be necessary to execute additional maintenance commands or maintenance procedures.

Data management strategies may also depend on third-party applications. For example, an MES application or QM application may provide the trigger for deleting or archiving entries that come from product-related files. In this case, a product ID reference is required in the file table.

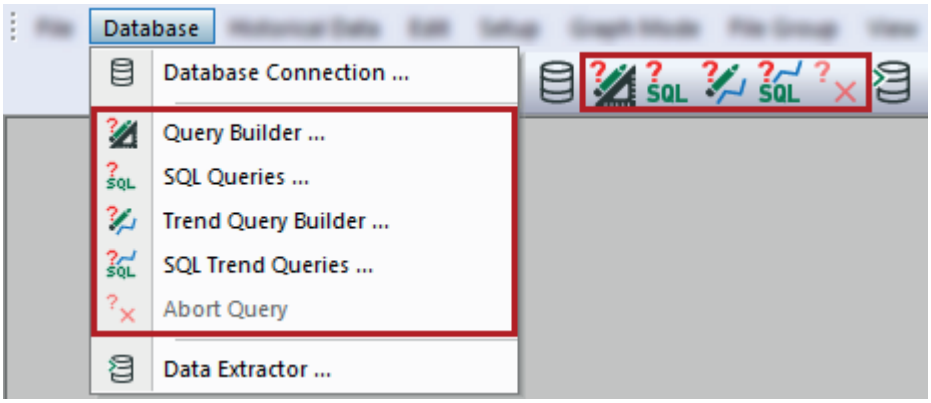
Another part of the data management strategy concerns disk space requirements. How much disk space is required for the implementation of *ibaAnalyzer* (e.g. for 100 days of storage)? The various set lengths, index space requirements, etc. can be calculated based on the information provided by the database provider. In practice, it is better to simulate typical data processing and monitor the space requirements. A typical database extraction can be easily configured with a small amount of test data.

If you want to use the extracted measurement data together with other data in a production environment or information system environment, you need additional indices for performance optimization or info columns filled by external procedures. All INSERT statements of *ibaAnalyzer* are "fully qualified". Therefore, adding NULL columns is not a problem. Adding indices is also possible and recommended. If database triggers are necessary, these may slow down load performance considerably, especially in the case of long segment tables.

A very efficient way to connect *ibaAnalyzer* tables with third-party tables is to define database views that hide the complexity of JOIN conditions.

6 Analysis of data from databases

Not only does *ibaAnalyzer* carry out the ETL (Extract Transform Load) process, it also provides powerful functions to query data from databases for default analysis or trend analyses in *ibaAnalyzer*. The queries can be started via the *Database* menu or directly via the toolbar.



Note

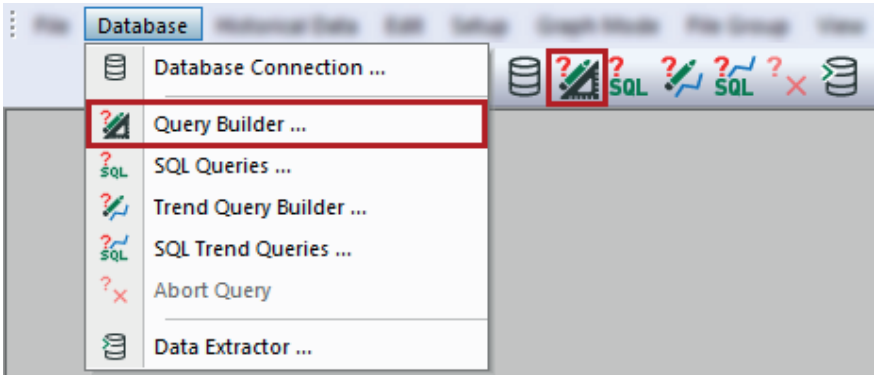
To read or query databases, the *ibaAnalyzer-DB-Read* license is needed.

6.1 Default queries

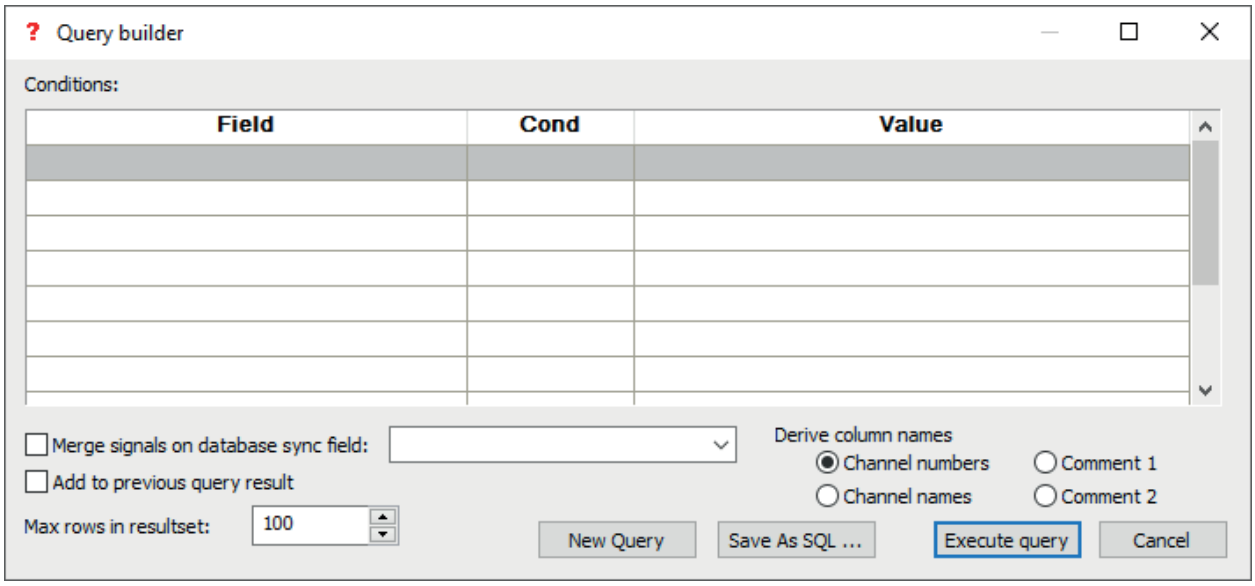
6.1.1 Query builder

You can use the query builder to create standard queries without any knowledge of SQL (structured query language).

You can open the query builder either via the *Database – Query Builder* menu or via the corresponding button in the toolbar.

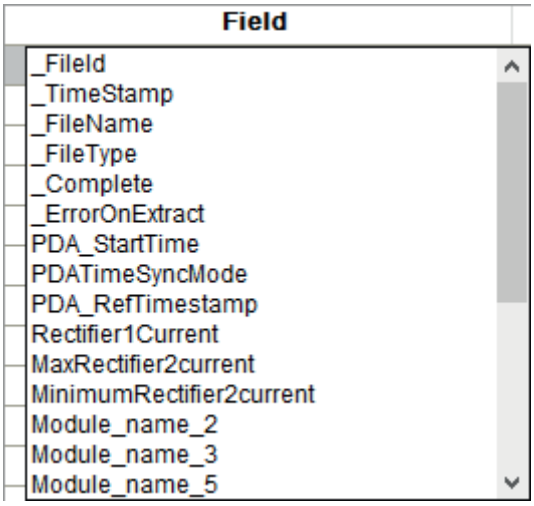


In the following window, you can configure the default queries.



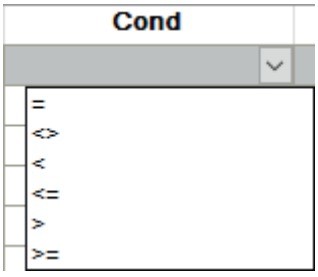
Conditions

You can define the conditions in the query builder based on the content of the main fields in a file table. When you open the drop-down list in the *Field* column, you can select the available fields.



These fields are the same as in the database file table. In addition to the default fields such as *_FileID*, *_TimeStamp*, *_FileName* or *_FileType*, you can also select custom fields from the info column, e.g. "STRIP_ID".

When you open the drop-down list in the *Cond* column, you can select a condition.

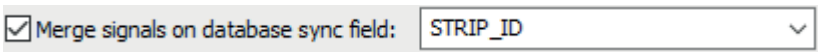


In the *Value* column, enter the value to be compared, which must be compatible with the database format. You can enter several rows of queries, which are then automatically combined with AND keywords.

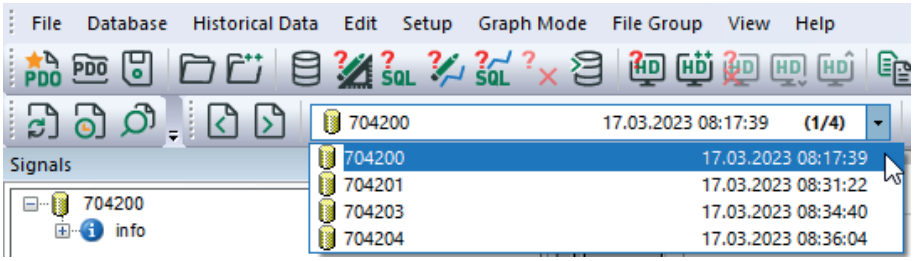
Field	Cond	Value
_TimeStampMicroSecs	>	100000
_TimeStampMicroSecs	<	140000

If the database field type of the selected field is *Date* or *DateTime*, a calendar is automatically shown when you open the drop-down list.

Merge signals on database sync field
If data from different DAT files (e.g. from different *ibaPDA* systems, *ibaLogic* systems or *ibaFiles* applications) are linked by a common identifier, such as the material ID, and these are to be merged, you can define a synchronization field, see ↗ *Segment tables in multi-column format*, page 63.



If you enable this option and select a synchronization field, all queried datasets (files) that belong to this field are merged. After the query, the query results are shown in the drop-down list of the DAT file group (toolbar) sorted by the values of the synchronization field.



Add to previous query result
If you enable this option, the result of this query is added to previous query results. The option corresponds to the *Add data file function* when working with DAT files. If you do not enable this option, all previous query results are deleted.

Max. rows in resultset x
To prevent unmonitored use of system resources (database server, network, etc.), enter a suitable value here. For example, using a value of "10" selects data from 10 DAT files. The value "0" disables this restriction.

Derive column names from
See ↗ *Multi-column schema mode*, page 35.

<New query>
Use this button to delete all entries in the query builder. Then you can define a new query.

<Save as SQL>
You can save the SQL command from the query builder as a text file. This function is useful to obtain a basic statement that you can customize or extend. In the *SQL queries* window, you can then execute this statement.

<Execute query>
Use this button to execute the query via the configured database connection. If successful, a message appears with the number of datasets queried.

Note

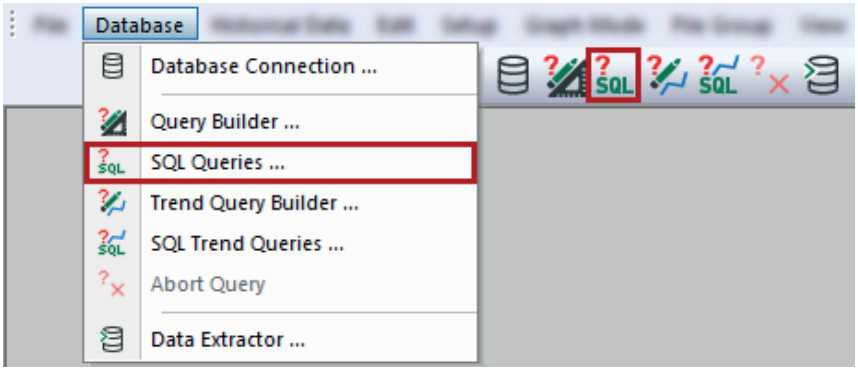
If the *Max. rows in resultset* parameter has been exceeded, two messages are displayed. Confirm these messages and reconfigure the query.

<Cancel>
The window closes without saving the query and any running query is canceled.

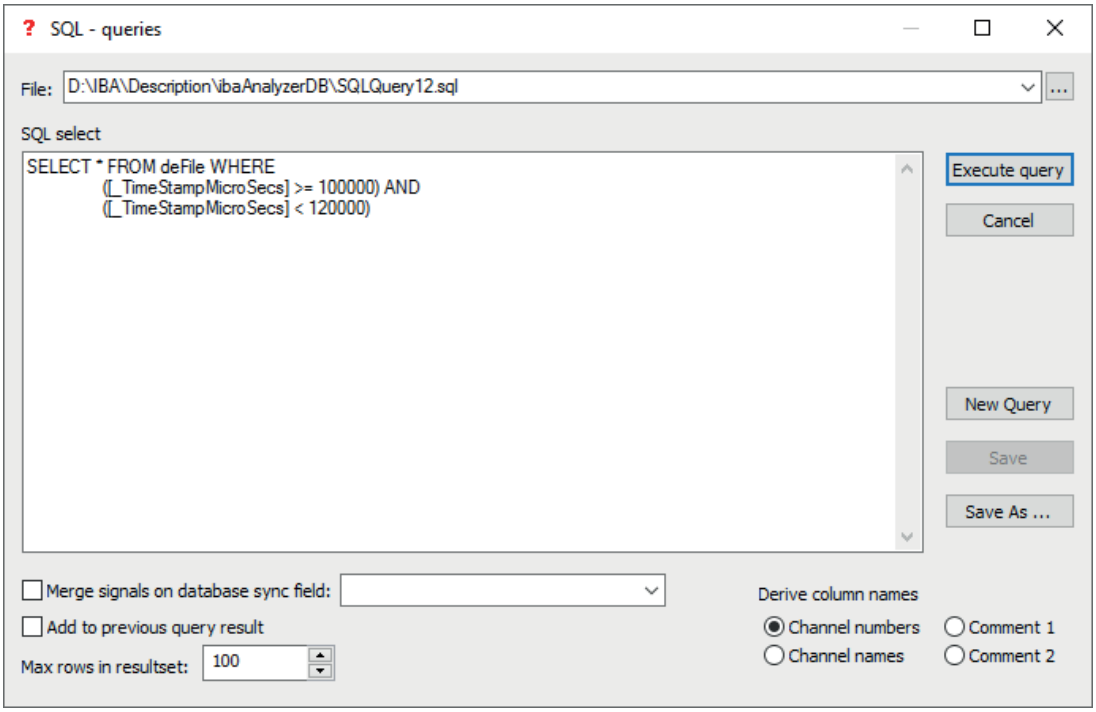
6.1.2 SQL queries

Queries can also be directly written in SQL syntax. This option offers full flexibility in the database or application environment used. In the WHERE condition, you can use any table that is accessible via the configured database connection.

You can open the SQL query either via the *Database – SQL queries* menu or via the corresponding button in the toolbar.



In the following window, you can configure SQL queries.



File
If available, select a previously created query file.

SQL select
Enter an SQL text. If an SQL file is open, the text is displayed here and can be edited.

<Save>
Save changes to the currently open query file.

<Save as>
Save as a new query file.

For more information on the following settings, see ↗ *Query builder, page 67*:

- Merge signals on database sync field
- Add to previous query result
- Maximum rows in the result set
- Derive column names
- <Execute query>
- <Cancel>
- <New query>

Note



The SQL query can only use the column names specified by the selected option *Derive column names*.

Note



The SQL syntax may differ depending on the database provider used.

Example: Syntax differences
Query of data from the last two days in SQL Server:

```
SELECT * FROM PDA_File WHERE _Timestamp > (getdate()- 2)
```

Query of data from the last two days in Oracle syntax:

```
SELECT * FROM PDA_File WHERE I_Timestamp > (SYSDATE - 2)
```

6.1.3 Handling queried datasets

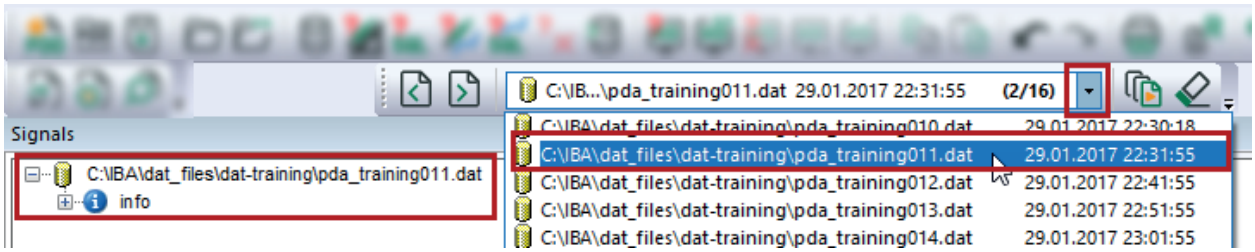
Once the datasets have been queried from the database, there are various options for opening and analyzing them in *ibaAnalyzer*.

The query results are only temporarily available after the query. If you have closed *ibaAnalyzer* since then, you need to run the query again.

Opening a single query result

From the queried datasets, a dataset can be selected and opened from the *File group* drop-down list.

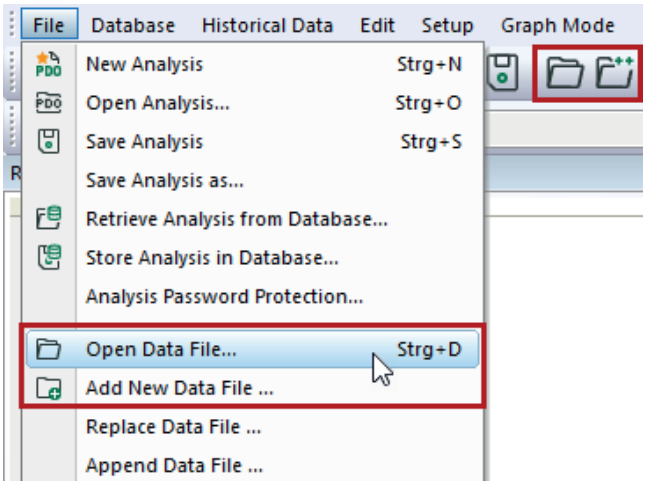
→ The selected result is displayed in the signal tree.



Opening multiple query results

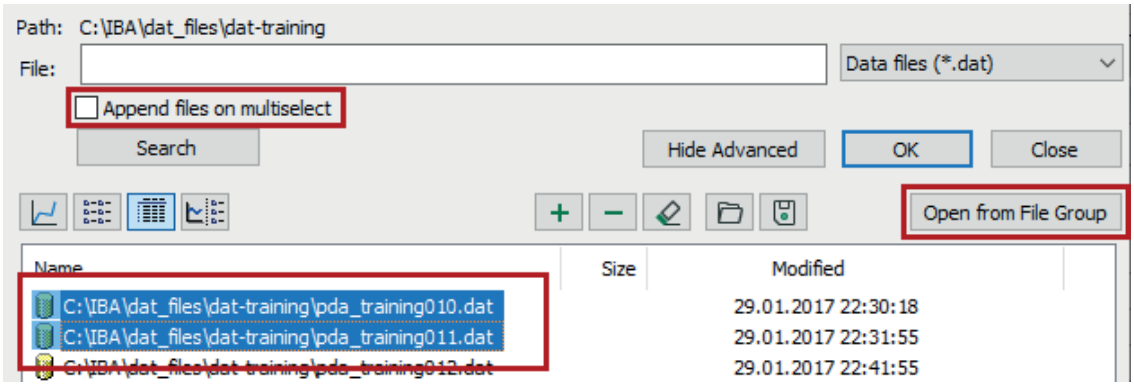
You can also open the queried datasets like files via the Advanced area in the *Open data file* or *Add data file* dialog.

1. Open the dialog to open or add a DAT file either via the *File – Open data file* or *Add data file* menu or via the corresponding button in the toolbar.



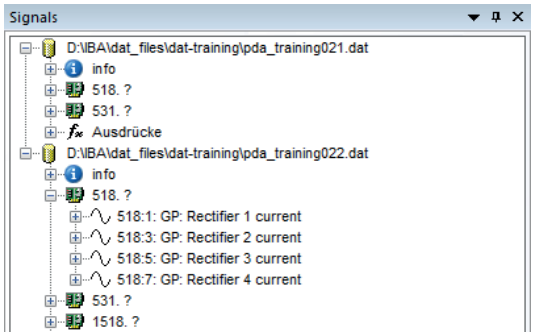
2. Select the desired query results from the Advanced area.
   If you do not see the Advanced area, click <Show Advanced>.

3. You can also use the *Append files on multiselect* option for queried data.
   If required, enable the *Append files on multiselect* option.



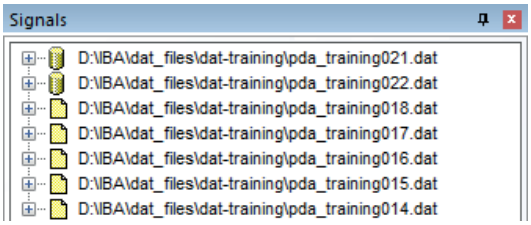
4. Click on <Open from file group>.

→ The selected database file then appears in the signal tree.

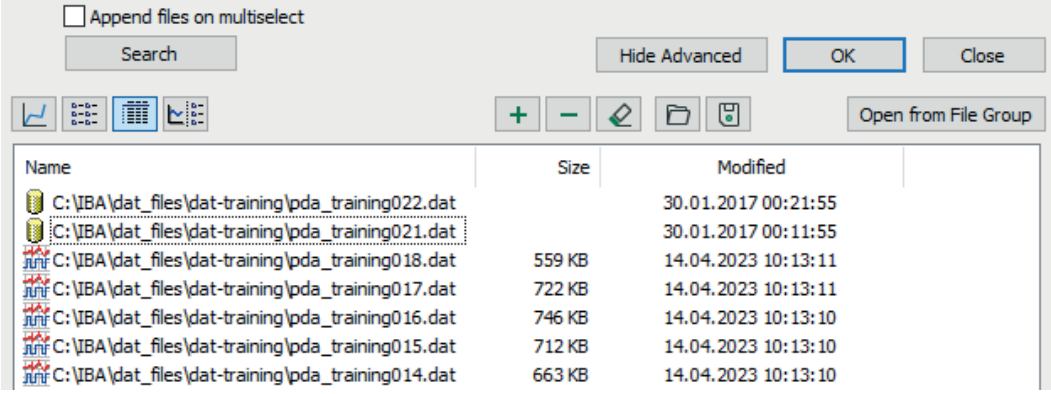


Open query results and DAT files at the same time

The results of database queries and DAT files can also be simultaneously displayed in the signal tree. The different data sources can be distinguished by the respective symbols.

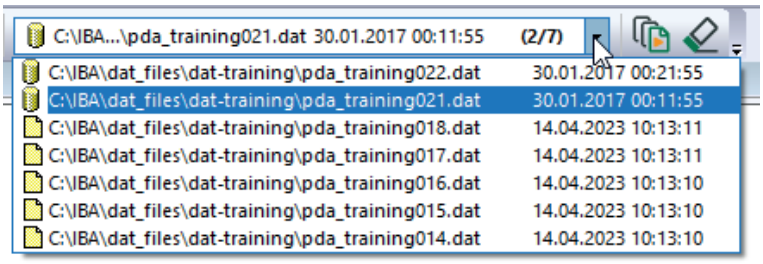


1. Open the dialog to open or add a DAT file either via the *File – Open data file* or *Add data file* menu or via the corresponding button in the toolbar.
2. Add the desired DAT files to the Advanced area of the dialog. If you do not see the Advanced area, click <Show Advanced>.

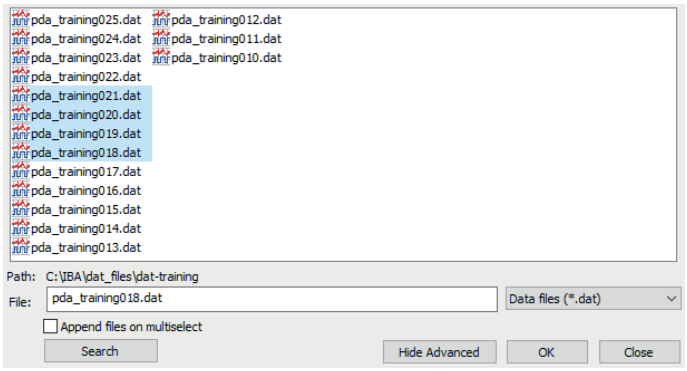


3. Select the desired query results and DAT files from the Advanced area.
4. Click on <Open from file group>.

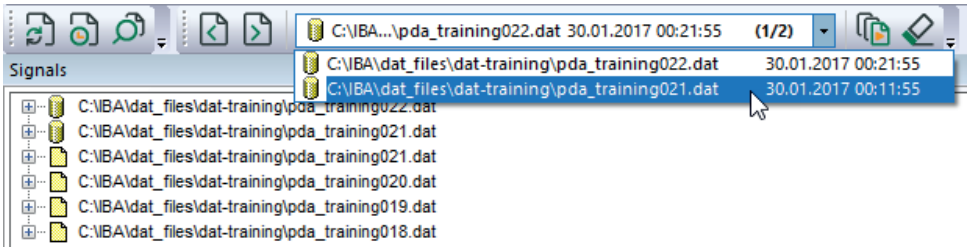
→ The DAT files and query results appear in the *File group* drop-down list and in the signal tree.



You can also open the DAT files in the *Open data files* dialog via a selection and with <OK> instead of via the Advanced area with the <Open from file group> button.

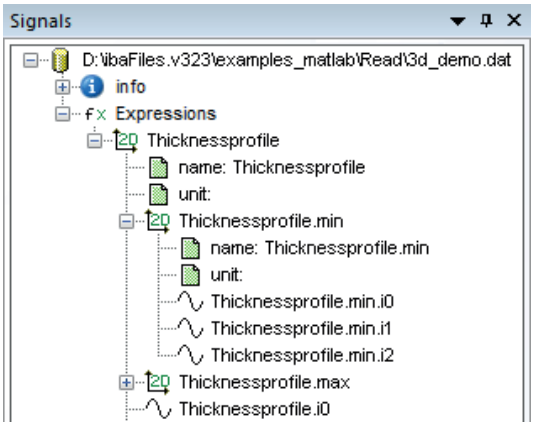


→ The DAT files then appear in the signal tree but not in the *File group* drop-down list.



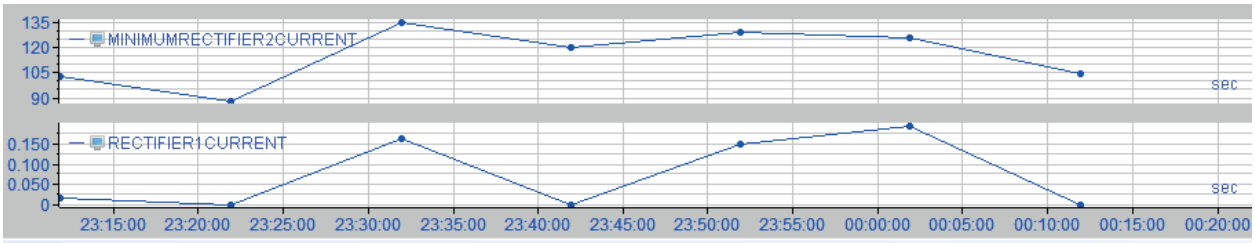
Multidimensional channel expressions

From the *Expressions* group, you can select the queried multidimensional channel expressions. You can display and evaluate the signals as usual. You can also select and use the sub-signals for calculations or expressions.



6.2 Trend queries

Instead of querying detailed measurement data by default queries, you can also query the info fields and computed columns which have an aggregation level based on DAT files (i.e. one value per DAT file). This resolution permits long-term analysis over days, weeks, months or years without generating excessive data volumes. The file table created by *ibaAnalyzer* is used to display a trend graph of this data. Theoretically, you can also query data from other database tables or views, provided that these tables contain a timestamp column and numerical data or string data. The results of a trend query are signals containing non-equidistant data points, similar to the XY view.

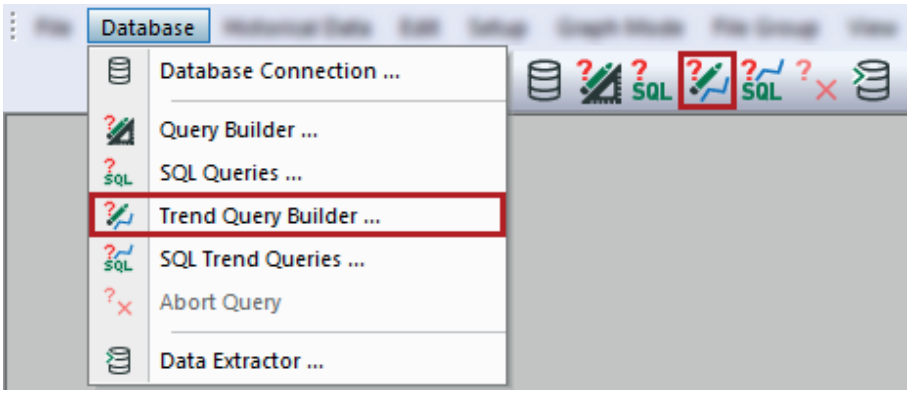


Each dataset in the query result is displayed by a point which has an x-coordinate that corresponds to the time stamp of the source file and a y-coordinate that corresponds to the aggregated value of the info field. You can use these generated signals in further calculations. However, in this case they are first converted to equidistant signals by linear interpolation before being used in the given expression. The sample interval of the newly sampled signal is the time interval between the two closest sample points in the original, non-equidistant signal.

6.2.1 Trend query builder

Similarly to the default query builder, the trend query builder can assist in the creation of SQL statements to perform a trend query.

You can open the trend query builder either via the *Database – Trend Query builder* menu or via the corresponding button in the tool bar.



In the following window, you can configure trend queries.

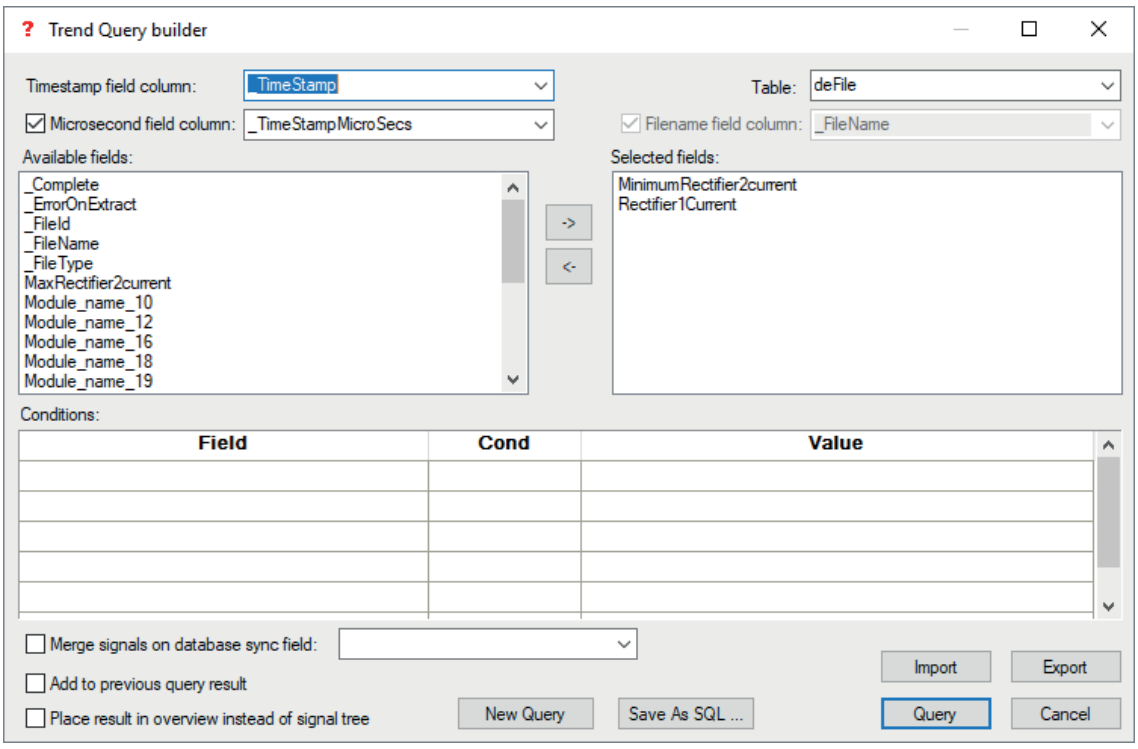
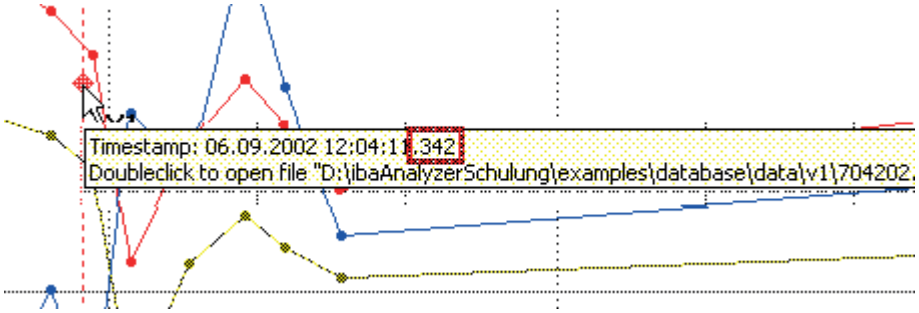


Table
Select the table to be queried.

Timestamp field column
Trend queries require a time stamp. Select the appropriate column here.

Microsecond field column
Enable this option if microsecond precision is required, see ↗ *Database table "deFile"*, page 57. Specify the column in which microseconds are available. Hovering the cursor over the diamond in the *Overview* window also displays the microseconds.



Filename field column
This column contains the filenames. This option is only available if the *Place result in overview instead of signal tree* option is enabled.

_FileName
D:\IBA\dat_files\dat-training\pda_training018.dat
D:\IBA\dat_files\dat-training\pda_training016.dat
D:\IBA\dat_files\dat-training\pda_training019.dat
D:\IBA\dat_files\dat-training\pda_training017.dat
D:\IBA\dat_files\dat-training\pda_training015.dat

If you enable this option, you can open the original files using the markers in the *Overview* window that correspond to the extracted data in the database, see ↗ *Markers, page 83*. If the table to be queried has the iba database layout, the correct column is preselected.

You can disable this option if the original files are not required or no longer present, as the query speed is then increased.

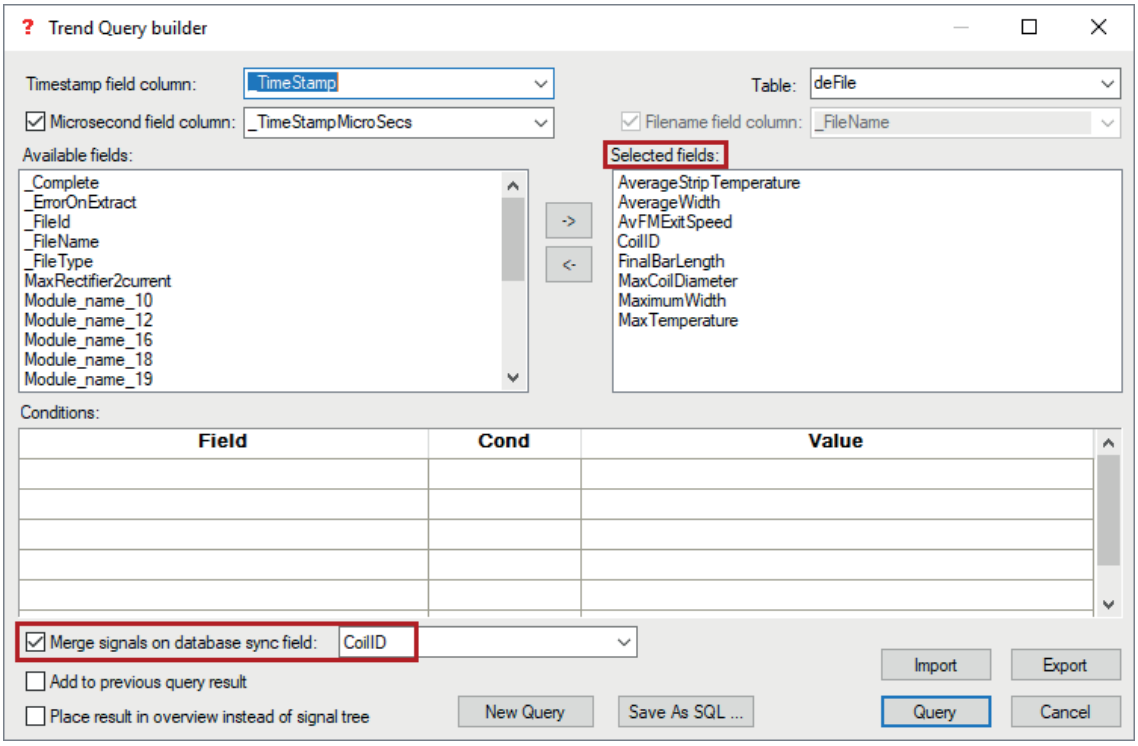
Available fields
All the fields which are available in the selected table (type "deFile") are displayed.

Selected fields
Using the arrow buttons, you can select the fields required for the query.

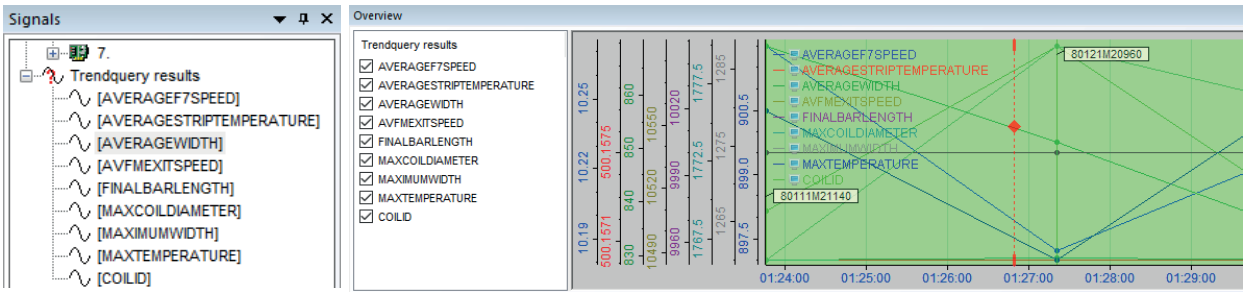
Table conditions
Configure the conditions in the same way as for default queries, see ↗ *Query builder, page 67*.

Connect signals to database synchronization field

This function is similar to the function for default queries, see ↗ *Query builder, page 67*.

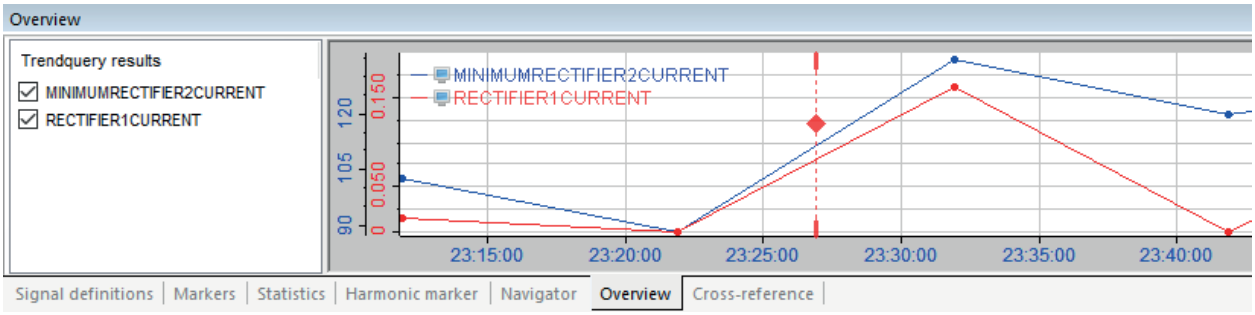


If you enable this option and select a synchronization field in the drop-down list (e.g. "CoilID"), all rows with the same value in the synchronization field column are merged and displayed as a single data point in the trend query. The selected fields appear under *Trend query results* in the signal tree or in the *Overview* window.



The trend query results display shows the values for the selected measured variables that have the oldest time stamp and are not "0". If the DB values are "0" or "NULL" (empty), "0" is displayed.

Place result in overview instead of signal tree
If you enable this option, the trend query results appear in the *Overview* window instead of the signal tree.



Note

Note that the *Add to previous query result* option is grayed out when this option is enabled. This is because there can only be one query result (possibly from several signals) present at a time in the overview. Any trend query result already displayed in the overview will be replaced by the next query.

<Execute query>
Each field in the set of results generates a signal with a name that is identical to the field name. The signal appears in the signal tree under the *Trend query results* node.

<Import>
You can re-import previously exported trend query builder settings. The current settings are overwritten.

<Export>
Trend query builder settings are exported as an INI file.

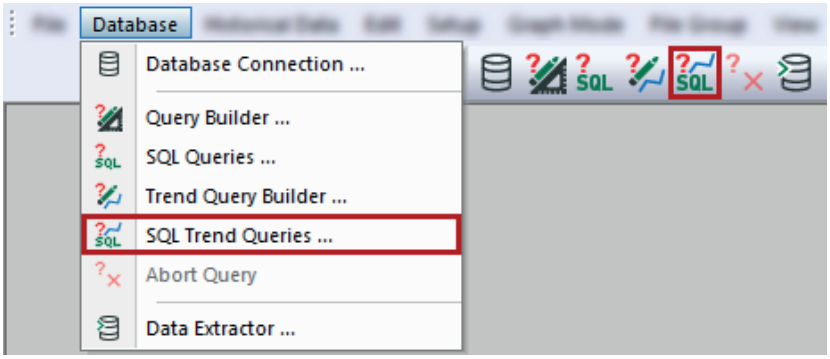
For more information on the following settings, see ↗ *Query builder, page 67*:

- Conditions
- Add to previous query result
- <New query>
- <Save as SQL>
- <Cancel>

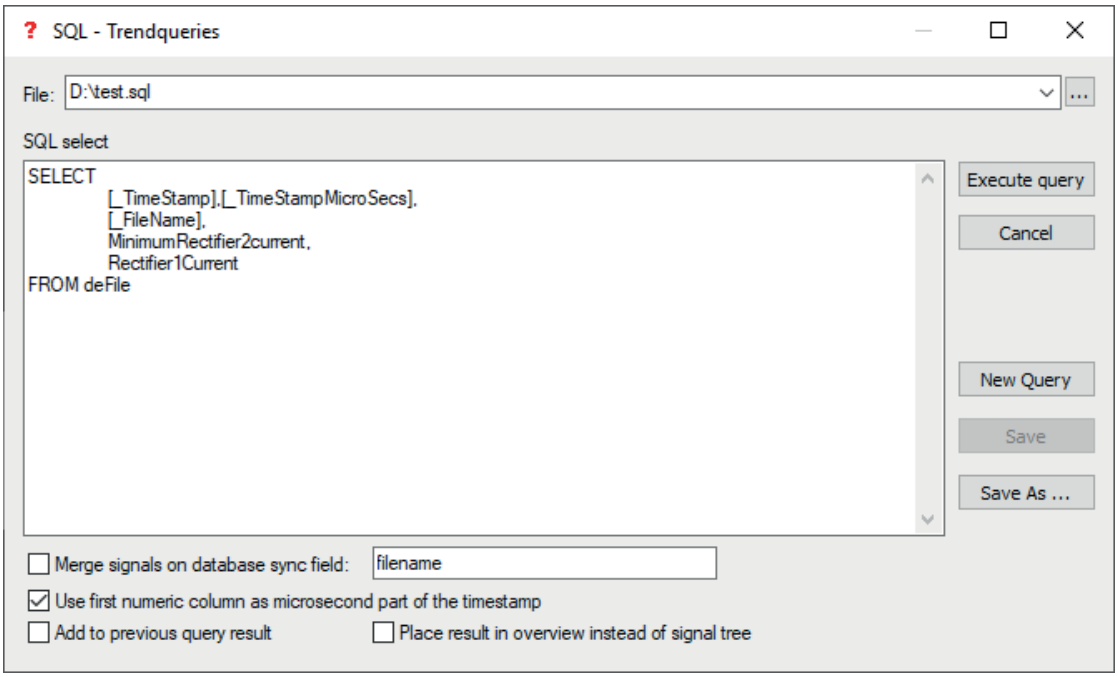
6.2.2 SQL trend queries

You can write trend queries in the same way as SQL queries, see ↗ *SQL queries, page 70*.

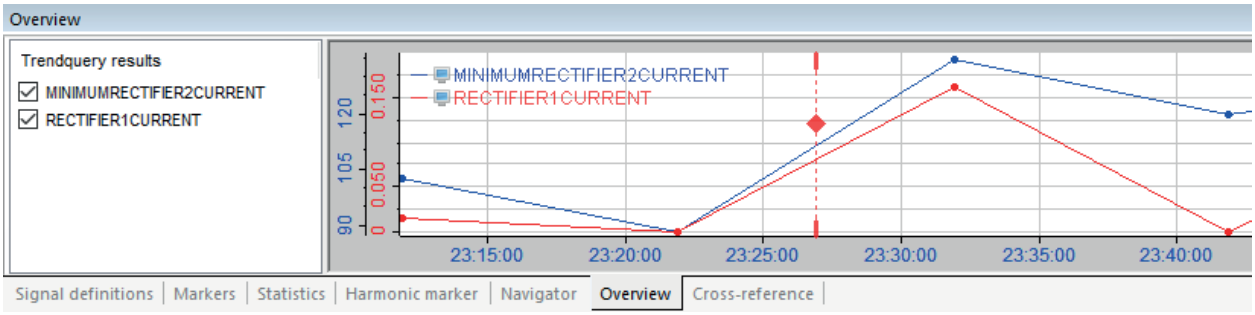
You can open the SQL trend query builder either via the *Database – SQL trend queries* menu or via the corresponding button in the toolbar.



In the following window, you can configure SQL trend queries. Enter SQL statements to query the data from the database.



Place result in overview instead of signal tree
If you enable this option, the trend query results appear in the *Overview* window instead of in the signal tree.



Note

Note that the *Add to previous query result* option is grayed out when this option is enabled. This is because there can only be one query result (possibly from several signals) present at a time in the overview. Any trend query result already displayed in the overview will be replaced by the next query.

<Execute query>
Each field in the set of results generates a signal with a name that is identical to the field name. The signal appears in the signal tree under the *Trend query results* node.

Use first numeric column as microsecond part of the time stamp

Note

This column must contain the correct data type, otherwise the result will be meaningless.

When executed, the statement must contain at least one numerical field and one set of results with a time stamp field. If there are several timestamp columns, only the first one is referenced, the others are displayed. The statement must also contain an ORDER BY clause for the time stamp field.

For more information on the following settings, see ↗ *Query builder, page 67*:

- Merge signals on database sync field
- Add to previous query result
- <Cancel>
- <New query>

For more information on the following settings, see↗ *SQL queries, page 70*:

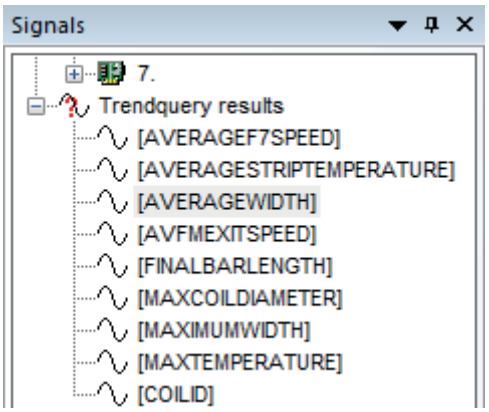
- File
- SQL select
- <Save>
- <Save as>

6.3 Trend query results

In addition to the default query results, you can also display and evaluate trend query results using functions such as overview display and drill down.

6.3.1 Signal tree of trend query

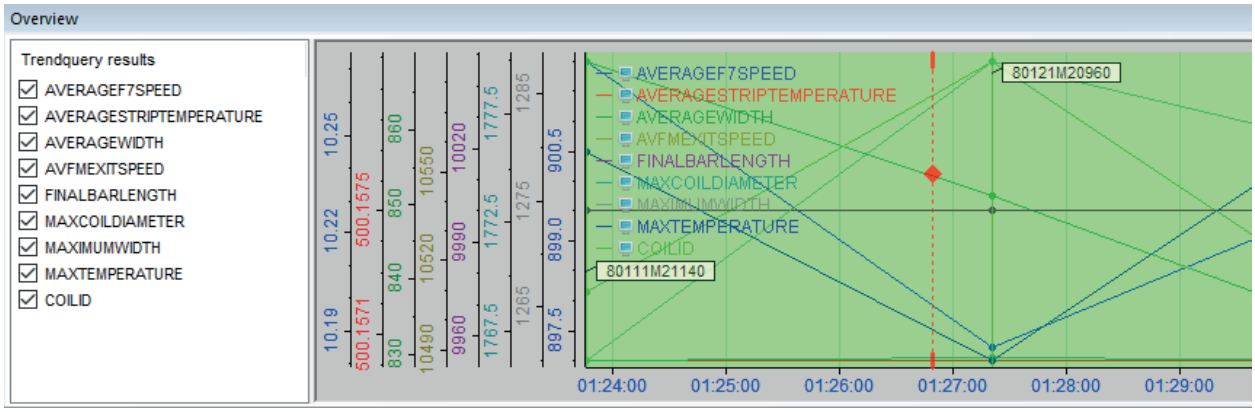
The results generated by the queries can be seen in the signal tree under the *Trend query results* node. To display the results in the recorder window, double-click on a result or drag a result into the recorder window.



To remove the query result from the signal tree, right-click on the result or on the *Trend query results* node and select *Clear trend query results* in the context menu. The *Close all data files* command in the context menu of the signal tree or in the *File* menu also deletes the query results. You can only erase a query result in the *Overview* window with a new query.

6.3.2 Trend queries in Overview window

You can display trend queries in the *Overview* window. On trend queries in this window, you cannot perform calculations. However, the *Overview* window has features that are not available for trend queries in the main window of the signal display. The Overview is particularly useful if you start a database query that searches for extracted files that correspond to the data points of the trend query results in the Overview. You can open the original DAT files that correspond to these extracted files here (drill down function).

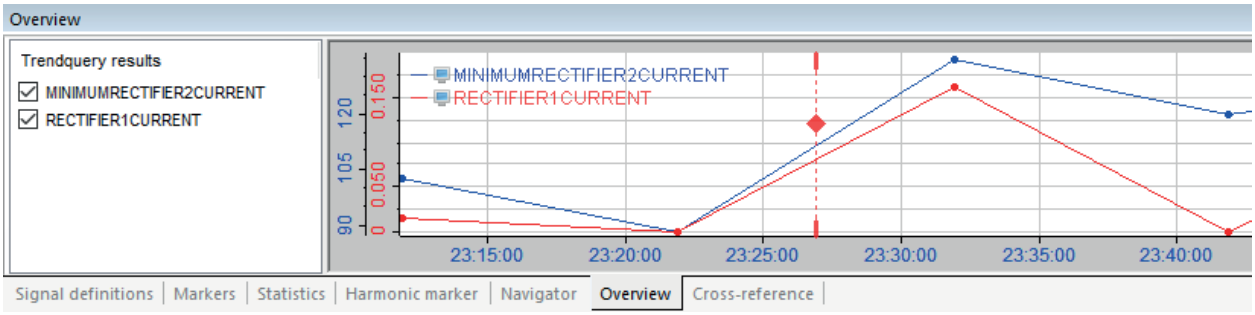


You can only erase a query result in the *Overview* window with a new query.

6.3.2.1 Results list and trend query chart

The overview consists of two parts:

- A chart on the right-hand side to display the trend query results.
- A list of signal names that are available in the trend query results on the left-hand side.



This list contains all column names resulting from the trend query. If the synchronization field is numerical, it is also available here.

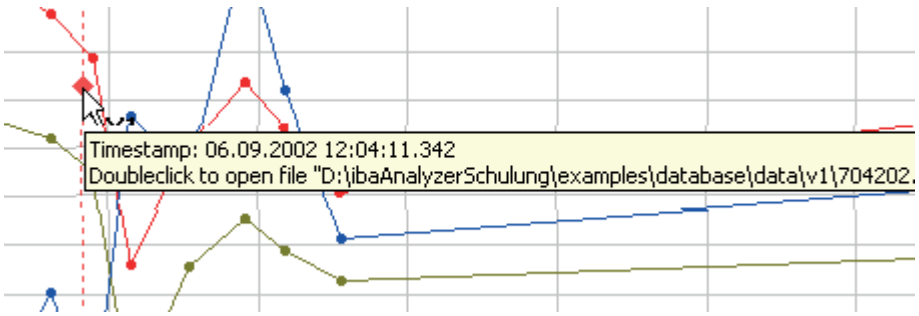
If you uncheck the checkboxes next to the column names, the signals in the chart on the right are hidden. If you select the result at the top of the list and click on *Trend query results*, the first selection field state is inherited by the remaining results. In this way, you can enable or disable multiple signals together.

The signal chart of the trend query results is similar to the signal display in the recorder window of *ibaAnalyzer*. You can drag and drop signals within the diagram onto the same Y-axis (but not out of the diagram). The zoom buttons in the *ibaAnalyzer* toolbar also work, provided that the overview is the active window (i.e. the last window clicked on).

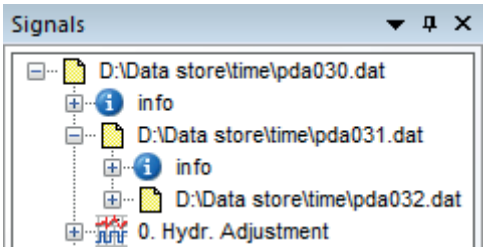
6.3.2.2 Markers

A pair of markers is available in the *Overview* window. These are like the normal markers but also have a diamond in their upper area. You can move these markers either by the ends or the diamond. If you press the <Ctrl> key while moving the markers, the markers jump to the next data point.

If you move the mouse over the diamond of a marker, a tooltip appears showing the exact timestamp of the data point and the double-click action. Depending on the settings, this either opens the original file or queries the database for the extracted file.



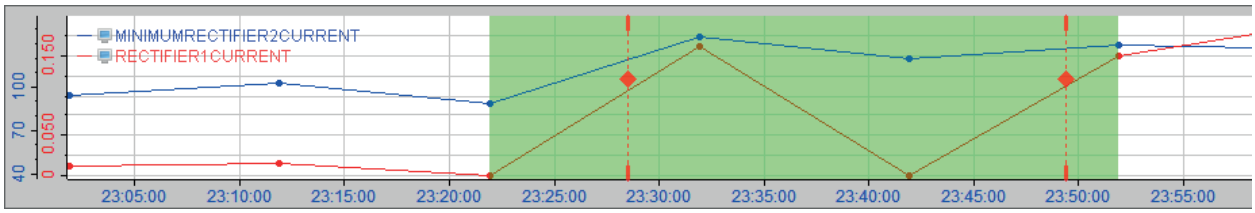
If you press the <Ctrl> key when double-clicking, all files enclosed by the markers are opened or queried and attached.



Note

Every data point is located at the start of the associated file. This means that the area to the right of a data point corresponds to the file up to the next data point or the next file, etc.

The green rectangle in the Overview window shows the time range visible in the Recorder window.



Marker context menu with drill down options

A right-click on the diamond opens a context menu with the following drill down options.

Open file at marker
Open the original file corresponding to the data point. This option is grayed out if you did not select the column with the filenames in the query dialog.

Append range of files between markers
Opens and appends all files (i. e. generates chain of files in the signal tree) enclosed by the markers. This option is grayed out if you did not select the column with the filenames in the query dialog.

Query database for file at marker
Instead of opening the original file, the extracted file in the database is queried.

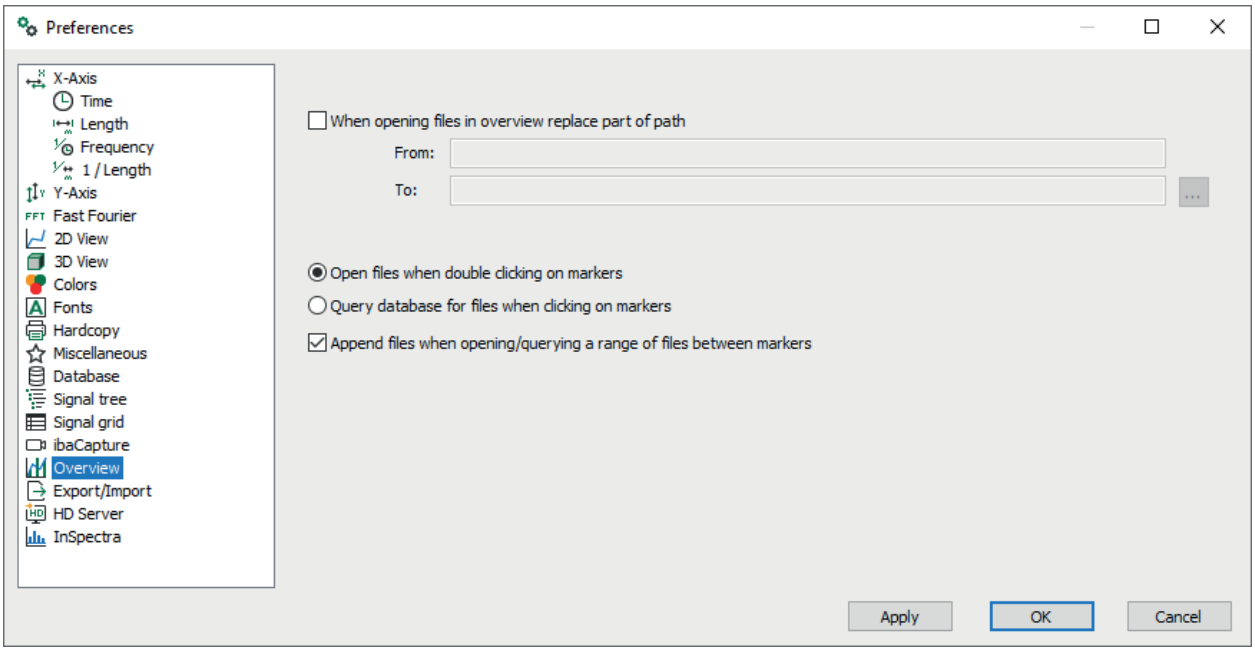
Query database for range of files between markers
Similar to *Append range of files between markers*, but queries and appends the extracted files instead.

Setup
Opens the *Preferences* dialog with the *Overview* tab, see ↗ *Overview options, page 85*. You can open this tab in the menu bar via *Setup – Preferences*.

6.3.2.3 Overview options

The following options are available in the *Preferences* dialog in the *Overview* node.

You can open this menu via the context menu of the diamond in the *Overview* window or in the menu bar via *Setup – Preferences*.



When opening files in overview replace part of path
If the original DAT files have been moved to another location, the entire path or part of the path (e.g. from the local *ibaPDA* system) can be replaced with the full address of a file server from the *FileName* column. The path can be selected with the browser button <...>.

Open files when double-clicking on markers or query database for files when double-clicking on markers
The default behavior when double-clicking on the marker diamond can be selected: either opening the original files or querying the database for the extracted files.

Append files when opening/querying a range of files between markers
If you enable this option, all the selected files are opened immediately and appended to each other. If you disable this option, only the first file is opened.

7 General handling of databases

This chapter provides a general overview of relational databases and describes some basic techniques for handling database data.

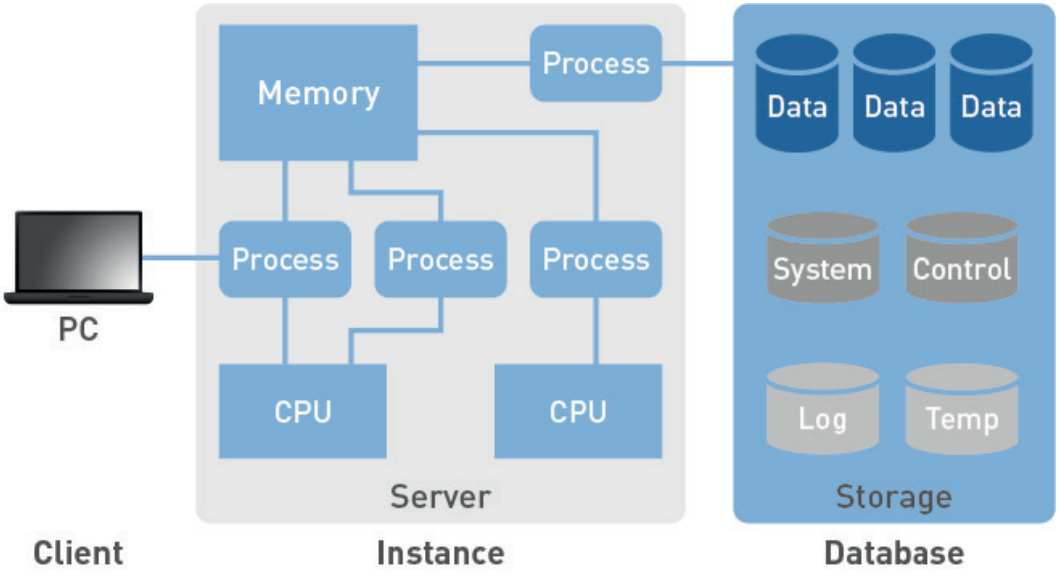
Database basics

A database is an organized collection of data that is typically stored in an electronic format. In a database, data can be quickly input, managed, organized and retrieved.

Traditional databases are organized by datasets (rows) and fields (columns), which are located in tables stored in the database files. Each row within a table corresponds to a single dataset and contains several attributes that describe the dataset.

	_FileId	_TimeStamp	_FileName	_FileType	_Complete	_ErrorOnExtract	coilID	Thickness_AVGt
1	1218089036	2008-12-17 10:52:40.000	C:\iba\training\dat\QDR\471100017.dat	QDR2	1	0	471100017	1.08266
2	1218354195	2008-12-17 10:47:40.000	C:\iba\training\dat\QDR\471100015.dat	QDR2	1	0	471100015	0.959364
3	1218732091	2008-12-17 10:51:48.000	C:\iba\training\dat\QDR\471100016.dat	QDR2	1	0	471100016	0.922059
4	1218904531	2008-12-17 10:46:23.000	C:\iba\training\dat\QDR\471100014.dat	QDR2	1	0	471100014	0.894734

The following figure shows the general structure in database management systems.



Data types and default data type categories

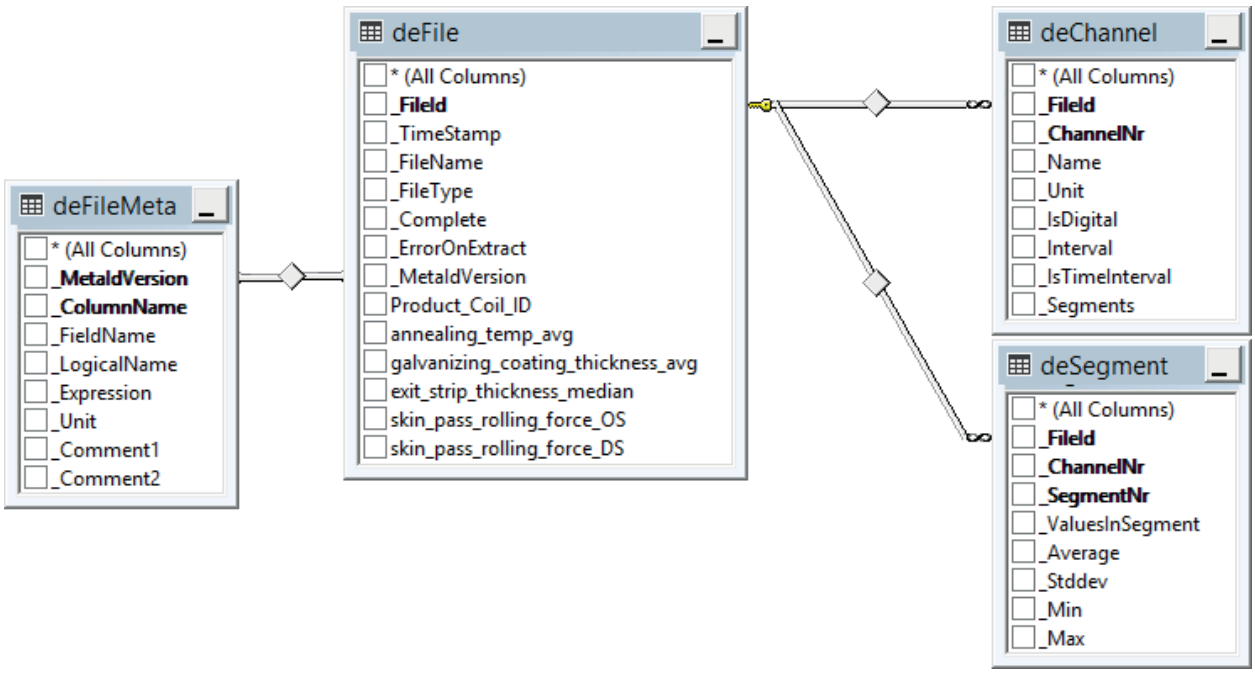
Databases use the following data type categories and data types by default.

- Exact numeric (bigint, bit, decimal, int, money, numeric, smallint)
- Approximate numeric (float, real)
- Date and time (date, datetime2, datetime, datetimeoffset, time)
- Character strings (char, varchar, text)
- Unicode character strings (nchar, ntext, nvarchar)
- Binary strings (binary, varbinary, image)
- Other data types (cursor, time stamp, unique identifier, table)
- Large value data types (varchar(max), nvarchar(max))
- Large object data types (text, ntext, image, xml)

Relational databases

A relational database is a collection of data tables that are formally described and organized according to the relational model. Each table must identify a column or group of columns using the primary key in order to uniquely identify each row.

ibaAnalyzer creates the following types of tables, see ↗ *Database format options*, page 57.



Referential integrity (RI) is a database concept used to ensure that the relationships between database tables remain synchronized during data changes and that the data is consistent. RI may also be helpful in optimizing the database environment and can assist in the early detection of errors.

A combination of primary key and foreign key constraints can be used to help enforce referential integrity of the database. In addition to a foreign key referring to a primary key constraint, a foreign key can also refer to a UNIQUE constraint to help maintain referential integrity.

Normalizing a database

Normalization is the process of organizing data in a database that includes creating tables and establishing relationships between the tables. The following normalization forms are used to help eliminate redundant data.

Five normalization forms (NFs)

- 1NF: Eliminate repeating groups
- 2NF: Eliminate redundant data
- 3NF: Eliminate columns not dependent on key
- 4NF: Isolate independent multiple relationships
- 5NF: Isolate semantically related multiple relationships

7.1 SQL statements for ibaAnalyzer

The following chapters describe the most common SQL statements for *ibaAnalyzer*. The examples are based on the MS SQL server syntax. Other databases (Oracle, IBM-DB2, etc.) may use different SQL syntaxes.

Categories of SQL statements

- Data manipulation language (DML):
  Statements to query and manipulate data (modify, insert, delete)
  e.g. SELECT, INSERT, UPDATE, DELETE
- Data definition language (DDL):
  Statements to define the database scheme or objects
  e.g. CREATE, ALTER, DROP
- Data control language (DCL):
  Statements for rights management
  e.g. GRANT, REVOKE, DENY

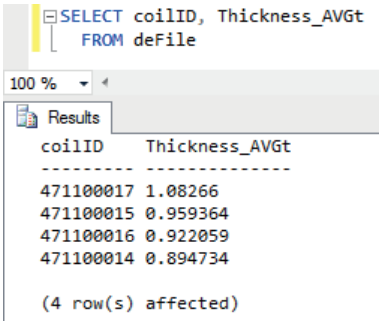
7.1.1 SELECT statements

Items of SELECT statement

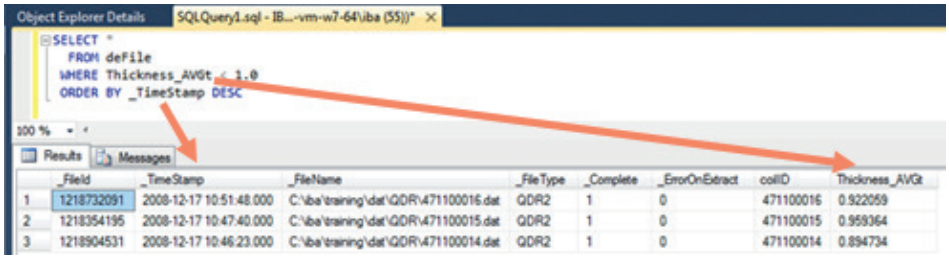
Clause	Expression
SELECT	**<select list>**
FROM	**<table source>**
WHERE	<search condition>
GROUP BY	<group by list>
ORDER BY	<order by list>

Examples:

Values:



Conditions and sorting:



7.1.2 AND, OR, NOT, BETWEEN, LIKE

```
SELECT *
  FROM deFile
 WHERE Thickness_AVGt < 1.0
   AND Thickness_AVGt > 0.9
    OR _TimeStamp NOT BETWEEN '01.01.2008' AND '01.01.2010'
    OR _FileName like '%17.dat'
```

100 %

Results | Messages

	_FileId	_TimeStamp	_FileName	_FileType	_Complete	_ErrorOnExtract	coilID	Thickness_AVGt
1	1218089036	2008-12-17 10:52:40.000	C:\iba\training\dat\QDR\471100017.dat	QDR2	1	0	471100017	1.08266
2	1218354195	2008-12-17 10:47:40.000	C:\iba\training\dat\QDR\471100015.dat	QDR2	1	0	471100015	0.959364
3	1218732091	2008-12-17 10:51:48.000	C:\iba\training\dat\QDR\471100016.dat	QDR2	1	0	471100016	0.922059

7.1.3 Calculations

```
SELECT Thickness_AVGt,
       (Thickness_AVGt + Thickness_AVGt) / 2 AS "calc test",
       _Timestamp,
       _Timestamp + 1                    AS "Shift 1 Day",
       _TimeStamp + ( 1 / 24 / 60 / 60)  AS "Shift 1 Second - wrong",
       _TimeStamp + ( 1.0 / 24 / 60 / 60) AS "Shift 1 Second - better",
       dateadd(ss, 1, _TimeStamp )  AS "Shift 1 Second - best"
  FROM deFile
```

100 %

Results | Messages

	Thickness_AVGt	calc test	_Timestamp	Shift 1 Day	Shift 1 Second - wrong	Shift 1 Second - better	Shift 1 Second - best
1	1.08266	1.08266	2008-12-17 10:52:40.000	2008-12-18 10:52:40.000	2008-12-17 10:52:40.000	2008-12-17 10:52:40.997	2008-12-17 10:52:41.000
2	0.959364	0.959364	2008-12-17 10:47:40.000	2008-12-18 10:47:40.000	2008-12-17 10:47:40.000	2008-12-17 10:47:40.997	2008-12-17 10:47:41.000
3	0.922059	0.922059	2008-12-17 10:51:48.000	2008-12-18 10:51:48.000	2008-12-17 10:51:48.000	2008-12-17 10:51:48.997	2008-12-17 10:51:49.000
4	0.894734	0.894734	2008-12-17 10:46:23.000	2008-12-18 10:46:23.000	2008-12-17 10:46:23.000	2008-12-17 10:46:23.997	2008-12-17 10:46:24.000

7.1.4 Advanced SELECT clauses (DISTINCT, TOP, CASE)

DISTINCT: Returns available channel names and units.

```
select DISTINCT _Name, _Unit from [IBA].[dbo].[deChannel]
```

100 %

Results | Messages

	_Name	_Unit
1	Comment 1 a__100	A
2	Comment 1 b__100	A
3	Comment 1 c__100	A
4	Comment 1 d__100	A
5	Comment 1 e__100	
6	current off	
7	GP: Rectifier 1 current	A
8	GP: Rectifier 2 current	A
9	GP: Rectifier 3 current	A
10	GP: Rectifier 4 current	A

TOP x: Selects top x files from a table and sort them ascending (ASC) or descending (DESC).

	_FileId	_TimeStamp	_FileName	_
1	-1138357270	2017-01-29 23:41:55.000	D:\IBA\dat_files\dat-training\pda_training018.dat	r
2	-1138094611	2017-01-29 23:21:55.000	D:\IBA\dat_files\dat-training\pda_training016.dat	r
3	-1137962969	2017-01-29 23:51:55.000	D:\IBA\dat_files\dat-training\pda_training019.dat	r
4	-1137699981	2017-01-29 23:31:55.000	D:\IBA\dat_files\dat-training\pda_training017.dat	r
5	-1137434540	2017-01-29 23:11:55.000	D:\IBA\dat_files\dat-training\pda_training015.dat	r
6	-1137320619	2017-01-30 00:01:55.000	D:\IBA\dat_files\dat-training\pda_training020.dat	r
7	-1137112986	2017-01-30 00:11:55.000	D:\IBA\dat_files\dat-training\pda_training021.dat	r

```
select TOP 2
        *
  from [IBA].[dbo].[deFile]
 order by _Timestamp DESC
```

100 %

Results Messages

	_FileId	_TimeStamp	_FileName
1	-1137112986	2017-01-30 00:11:55.000	D:\IBA\dat_files\dat-training\pda_training021.dat
2	-1137320619	2017-01-30 00:01:55.000	D:\IBA\dat_files\dat-training\pda_training020.dat

CASE: Displays results according to set conditions.

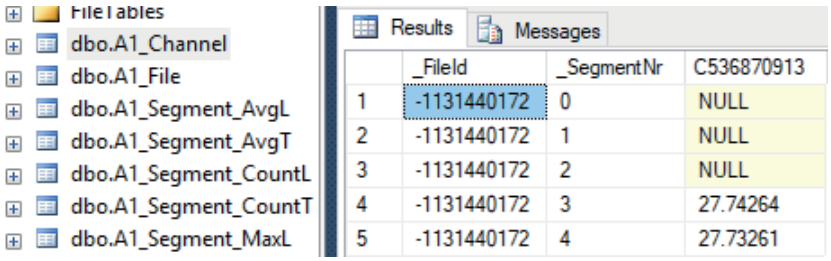
```
select [_FileName],
       [MinimumRectifier2current],
       CASE WHEN  [MinimumRectifier2current] < 105
            THEN 0
            ELSE [MinimumRectifier2current]
        END AS "Current less than 105"
  from [IBA].[dbo].[deFile]
```

100 %

Results Messages

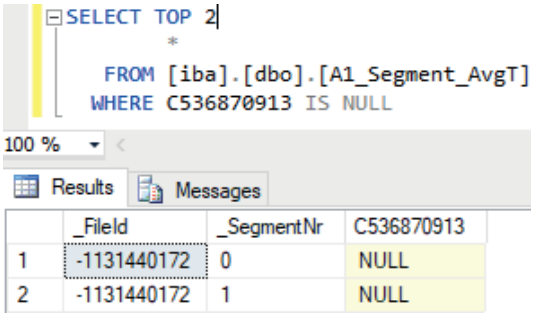
	_FileName	MinimumRectifier2current	Current less than 105
1	D:\IBA\dat_files\dat-training\pda_training018.dat	119.924	119.924
2	D:\IBA\dat_files\dat-training\pda_training016.dat	88.1941	0
3	D:\IBA\dat_files\dat-training\pda_training019.dat	128.955	128.955
4	D:\IBA\dat_files\dat-training\pda_training017.dat	134.488	134.488
5	D:\IBA\dat_files\dat-training\pda_training015.dat	102.595	0
6	D:\IBA\dat_files\dat-training\pda_training020.dat	126.108	126.108
7	D:\IBA\dat_files\dat-training\pda_training021.dat	103.815	0

7.1.5 Dealing with NULL



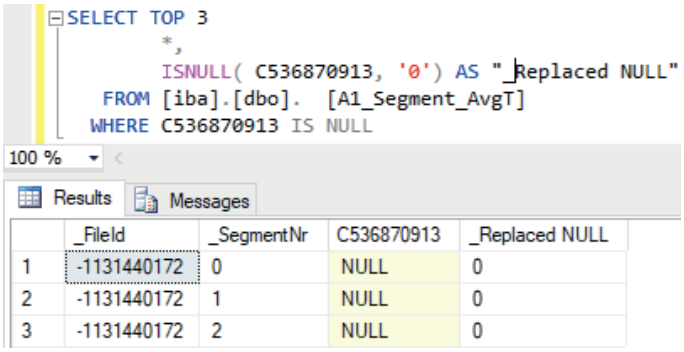
	_FileId	_SegmentNr	C536870913
1	-1131440172	0	NULL
2	-1131440172	1	NULL
3	-1131440172	2	NULL
4	-1131440172	3	27.74264
5	-1131440172	4	27.73261

Selecting top 2 "NULL" segments:



	_FileId	_SegmentNr	C536870913
1	-1131440172	0	NULL
2	-1131440172	1	NULL

Replacing "NULL" with "0":

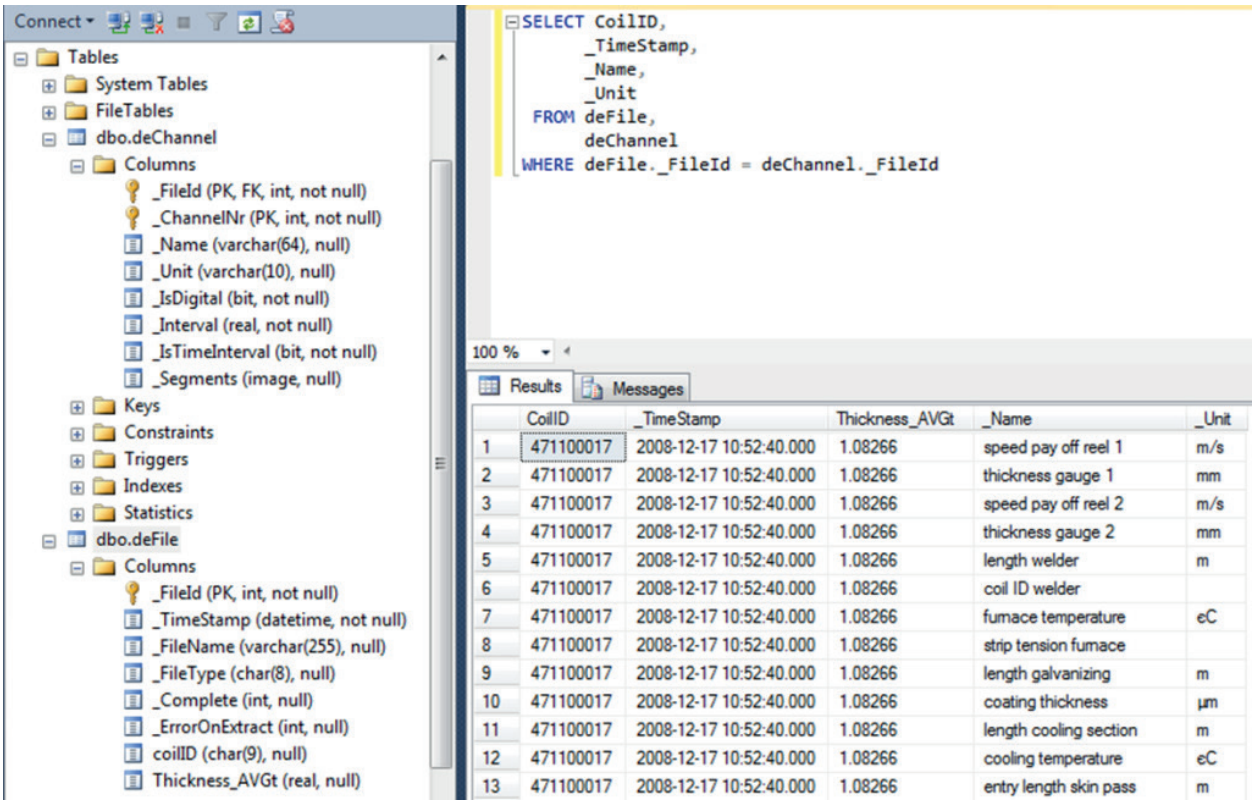


	_FileId	_SegmentNr	C536870913	_Replaced NULL
1	-1131440172	0	NULL	0
2	-1131440172	1	NULL	0
3	-1131440172	2	NULL	0

7.1.6 JOIN clause

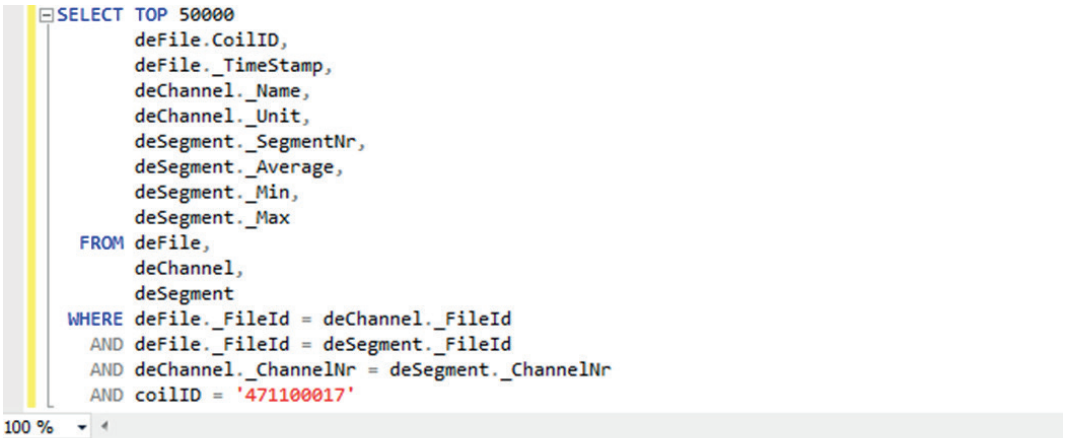
The JOIN clause combines related data from multiple tables into one set of results.

Selecting data from two tables with a common _FileId.



	CoilID	_TimeStamp	Thickness_AVGt	_Name	_Unit
1	471100017	2008-12-17 10:52:40.000	1.08266	speed pay off reel 1	m/s
2	471100017	2008-12-17 10:52:40.000	1.08266	thickness gauge 1	mm
3	471100017	2008-12-17 10:52:40.000	1.08266	speed pay off reel 2	m/s
4	471100017	2008-12-17 10:52:40.000	1.08266	thickness gauge 2	mm
5	471100017	2008-12-17 10:52:40.000	1.08266	length welder	m
6	471100017	2008-12-17 10:52:40.000	1.08266	coil ID welder	
7	471100017	2008-12-17 10:52:40.000	1.08266	furnace temperature	eC
8	471100017	2008-12-17 10:52:40.000	1.08266	strip tension furnace	
9	471100017	2008-12-17 10:52:40.000	1.08266	length galvanizing	m
10	471100017	2008-12-17 10:52:40.000	1.08266	coating thickness	µm
11	471100017	2008-12-17 10:52:40.000	1.08266	length cooling section	m
12	471100017	2008-12-17 10:52:40.000	1.08266	cooling temperature	eC
13	471100017	2008-12-17 10:52:40.000	1.08266	entry length skin pass	m

Selecting data from three tables with common _FileId and _ChannelNr for top 50000 rows.



	CoilID	_TimeStamp	_Name	_Unit	_SegmentNr	_Average	_Min	_Max
1	471100017	2008-12-17 10:52:40.000	speed pay off reel 2	m/s	0	0.9999964	0.9999963	0.9999964
2	471100017	2008-12-17 10:52:40.000	speed pay off reel 2	m/s	1	3.82224	0.9666631	10.03296
3	471100017	2008-12-17 10:52:40.000	speed pay off reel 2	m/s	2	13.47231	10.03296	14.99915
4	471100017	2008-12-17 10:52:40.000	speed pay off reel 2	m/s	3	14.99914	14.99912	14.99914

7.1.7 Aggregate functions

```
SELECT  deFile.CoilID,
        deFile._TimeStamp,
        deChannel._Name,
        deChannel._Unit,
        deChannel._ChannelNr,
        COUNT(deSegment._SegmentNr) AS "# Segments",
        AVG(deSegment._Average) AS "Avg Avg",
        MIN(deSegment._Min) AS "Min Min",
        MAX(deSegment._Max) AS "Max Max"
   FROM deFile
   INNER JOIN deChannel ON deFile._FileId = deChannel._FileId
   INNER JOIN deSegment ON deChannel._FileId = deSegment._FileId AND deChannel._ChannelNr = deSegment._ChannelNr
   WHERE deChannel._Name like '%force%'
   GROUP BY deFile.CoilID,
        deFile._TimeStamp,
        deChannel._Name,
        deChannel._Unit,
        deChannel._ChannelNr
```

100 %

Results | Messages

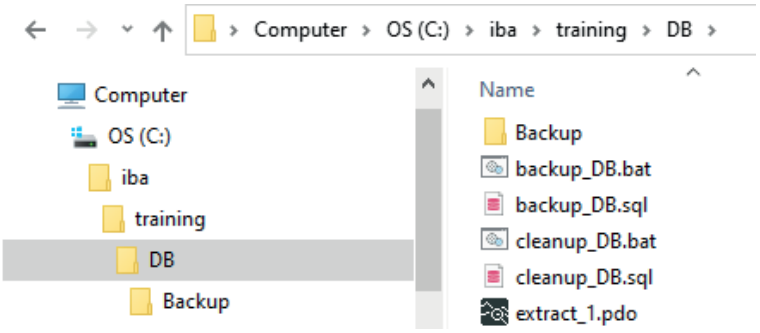
	CoilID	_TimeStamp	_Name	_Unit	_ChannelNr	# Segments	Avg Avg	Min Min	Max Max
1	471100014	2008-12-17 10:46:23.000	rolling force AS	N	-2142896007	72	23.6523003458149	0.001651668	29.51965
2	471100014	2008-12-17 10:46:23.000	rolling force BS	N	-2142896006	72	23.8890460899307	0.001166938	29.93768
3	471100015	2008-12-17 10:47:40.000	rolling force AS	N	-2142896007	246	24.6971272957034	0.004510283	28.22289
4	471100015	2008-12-17 10:47:40.000	rolling force BS	N	-2142896006	246	24.9437023992461	0.007483545	28.53873
5	471100016	2008-12-17 10:51:48.000	rolling force AS	N	-2142896007	47	23.0007387735742	0.00595638	41.14486
6	471100016	2008-12-17 10:51:48.000	rolling force BS	N	-2142896006	47	23.2309047922175	0.01031394	41.18505
7	471100017	2008-12-17 10:52:40.000	rolling force AS	N	-2142896007	122	24.2788492758743	0.01424175	28.80285
8	471100017	2008-12-17 10:52:40.000	rolling force BS	N	-2142896006	122	24.5154788455025	0.00861586	29.26654

7.2 SQL scripts and scheduling

T-SQL batches are collections of one or more T-SQL statements sent to SQL Server as a unit for parsing, optimization and execution. T-SQL batches end with the GO clause.

Some statements (e.g. CREATE FUNCTION, CREATE PROCEDURE, CREATE VIEW) may not be combined with others in the same batch.

Useful applications for backup and clean up



cleanup_DB.bat

```
SQLCMD -S .\SQLEXPRESS -i C:\iba\training\DB\cleanup_DB.sql
```

cleanup_DB.sql, example from ibaDatawyzer-ICC-system:

```
USE IBA
GO
DELETE iba_genealogic_file WHERE [_TimeStamp] < getdate() - 365
GO
DELETE iba_fingerprint_file
WHERE [_FileId] NOT IN (SELECT [_FileId] FROM iba_genealogic_file)
GO
```

backup_DB.bat

```
SQLCMD -S .\SQLEXPRESS -i C:\iba\training\DB\backup_DB.sql
```

backup_DB.sql:

```
BACKUP DATABASE IBA_TTT
TO DISK = N'C:\iba\training\DB\Backup\IBA_TTT.bak'
WITH NOFORMAT, INIT,
NAME = N'IBA_TTT Backup',
SKIP, NOREWIND, NOUNLOAD, STATS = 10
```

8 Extraction to database via the command line

Note



To use the command line switches for extraction, the *ibaDatCoordinator-DB* license is required.

If you execute *ibaAnalyzer* via the command line, some of the command line switches for *ibaAnalyzer* are particularly suitable for extraction in databases. This allows post-processing to be automated. The following syntax is used:

```
ibaAnalyzer.exe datfilename1 [datfilename2] … [datfilenamen]
[pdofilename] [/switch]
```

ibaAnalyzer then opens with various DAT files, an analysis and various options, which are described below. In each case, *ibaAnalyzer* automatically opens all specified DAT files and carries out the assigned tasks. Please note that the analysis is mandatory in some cases, e.g. if the information required for the task is only available in the analysis file.

The following command line switches are suitable for extraction in databases. For a detailed description of the switches see part 1 of the *ibaAnalyzer* manual, section *Using switches in the command line*.

- /extract[:filename]
- /append
- /reuse
- /print
- /sql:filename.sql[;sync:"syncFieldName"]
- /trendsql:filename.sql
- /overviewsql:filename.sql
- /report[:filename]

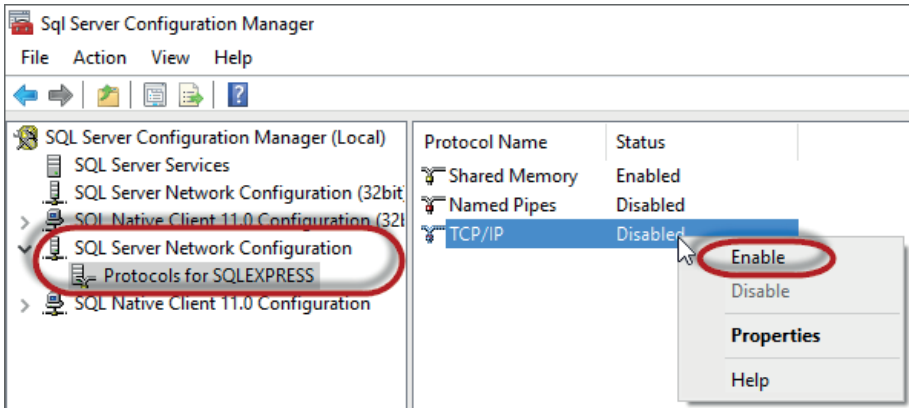
9 Remote access to an SQL Server

For remote access to the SQL Server, you have to change the configuration of the SQL Server first, see ↗ *Configuring remote access, page 96*.

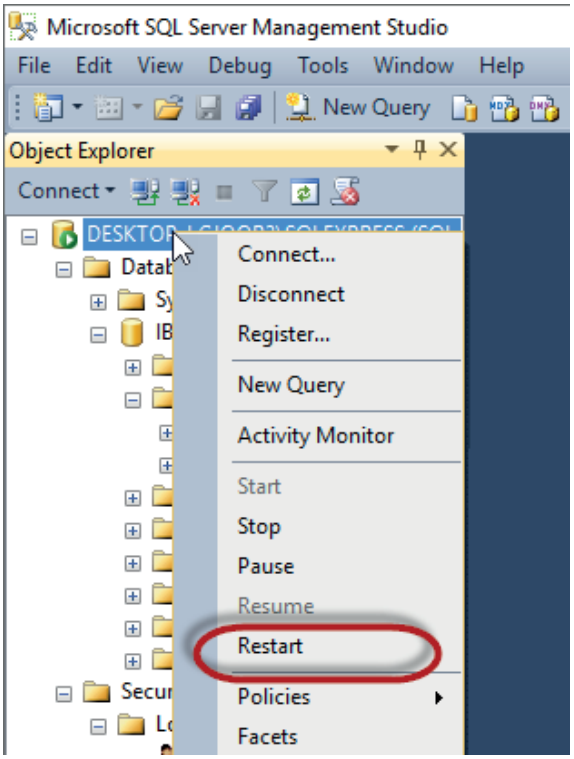
Additionally, you have to adjust the firewall settings, see ↗ *Firewall settings for remote access, page 97*.

9.1 Configuring remote access

1. Open the SQL Server Configuration Manager and allow the program to make changes to your computer.
2. Under *SQL Server Network Configuration – Protocols for SQLEXPRESS* right-click on *TCP/IP* and enable the TCP/IP protocol.



3. Restart the SQL Server via the Microsoft SQL Server Management Studio for the changes to take effect.



9.2 Firewall settings for remote access

For remote access to the SQL server, you have to configure three inbound rules in the Windows firewall. These inbound rules address the following:

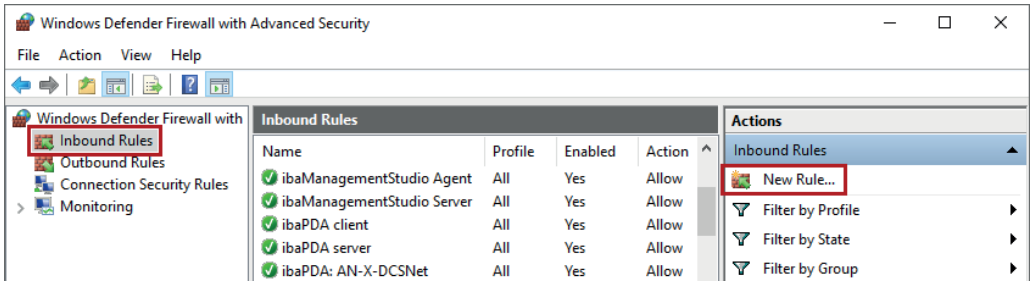
- Executable file `sqlservr.exe`
- TCP-Port 1433
- UDP-Port 1434

To configure these rules, open the dialog for the advanced firewall settings *Windows Defender Firewall with advanced security*, if necessary via the *Windows Defender Firewall* settings.

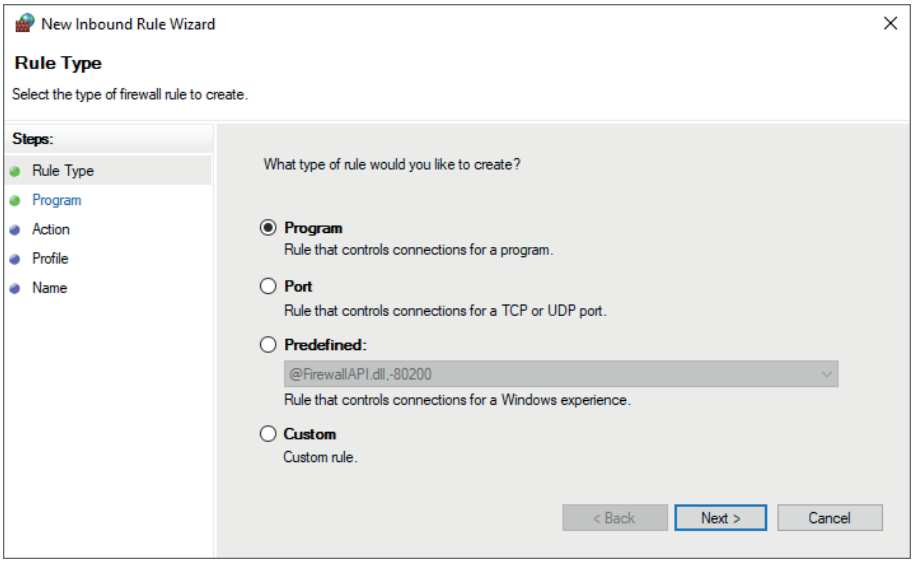
After configuring these rules, test the connection to the database in *ibaAnalyzer*.

Creating an inbound rule for sqlservr.exe

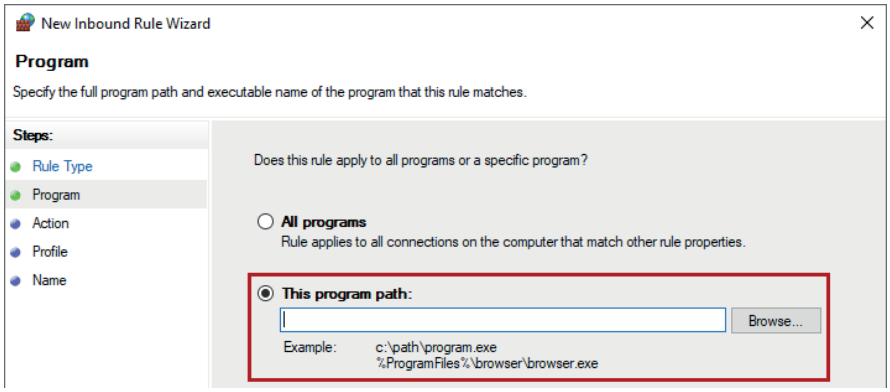
1. Define a new inbound rule.



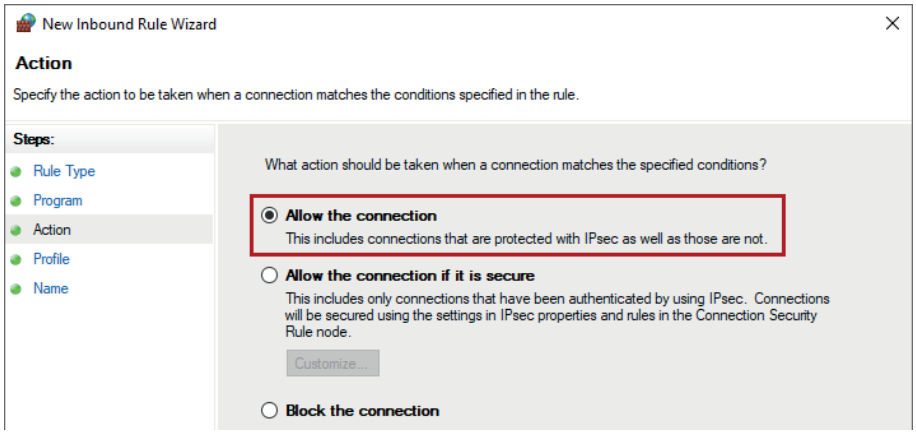
2. In the rule wizard, select the *Program* rule type and click <Next>.



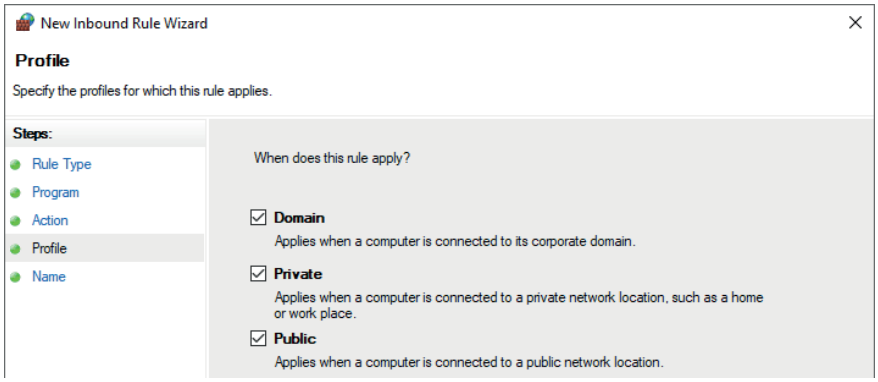
3. Select *This program path* and enter the path to `sqlservr.exe`
   Click <Next>.



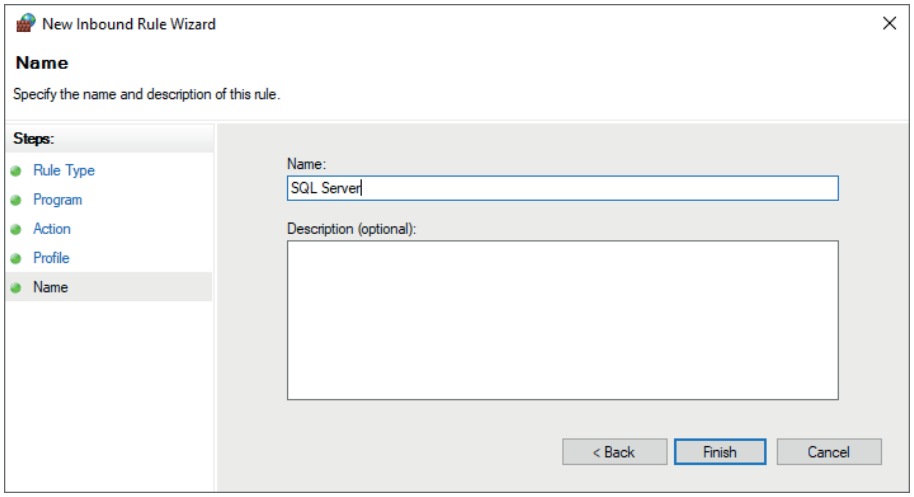
4. Select the operation *Allow the connection* and click <Next>.



5. Select the profiles and click <Next>.



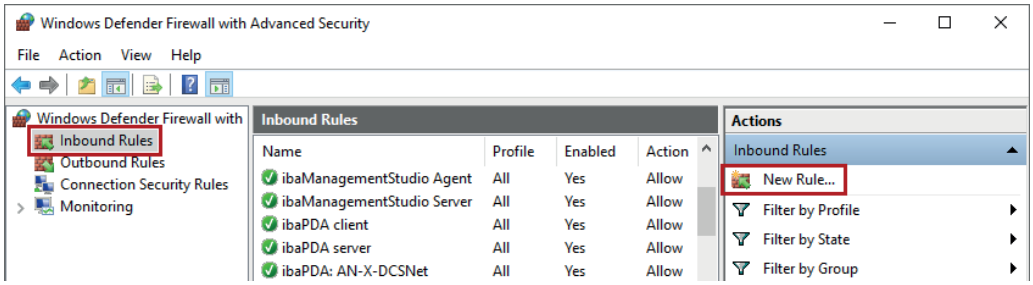
6. Specify a name for the rule and, if needed, a description.
   Click <Finish>.



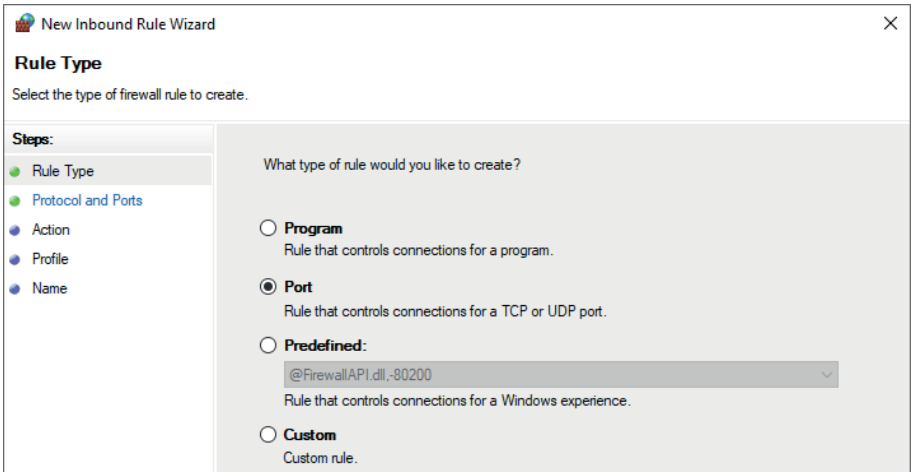
Creating inbound rules for TCP port 1433 and UDP port 1434

Carry out the following steps once for the TCP port and once for the UDP port.

1. Define a new inbound rule.



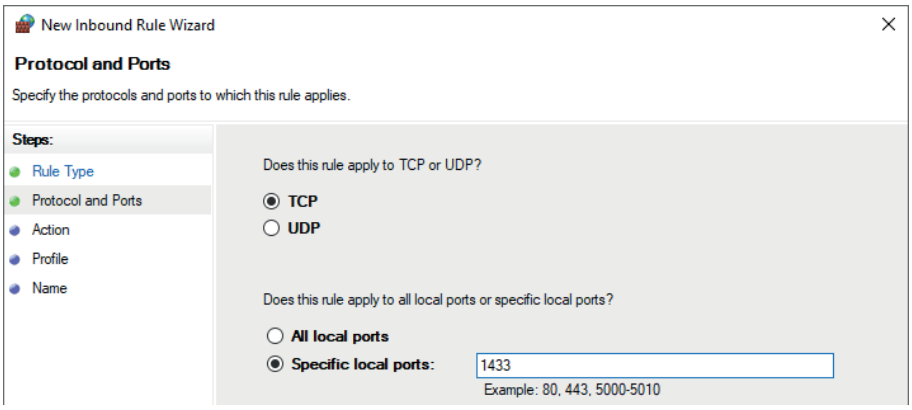
2. Select the *Port* rule type and click <Next>.



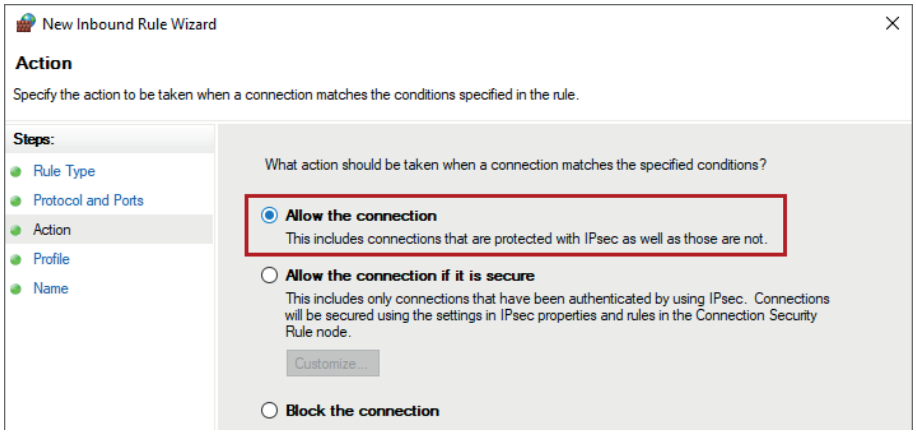
3. **For the TCP port:** Select *TCP* and enter the port "1433" in *Specific local ports*.

   For the UDP port: Select *UDP* and enter the port "1434" in *Specific local ports*.

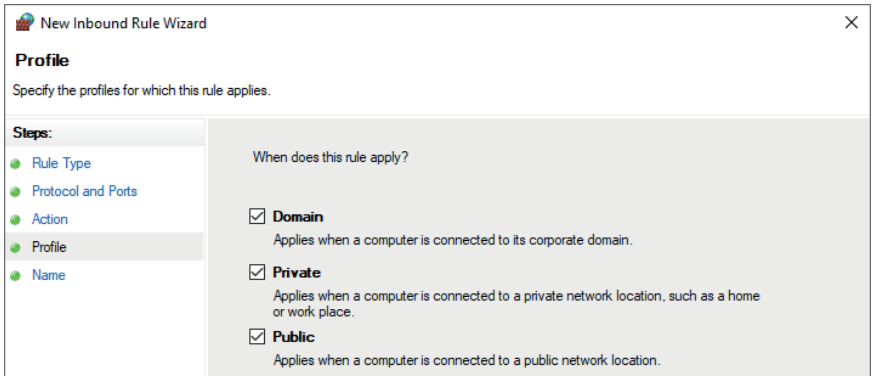
   Click <Next>.



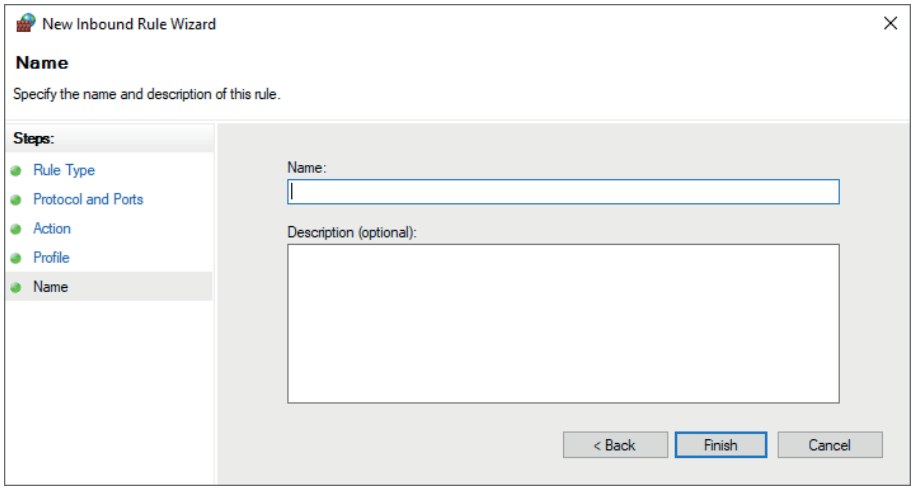
4. Select the operation *Allow the connection* and click <Next>.



5. Select the profiles and click <Next>.



6. Specify a name for the rule and, if needed, a description.
   Click <Finish>.



10 Support and contact

Support

Phone: +49 911 97282-14

Email: support@iba-ag.com

Note



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

Product Security Incident Response Team (PSIRT)

Current advisories: www.iba-ag.com/en/security

Email: psirt@iba-ag.com

Contact

Headquarters
iba AG
Gebhardtstrasse 10–20
90762 Fuerth
Germany

Phone: +49 911 97282-0

Email: iba@iba-ag.com

Mailing address
iba AG
Postbox 1828
D-90708 Fuerth, Germany

Delivery address
iba AG
Gebhardtstrasse 10
90762 Fuerth, Germany

Regional and worldwide

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

www.iba-ag.com