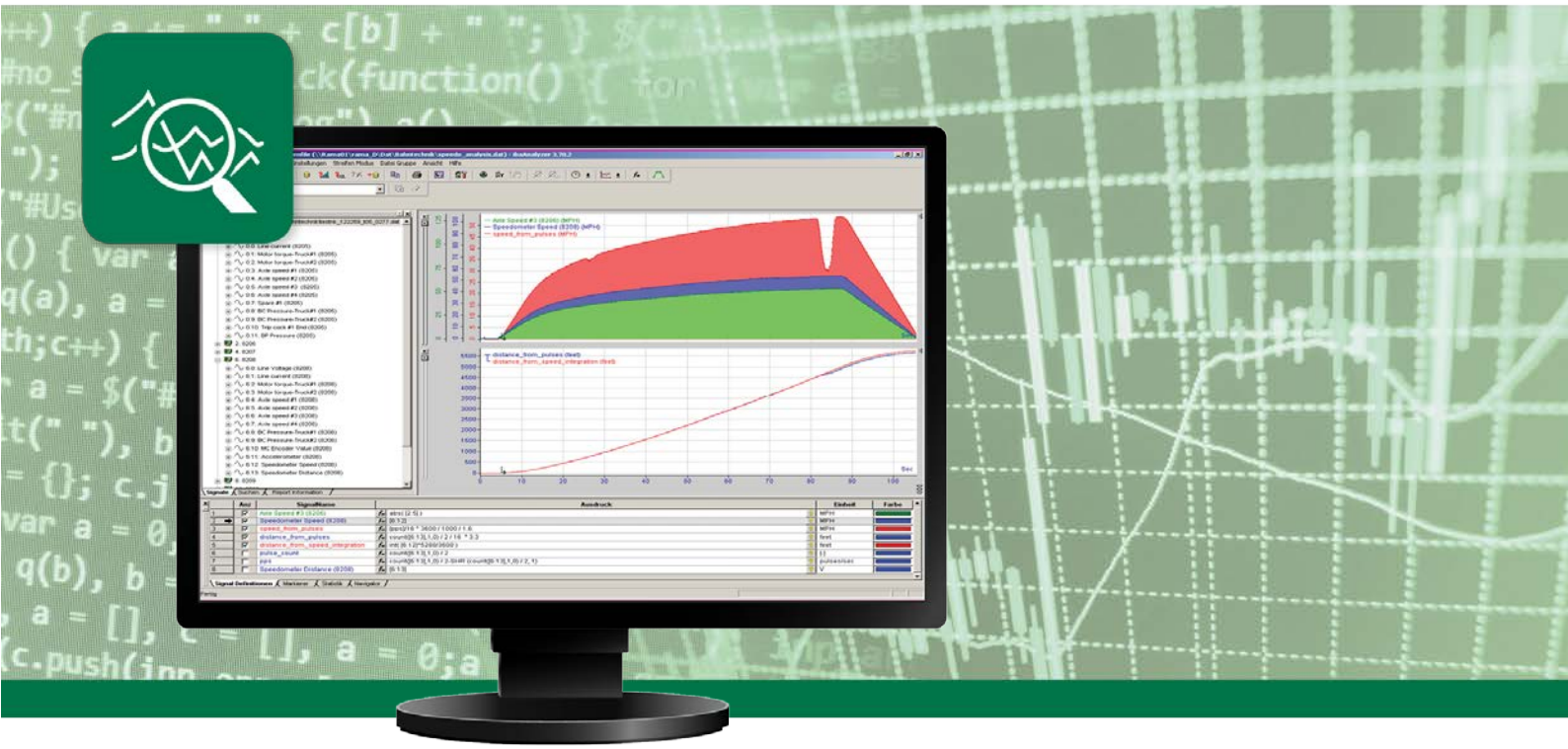
See the Big Picture



ibaAnalyzer-DB

Flexible Database Integration for iba Systems

Manual

Issue 2.0

Measurement and Automation Systems

Manufacturer

iba AG

Koenigswarterstr. 44

90762 Fuerth

Germany

Contacts

Main office	+49 911 97282-0
Fax	+49 911 97282-33
Support	+49 911 97282-14
Engineering	+49 911 97282-13
E-Mail	iba@iba-ag.com
Web	www.iba-ag.com

The content of this publication has been checked for compliance with the described hardware and software. Nevertheless, deviations cannot be excluded completely so that the full compliance is not guaranteed. However, the information in this publication is updated regularly. Required corrections are contained in the following issues or can be downloaded from the Internet.

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

Issue	Date	Revision	Author	Version SW
2.0	02-01-2018	Revised edition	RM	6.10.0

Table of Contents

1 About this manual

This document describes the function and application of the software *ibaAnalyzer-DB*.

1.1 Target group

This manual addresses in particular the qualified professionals who are familiar with handling electrical and electronic modules as well as communication and measurement technology. A person is regarded as professional if he/she is capable of assessing safety and recognizing possible consequences and risks on the basis of his/her specialist training, knowledge and experience and knowledge of the standard regulations.

This documentation 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-V6* (creation and structure of the measuring data files)
- ❑ For DB extraction, knowledge about using the database system in question

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>
File names, paths	"Filename", "Path" Example: "Test.doc"

1.3 Used symbols

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

⚠ DANGER

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

- ❑ From an electric shock!
- ❑ Due to the improper handling of software products which are coupled to input and output procedures with control function!

⚠ WARNING

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

⚠ CAUTION

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

Note

A note specifies special requirements or actions to be observed.

Important note

Note if some special features must be observed, for example exceptions from the rule.

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.

Example

Configuration and application examples for a better understanding

2 Introduction

2.1 What is ibaAnalyzer-DB?

ibaAnalyzer-DB extractor is a purchasable option which provides ETL (Extract Transform Load) functions for commonly used database systems and enables trend and detail analyses based on databases. Also the report engine of *ibaAnalyzer* can process data provided by *ibaAnalyzer-DB*.

2.2 Functions and application

The database extraction functionality is an integral part of current *ibaAnalyzer* versions and is activated during the installation of *ibaAnalyzer* whenever the respective dongle is present. All data in iba data file format (*.dat), coming from *ibaPDA*, *ibaQDR*, *ibaLogic*, or third party applications which have used the ibaFiles library can be extracted, transformed and loaded easily.

iba data files (*.dat), which have a lossless and very efficiently compressed binary format, can be opened by *ibaAnalyzer* permitting the easy handling of large amounts of data. All 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. Thus each data extraction process can lead to a significant reduction of storage space by compressing information. Multiple databases are supported by Microsoft OLE-DB API: SQL-Server (on premises or in MS Azure cloud), Oracle, DB2-UDB or by ODBC: MySQL/Maria-DB, PostgreSQL, SQLite, MS Access, etc. (Please ask iba Support for further information.). These databases may reside on the same PC or may be installed on a separate database server which has a network connection to the *ibaAnalyzer-DB* PC.

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

Note that in most cases, this would be impossible with the huge original data sets.

Alternatively iba provides further extensions of *ibaAnalyzer* for extraction to new *.dat or ASCII files as well as files in COMTRADE format (*ibaAnalyzer-DAT-Extractor*). The values extracted to a database can now be handled by *ibaAnalyzer-DB* as if they were values in .dat files. It is possible to query these data using a query-assistant or direct sql commands. Data aggregated on a file basis can be used for long term analyses (so called trend queries). Moreover these trend queries provide a navigation view for drill-down analysis that can be based on database values or the linked original data files.

Summary of the major functions:

- ❑ Data "Extraction" from iba data file
- ❑ Data "Transformation" or generation of new virtual signals
- ❑ Data "Load" into OLEDB or ODBC data bases
- ❑ Data "Detail Analyses" from database instead of data file
- ❑ Data "Trend Analyses" based on aggregated values
- ❑ Data "Drill Down" option from trend to detail.

2.3 System pre-requisites

Depending on database system:

- ODBC driver and/or Database Client Software (Oracle, DB2-UDB)

Important information

Depending on the availability of the ODBC driver and/or on the Database Client Software (32 or 64 bit), the 32 or 64bit version of *ibaAnalyzer* is required.

Two versions of the installer are available:

- The 32 bit version: ibaAnalyzerInstall_x86_v6.x.y.exe.

- The 64 bit version: ibaAnalyzerInstall_x64_v6.x.y.exe.

Only one of these versions can be installed on one computer (v6.10.0 or higher).

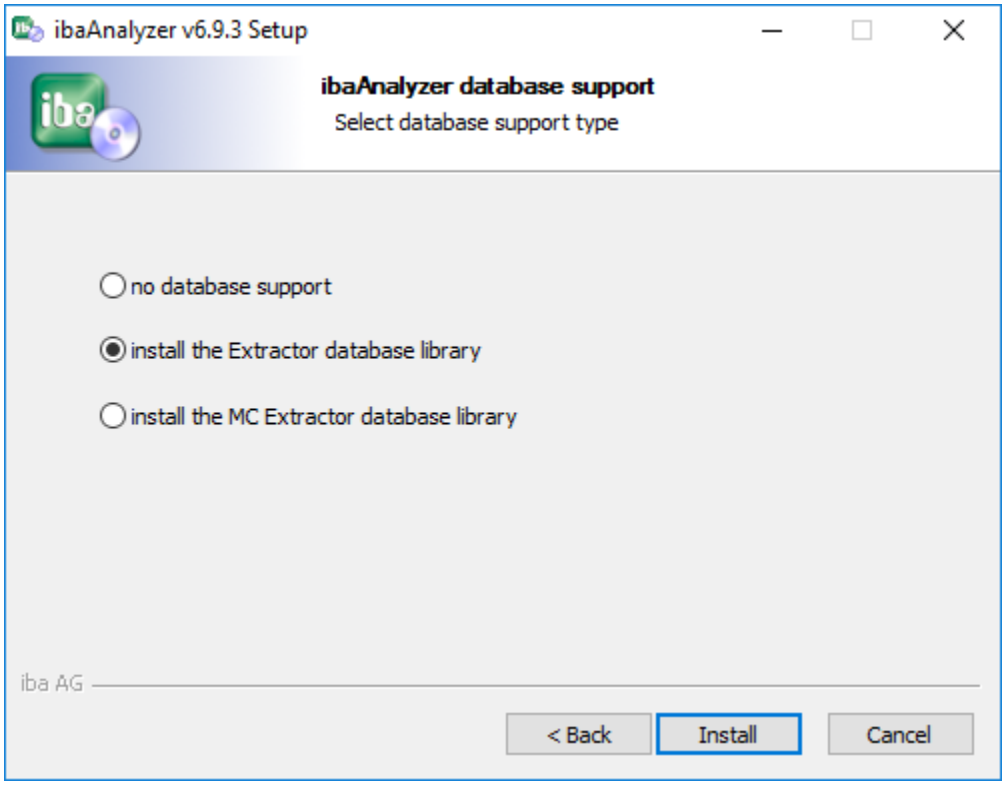
3 Installation

3.1 Installation of ibaAnalyzer database support

ibaAnalyzer-DB is installed with *ibaAnalyzer* and activated by the associated license (dongle).

The following *ibaAnalyzer* installation steps are required for *ibaAnalyzer-DB*:

❑ Choose the database support type when prompted (see Database Format Options)



If *ibaAnalyzer* has been installed without database support it is possible to add it later by starting one of the batch files ("reg_dataextractor.bat" or "reg_dataextractorMC.bat") in the home directory specified during the installation of *ibaAnalyzer* (default: C:\Program Files\iba\ibaAnalyzer) or by rerunning the installation procedure and selecting one of the database options.

3.2 Preparation of the target database

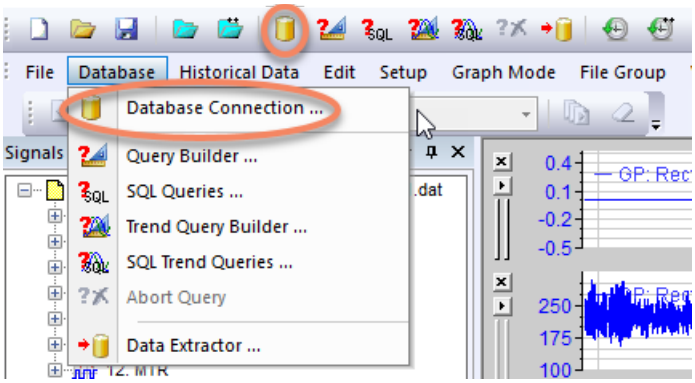
Depending on the target database and kind of application some additional preparation might be necessary:

- ❑ Create database
- ❑ Create structure
- ❑ Create storage (database files, table space,…)
- ❑ Create login (user/password)
- ❑ Configure network access (e.g. database listener)

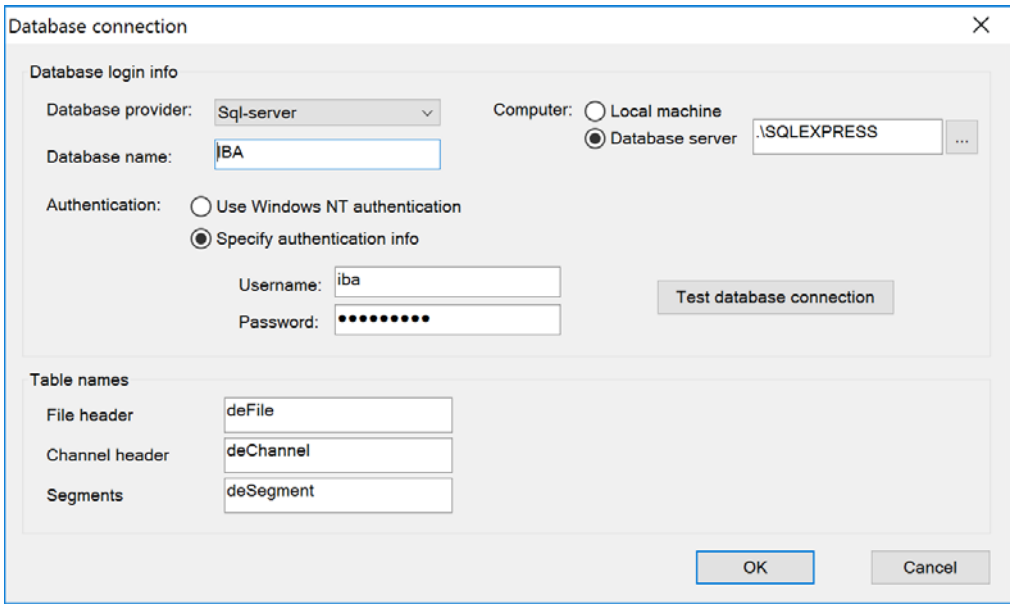
The procedures required to install the necessary database components are described in the chapter "Setting Up Databases".

4 Configuring the database connection

If the required dongle is connected and, if the software components are properly installed, the "Database" menu will be activated in the *ibaAnalyzer* window.



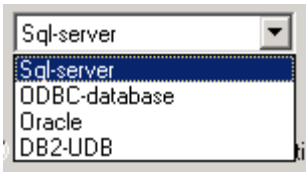
If the database connection has not been previously configured the following default configuration will be shown. Otherwise the previously selected configuration will be shown.



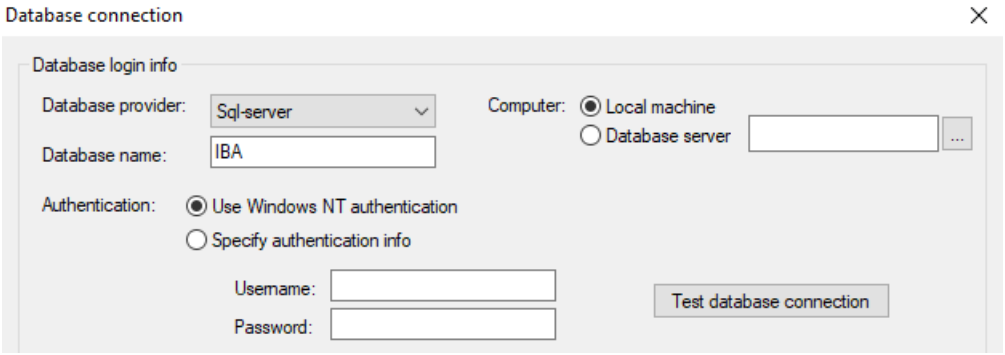
The following describes the steps needed to link *ibaAnalyzer* to the selected and configured database.

4.1 Database connection options

First choose the *Database provider* from the drop down menu.



4.1.1 SQL-Server



❑ Database provider
Select <Sql-server>

❑ Database name
Enter name specified during database installation

❑ Computer
Select <Local machine> or enter the name of the <Database server>\<Instance>

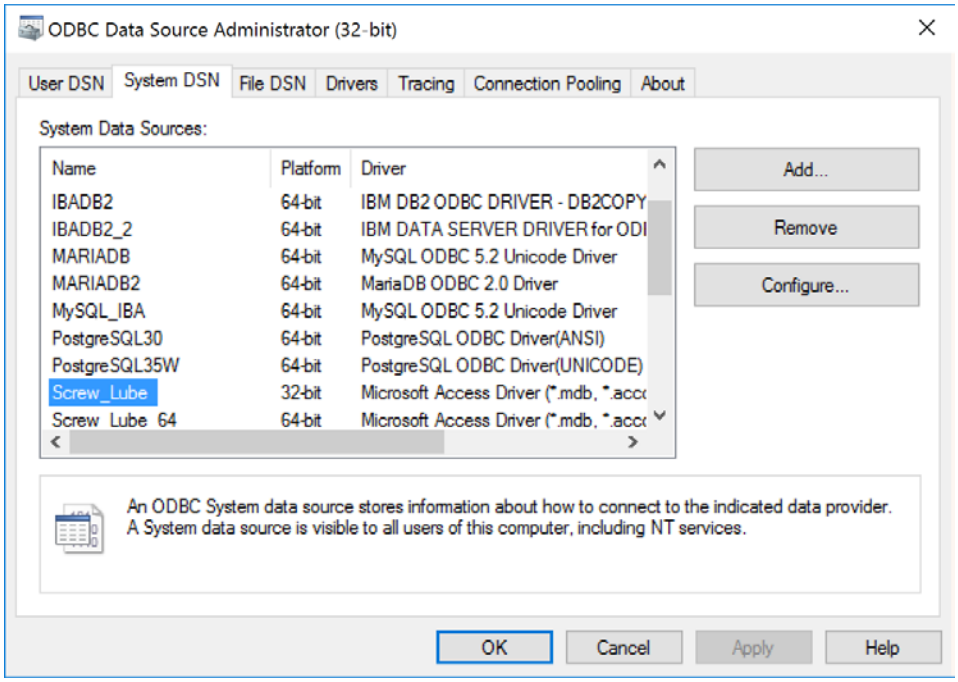
If the Database server is the localhost the following notation with wildcard "." is possible ".\<Instance>". (see example on previous page)

❑ Authentication
Select either:

- Use Windows NT authentication or
- Specify authentication info "Username" and "Password"

4.1.2 ODBC-Database

For ODBC Databases (MySQL/MariaDB, PostgreSQL, MS Access, SQLite, …) use DSN (Data Source Name) configured in the "ODBC Data Source Administrator" (Control Panel "Administrative Tools" Data Sources ODBC):



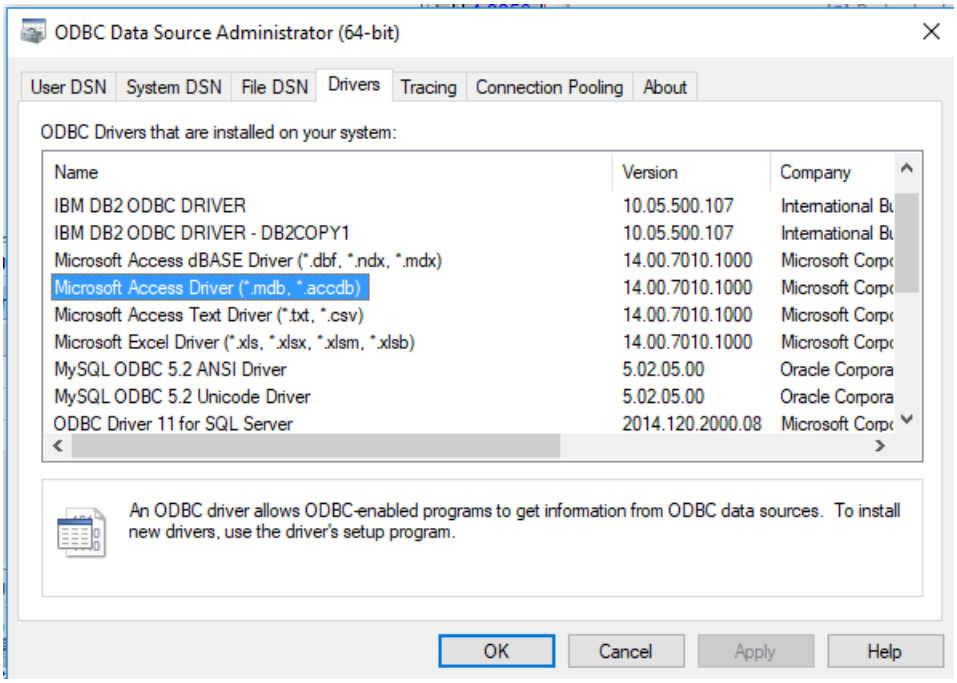
The ODBC DSN Administrator is located in:

- C:\Windows\System32\odbcad32.exe (64bit ODBC on 64bit windows OS)
  or
- C:\Windows\SysWOW64\odbcad32.exe (32bit ODBC on 64bit windows OS)

4.1.2.1 MS-Access

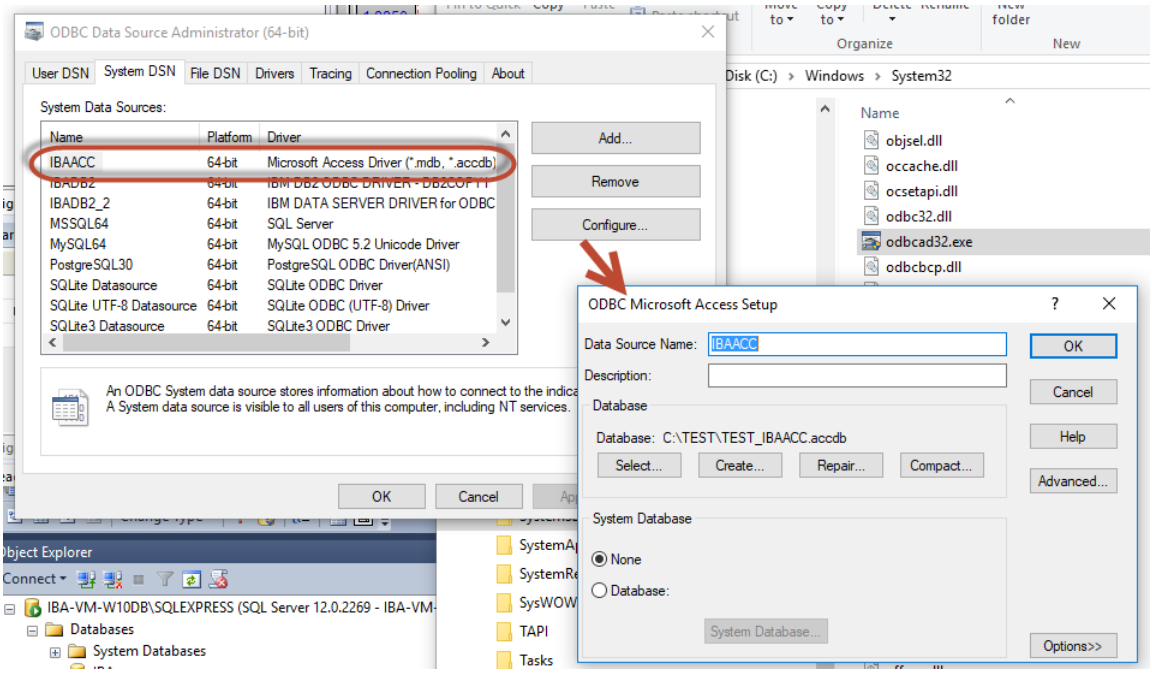
In a simple local test environment with an ODBC connection to Microsoft Access, only the ODBC-DSN needs to be configured.

Verify if MS-Access ODBC driver is available in ODBC DSN Administrator:



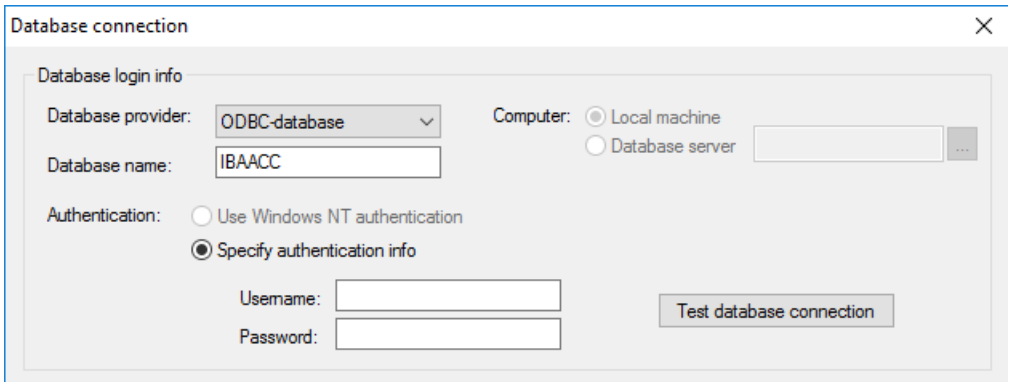
If the driver is not available it can be downloaded from the Microsoft website.

We recommend using/creating a System-DSN:



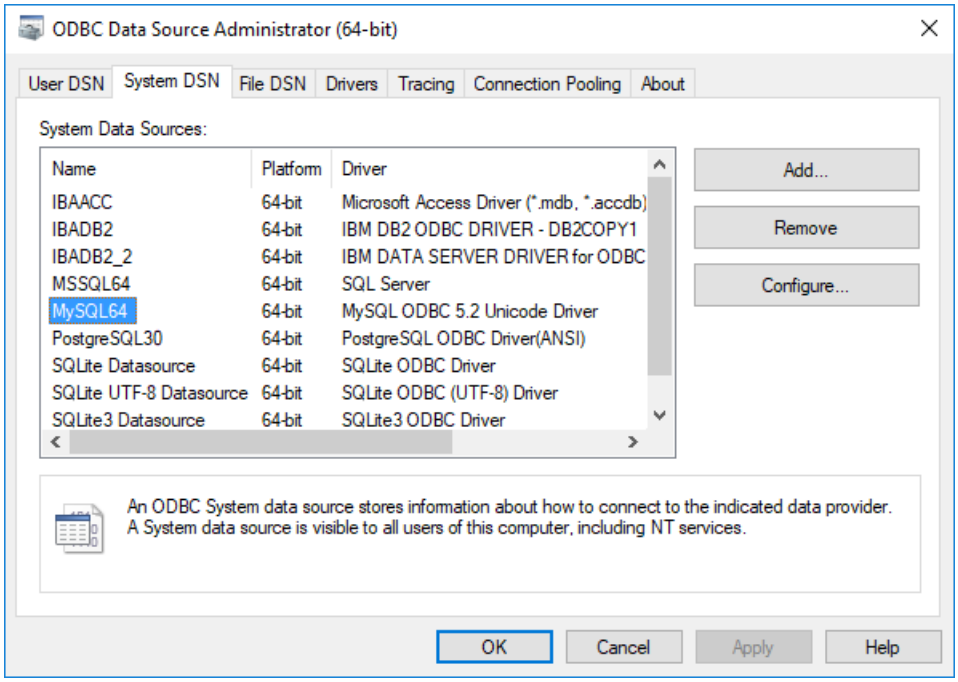
It is possible to use an existing DB "Select…" or to create a new DB "Create…".

Select ODBC-database as "Database provider" and use the DSN (Data Source Name) as "Database name":



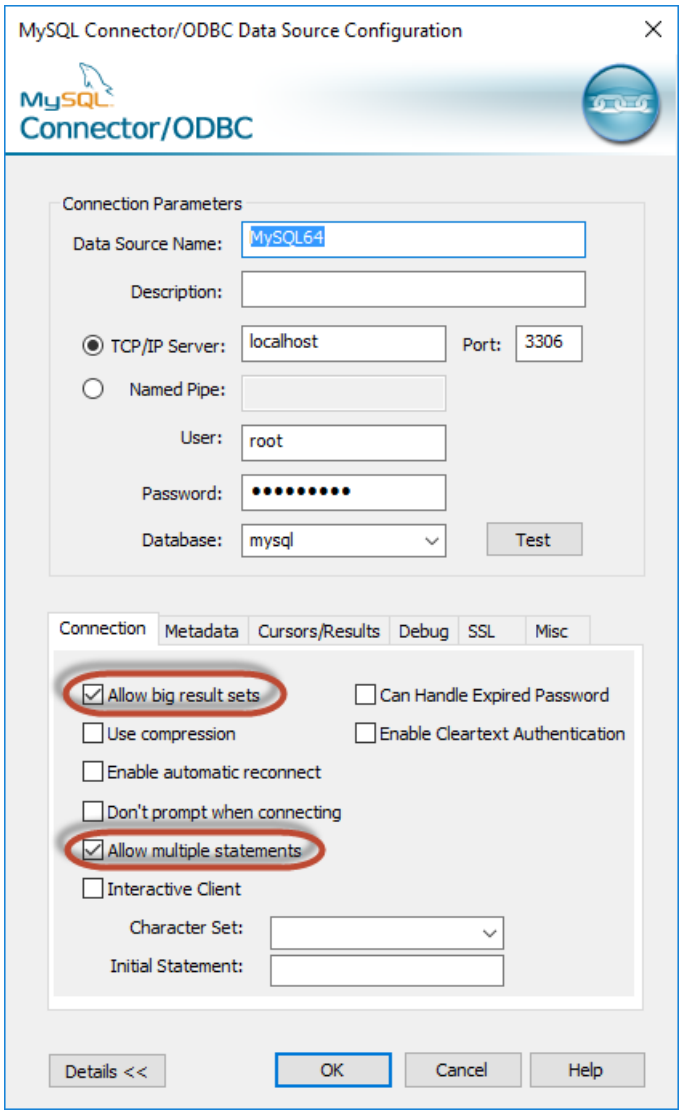
4.1.2.2 MySQL/Maria-DB

For both MySQL and Maria-DB we recommend the ODBC driver (32 or 64 bit) of MySQL.

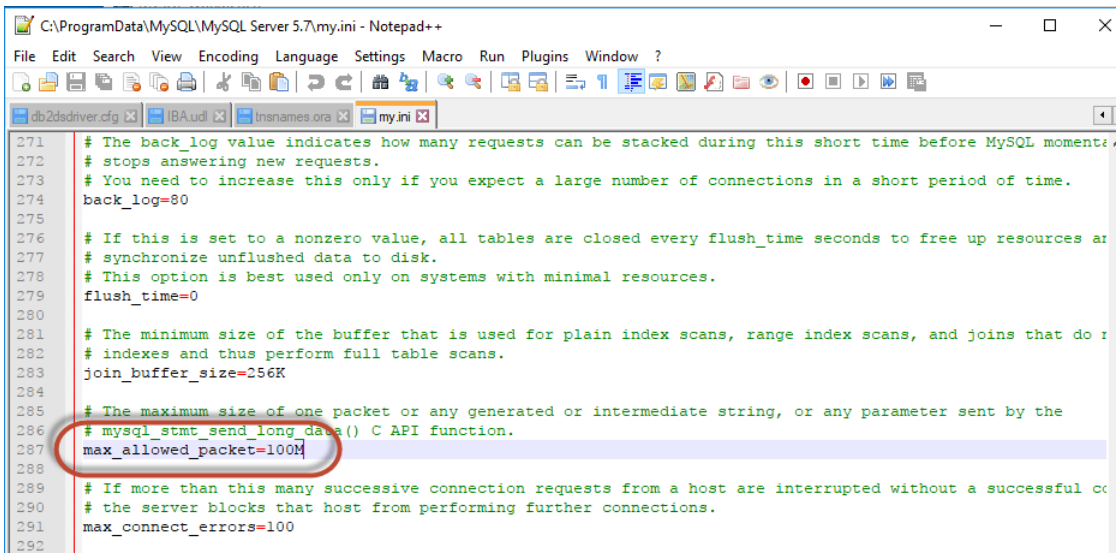


Create a new (<Add…>) or configure an existing (<Configure…>) System DSN.

If BLOB storage is requested the default settings should be adapted as shown in screenshot:

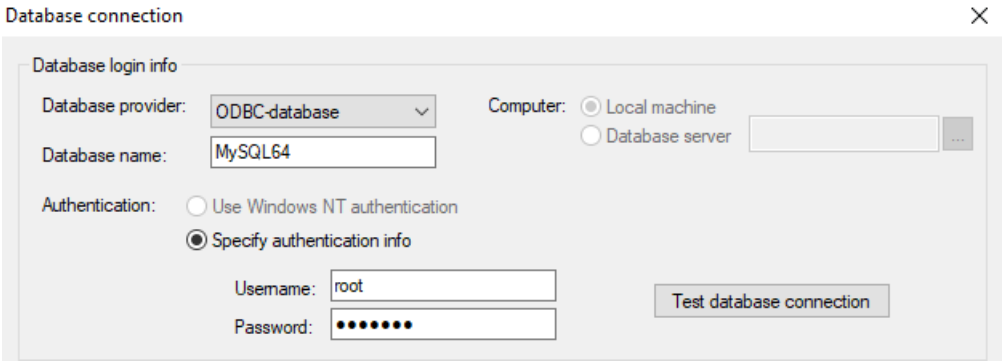


When using BLOBs the init parameter "max_allowed_packet" of MySQL "C:\ProgramData\MySQL\MySQL Server 5.7\ my.ini" also needs to be increased.



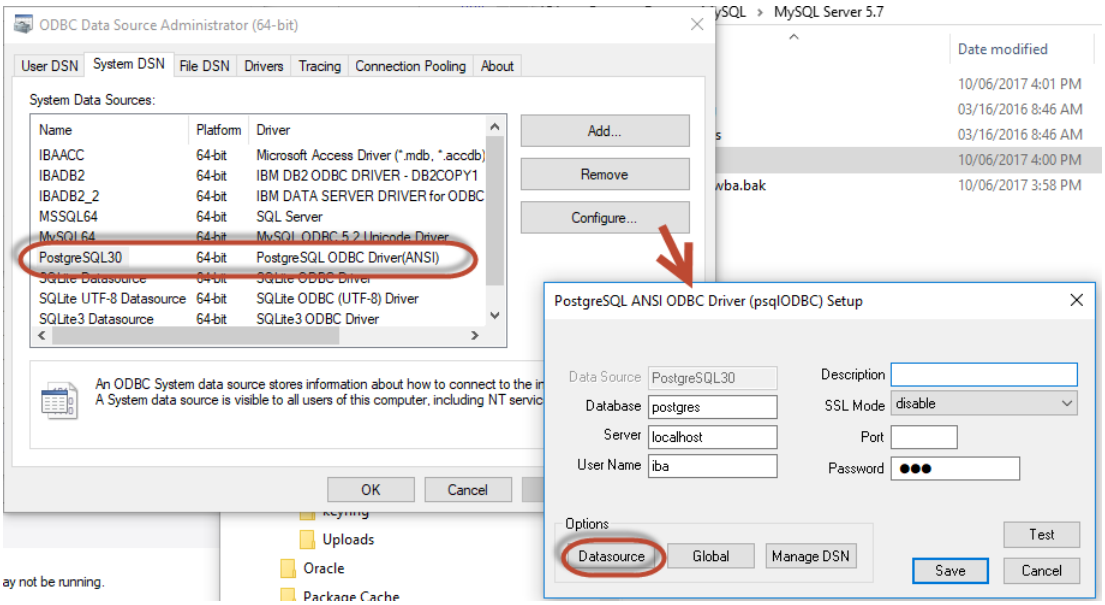
Subsequently the MySQL Server must be restarted.

Select ODBC-database as "Database provider" and use the specified DSN as "Database name":

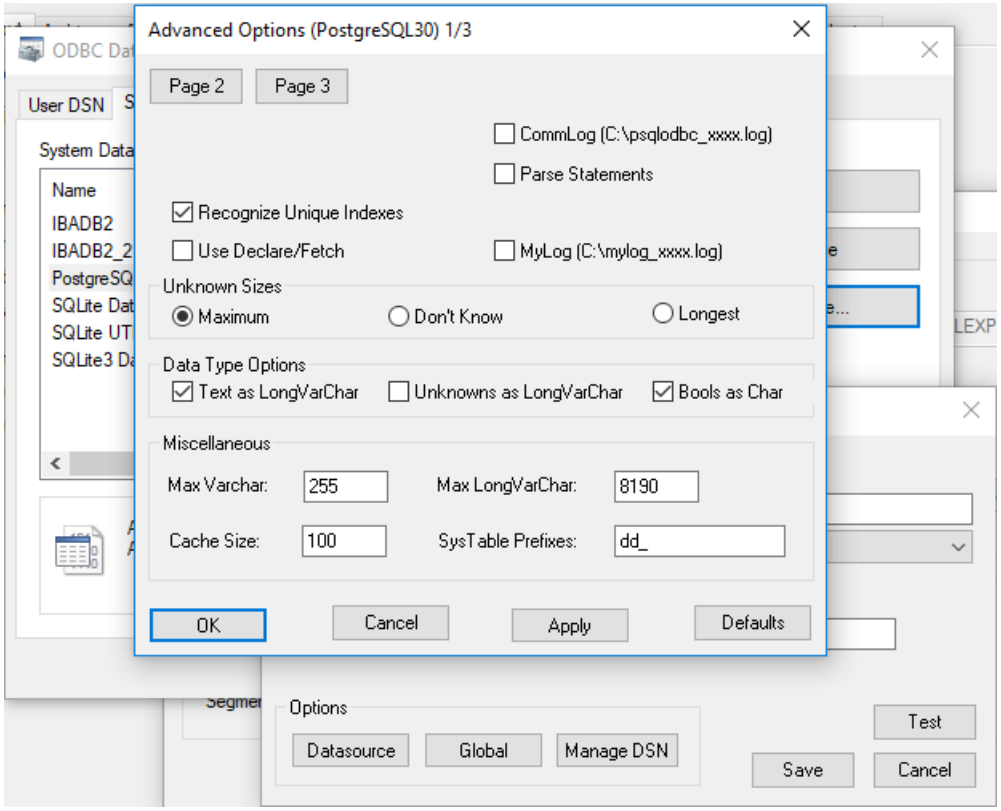


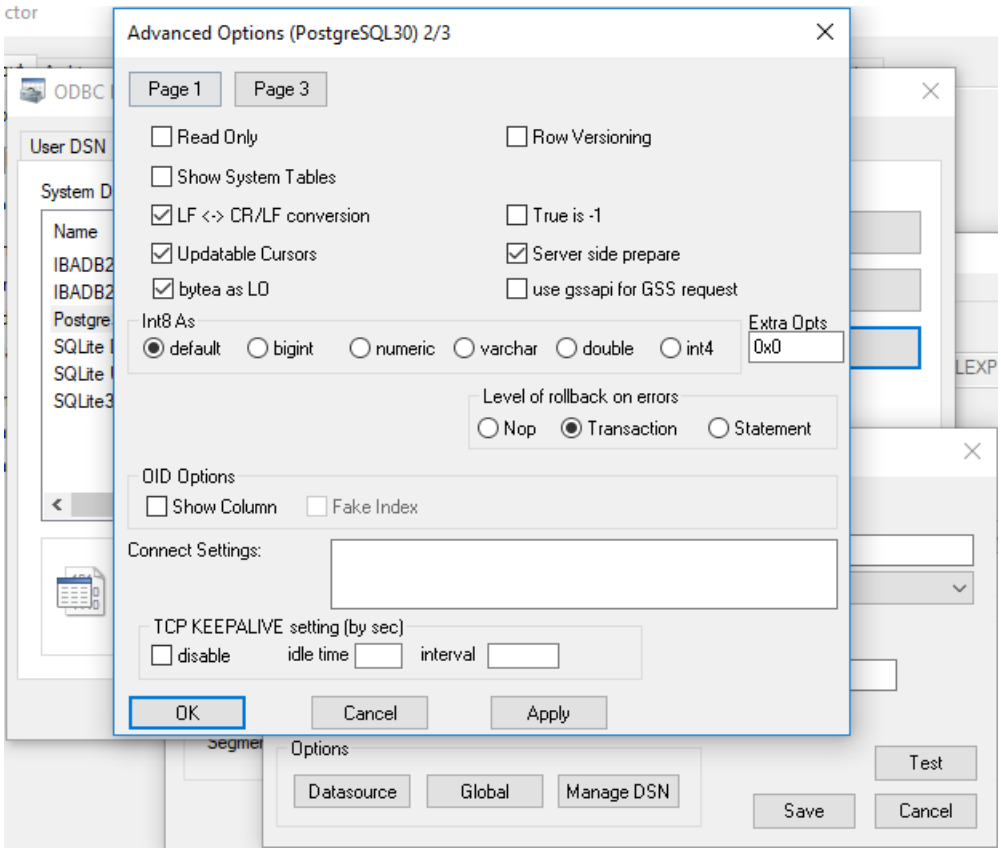
4.1.3 PostgreSQL

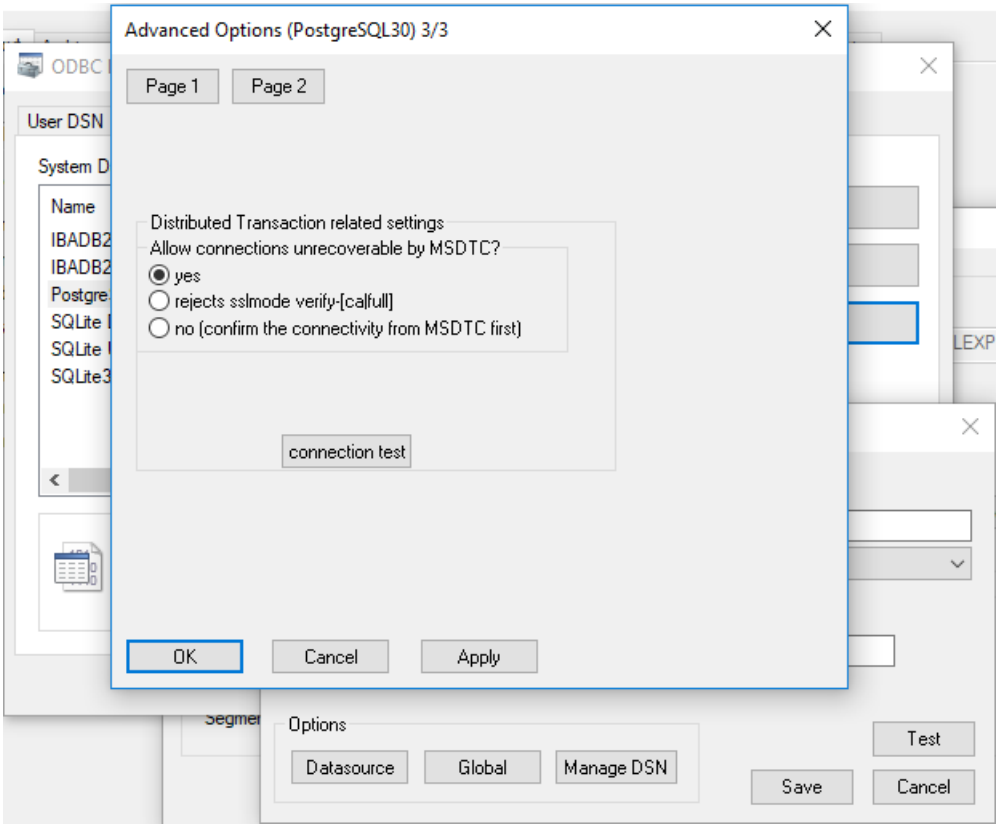
For PostgreSQL use the ODBC driver (32 or 64 bit) and add/configure a System DSN:



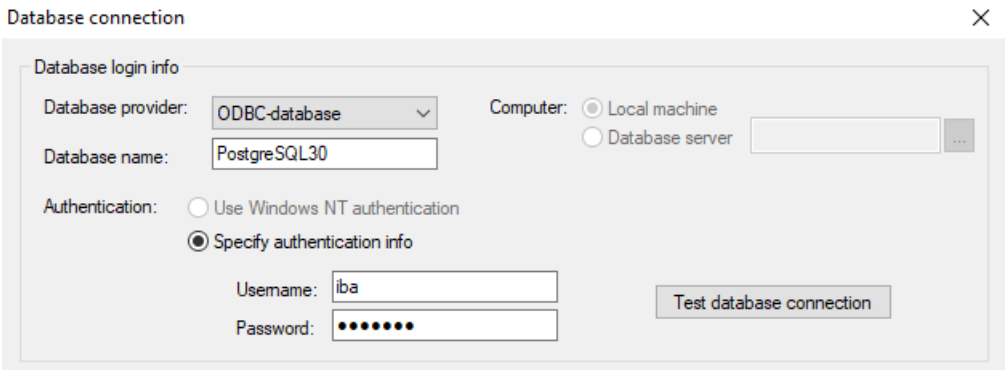
It is important to configure the <Datasource> option on "Page 1 and "Page 2" of "Advanced Options".





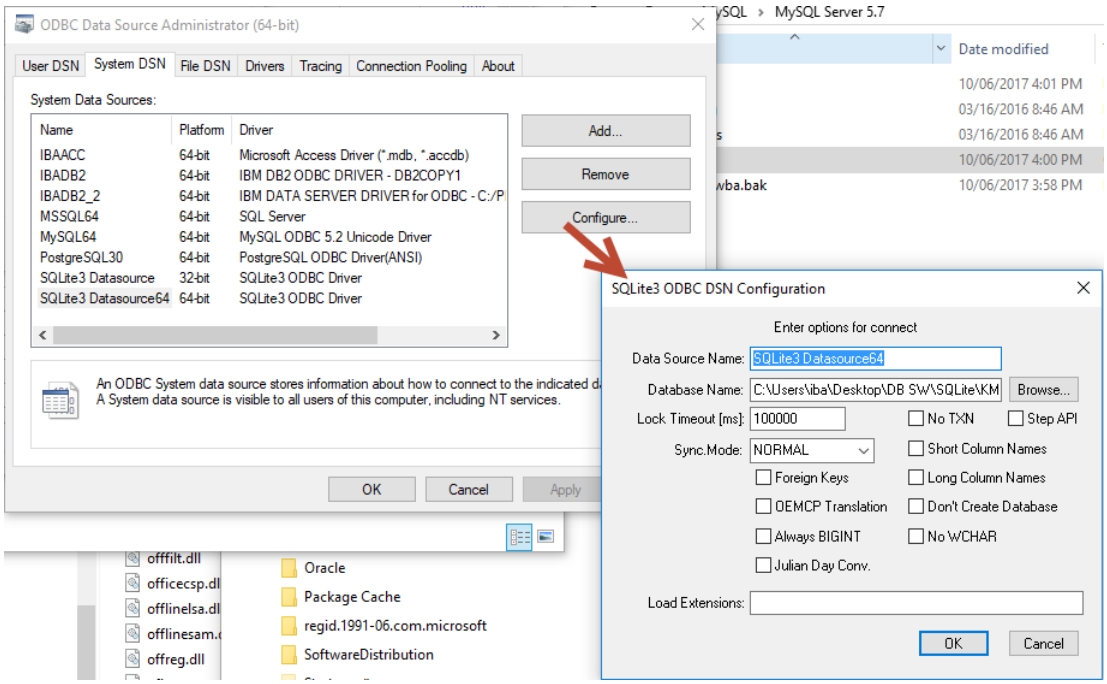


Select ODBC-database as "Database provider" and use the DSN as "Database name":



4.1.4 SQLite

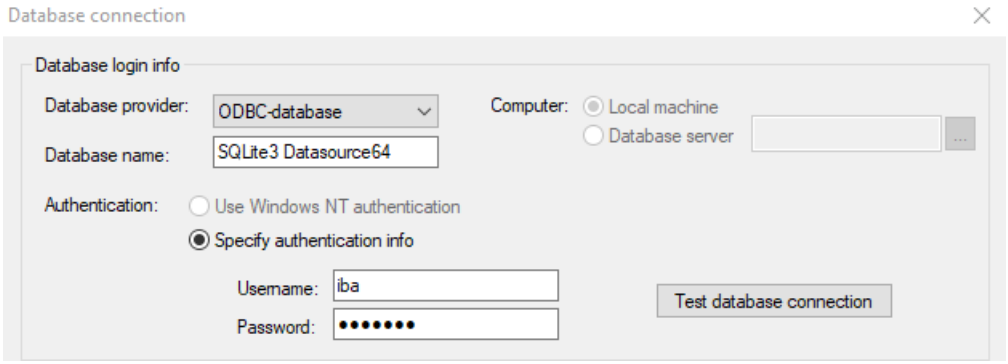
For SQLite use the ODBC driver (32 or 64 bit) and add/configure a System DSN:



Note

Note: BLOBs are not supported for SQLite.

Select ODBC-database as "Database provider" and use the DSN as "Database name".

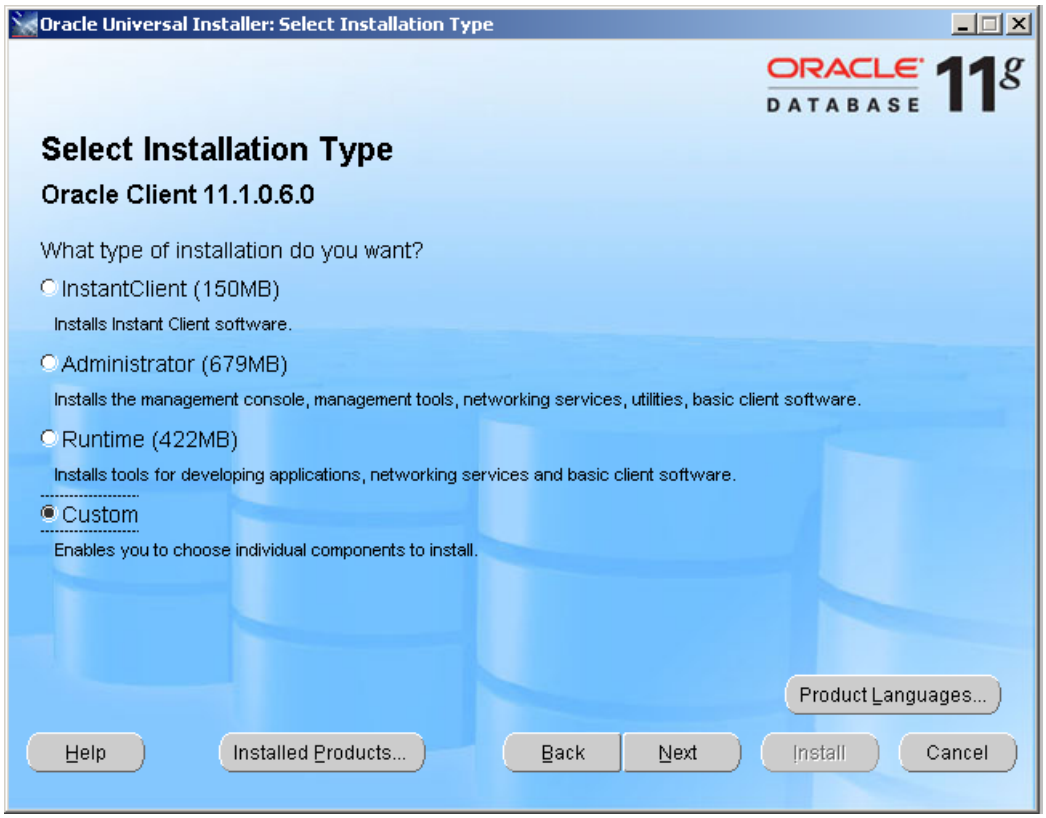


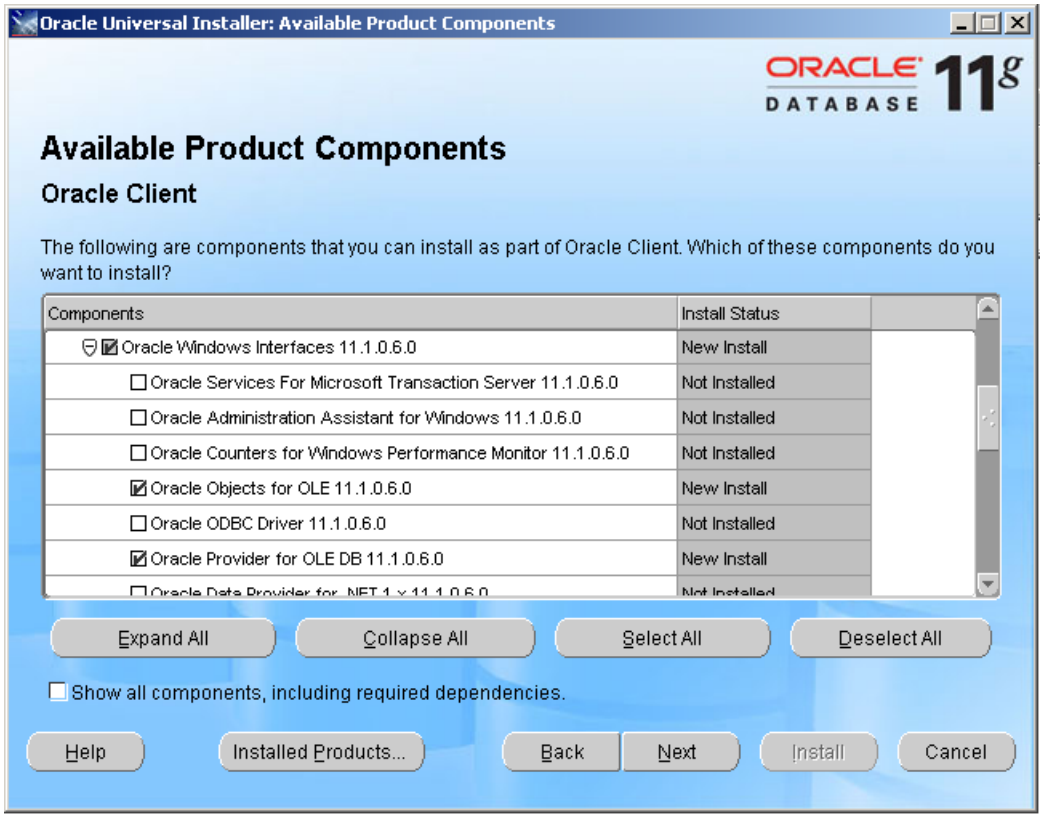
4.1.5 Oracle

If Oracle is used, the Local Net Service Name (TNS-Name), configured in the Oracle client software (see $ORACLE_HOME\NETWORK\ADMIN\tnsnames.ora) must be specified in the "Database name:" field and then instead of OS-identification ("Use Windows NT authentication") the use of an oracle username and password is recommended throughout the database system.

Important information

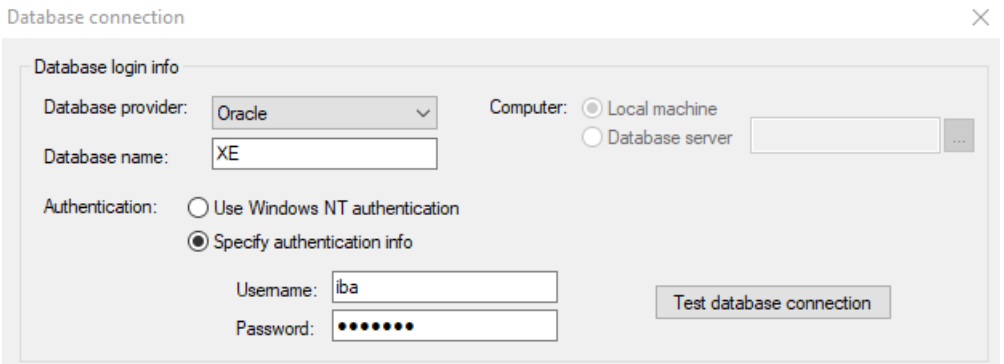
For oracle client installation it is necessary to install the OLE DB option, which is not included in standard installation types.





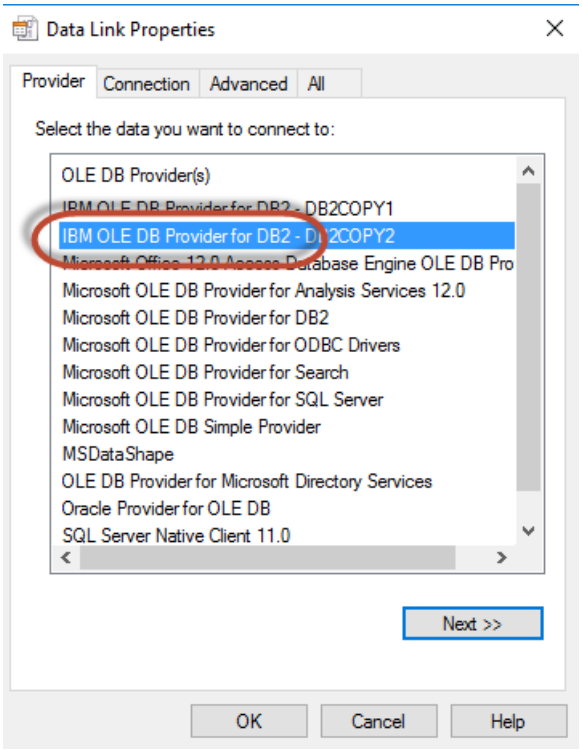
In a centralized Oracle multiuser DB environment it's recommended to create a dedicated user, scheme, storage (table space) and to grant the necessary privileges (create session, create table ...) to this user for the *ibaAnalyzer-DB* load operation. For interactive database queries from different people, a single shared or multiple dedicated database users should be created with adequate privileges (grant select on …). This applies to large installations where the iba data is not to be mixed with data from existing applications. The data base should be configured that the user has read and/or write access thus preventing, for example, the deletion of existing data using the option "create default tables".

Select Oracle as "Database provider" and TNS-Name (XE) as "Database name".

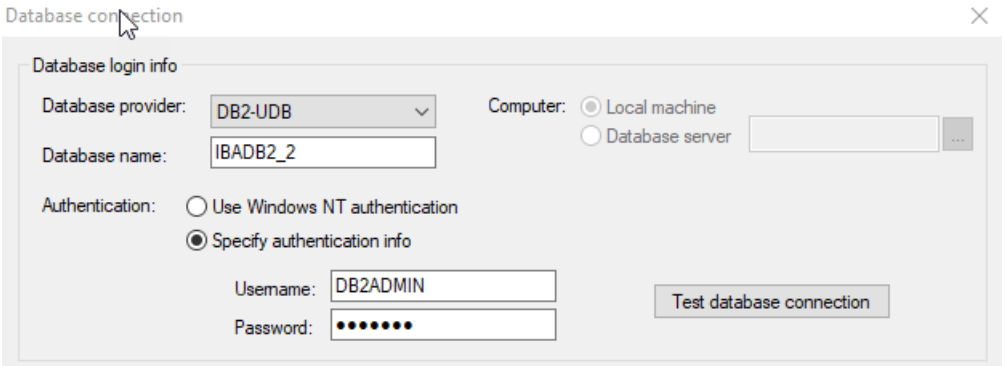


4.1.6 DB2-UDB

For DB2-UDB we recommend using the OLE DB client interface provided via IBM Software:



Select DB2-UDB as "Database provider" and use the database alias name configured in the DB2-UDB client software as "Database name:".

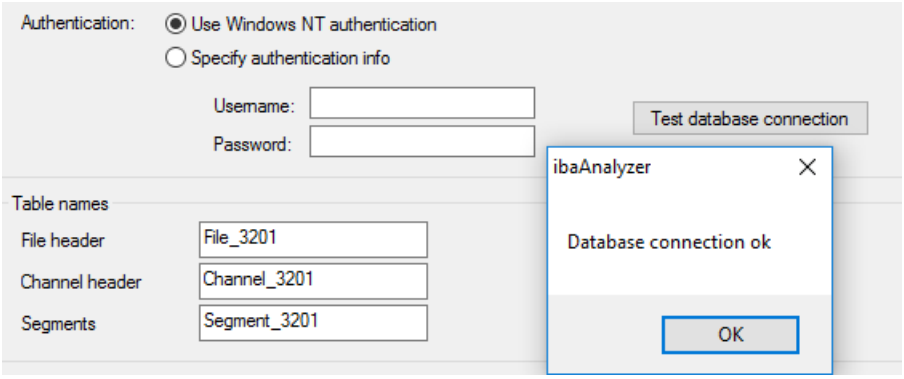


4.2 Testing the database connection in ibaAnalyzer

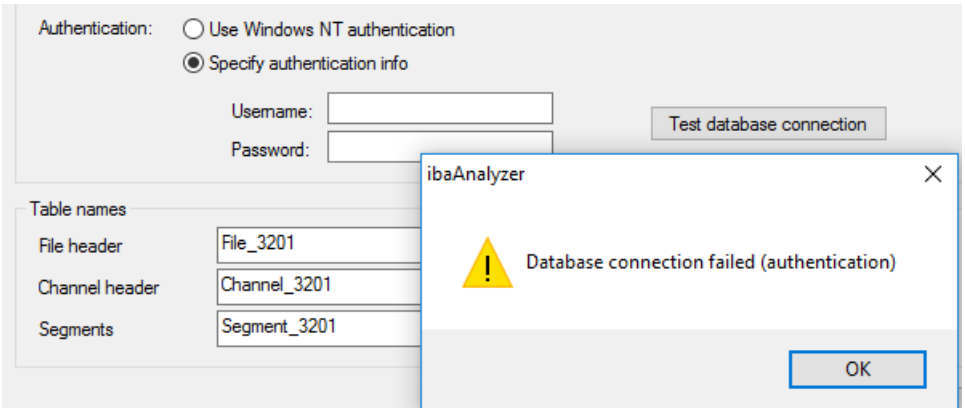
Before closing the window it is recommended to test the database connection by clicking on the corresponding <Test database connection> button.

❑ <Test database connection>

Successful connection



Unsuccessful connection



4.3 Specify database table names

The names of the *File header*, *Channel header* and *Segments* database tables must be entered into the lower part of the *Database connection* window. It is recommended to rename these tables according to the naming conventions of the used database system. The default names "deFile", "deChannel" and "deSegment" can be used for test purposes.

Important information

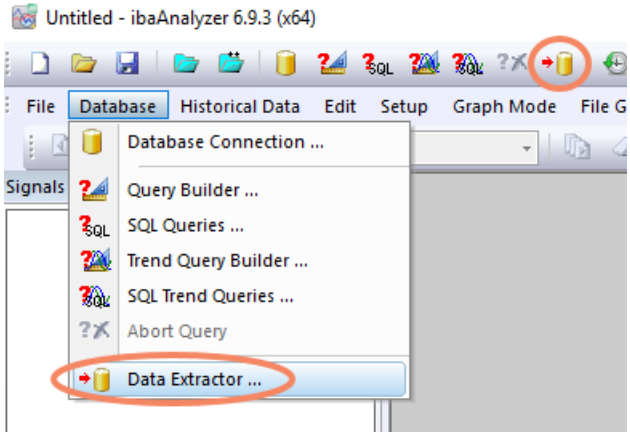
The names must be compatible with the naming restrictions of the employed database system. For example, if Oracle is used and the database support type "MC extractor" is installed, the maximum length for the "Segments" name is 23 characters, since 7 characters are required for the segment table suffixes "_CountT" and "_CountL"

4.4 Save configuration

Close the *Database connection* window by clicking the <OK> button and save the configuration as an ordinary analysis file (*.pdo) of ibaAnalyzer. The latest connection configuration will also be applicable (as are the other preferences) if ibaAnalyzer is opened without an analysis file.

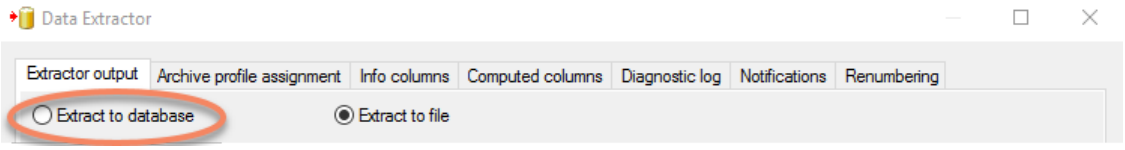
5 Data Extractor

The *Data Extractor* window can be opened in *ibaAnalyzer* by one of the two methods shown in the figure.



The *Data Extractor* has 7 tabs:

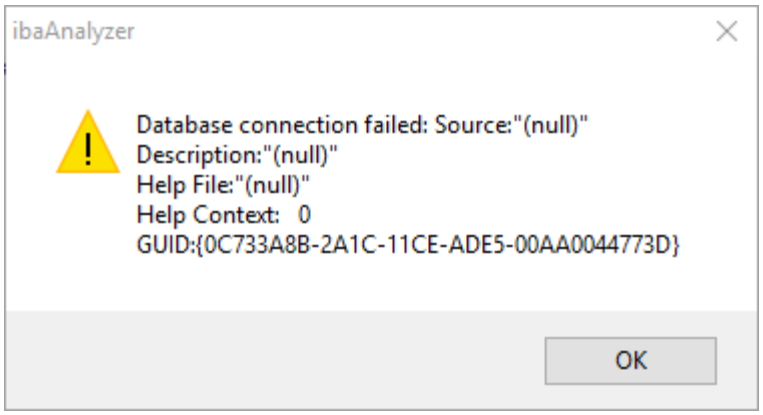
- ❑ Extractor output
- ❑ Archive profile assignment
- ❑ Info columns
- ❑ Computed columns
- ❑ Diagnostic log
- ❑ Notifications
- ❑ Renumbering



❑ Extract to database

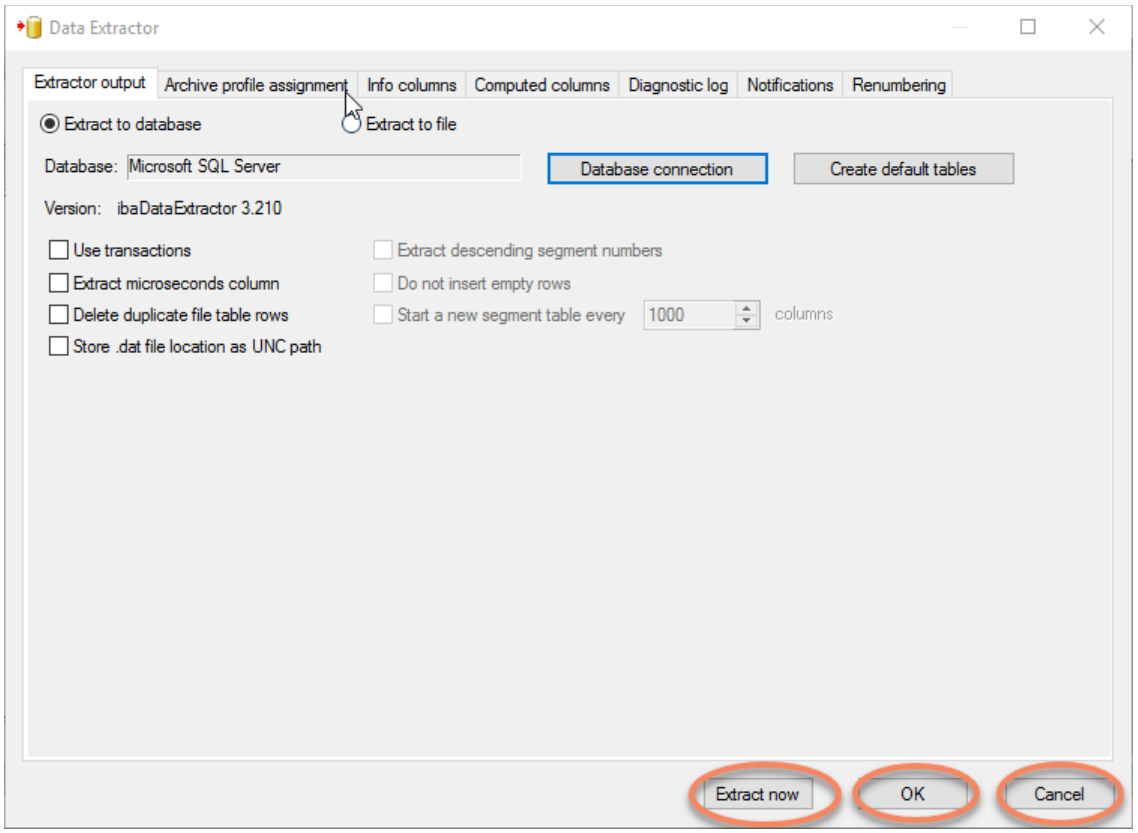
The "Extractor output" tab is displayed automatically in accordance with the selected database type.

After selecting *Extract to database ibaAnalyzer-DB* automatically tries to connect to the database that is configured. If no database connection was configured before (connect timeout) an error message will be triggered.



Click <OK> and select <Database connection> (see chapter "*Configuring the database connection*, page 10")

Each *Data Extractor* tab contains the following buttons:

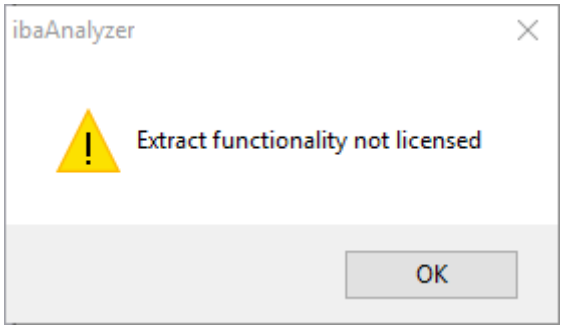


❑ <Extract now>
The database tables will be filled based on all the current settings in the „Data Extractor" window.

❑ <OK>
All current settings in the "Data Extractor" window are saved and window is closed.

❑ <Cancel>
All settings changed after the "Data Extractor" window was opened are discarded and window is closed.

The following message will appear if the license is not recognized.

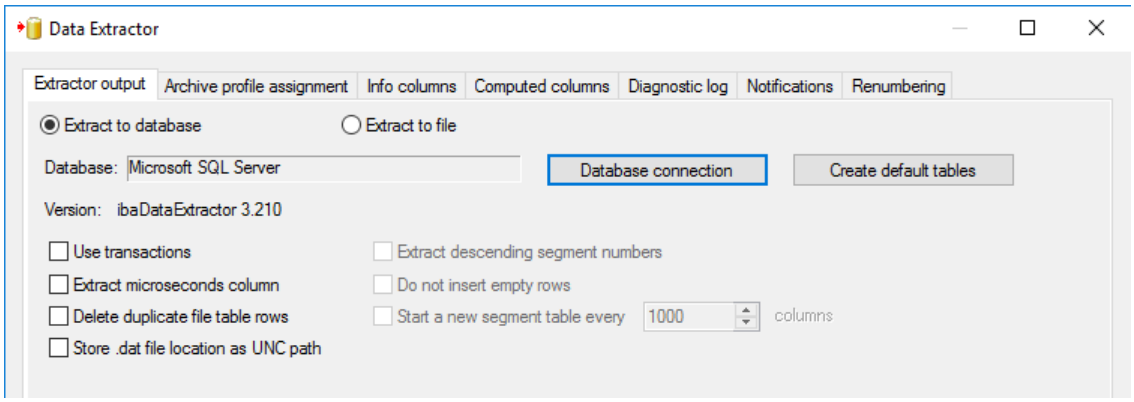


5.1 Extractor output

The "Extractor output" window is opened automatically when "Data Extractor" is selected.

5.1.1 Option Extractor Database Library (Standard)

If ibaAnalyzer is installed with the database support type "install the Extractor database library" the following window is displayed.



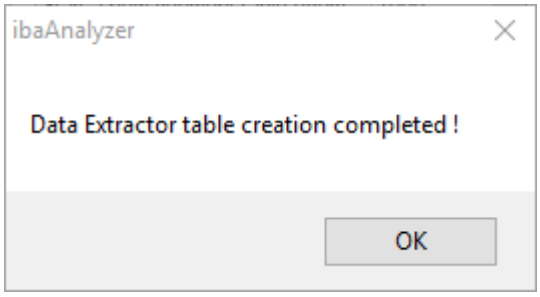
❑ "Database:"
This displays the connected database type.

❑ <Database connection>
This opens the "Database connection" window (see chapter *Configuring the database connection*, page 10).

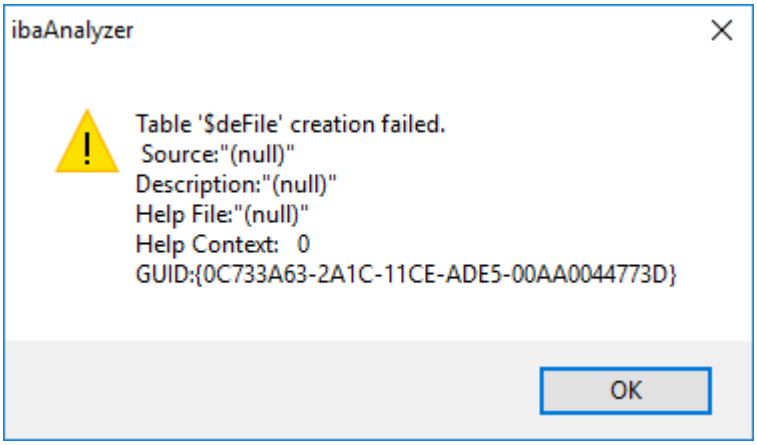
❑ <Create default tables>
This creates tables in the database with the names specified in the "Database connection" window. See chapter *Specify database table names*, page 24.

When you click this button, *ibaAnalyzer-DB* checks whether the database already contains tables with the same names. If there is no conflict with existing objects, tables, indexes and constraints the tables are created. If tables with the specified names already exist, *ibaAnalyzer-DB* will ask if these should be replaced.

Successful creation is indicated by:



Unsuccessful creation triggers an error message, e.g.

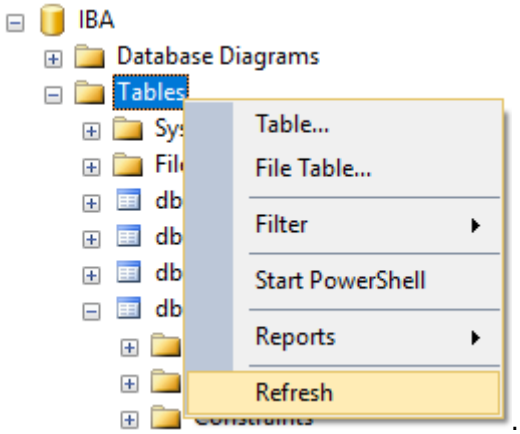


Important information

The naming conventions of the connected database must be adhered to, e.g. avoid blanks, special characters, etc.

Important information

Use the "Tables Refresh" command in the database to ensure that the transaction results are displayed.



Important information

Ensure user privileges. In multiuser environments only the application administrator should be the object owner. Other users should have different database logins with appropriate restrictions (e. g. only SELECT privileges).

In large, centralized, multiuser DB environments (Oracle, DB2-UDB) the standard DDL-commands without storage options and simple indices might not be sufficient and the creation of database tables by <create default table> using *ibaAnalyzer* will not be permitted. On such systems, the database administrator can create the default objects in a test environment, and by reverse DDL-script generators he can produce and then customize his own DDL-scripts. Depending on the integration scenario it might also be necessary to create further INDEXES on "Technostring columns".

❑ Use transactions

All SQL INSERT operations take place within one transaction. If any operation within the transaction fails, the whole transaction is aborted.

❑ Extract microseconds column

This creates a "_TimeStampMicroSecs" column, which permits correct timestamp referencing on a microsecond basis. See also the item "TimeStampMicroSecs".

❑ Delete duplicate file rows

This ensures that any previously saved files having the same name (see column "_FileName") as the current file to be extracted are deleted.

Before extraction:

	_FileId	_TimeStamp	_FileName	_FileType	_Complete	_ErrorOnExtract	Module_name_2
1	-1243620028	2017-01-30 00:11:55.000	D:\IBA\dat_files\dat-training\pda_training021.dat	real	1	0	MRG_N
2	-1240416044	2017-01-30 00:11:55.000	D:\IBA\dat_files\dat-training\pda_training021.dat	real	1	0	MRG_N

After next extraction:

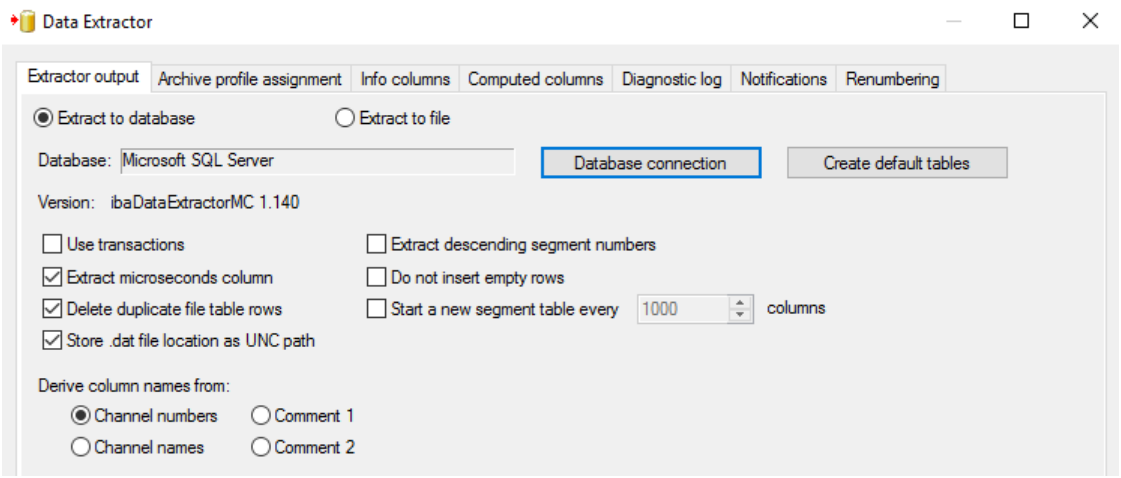
	_FileId	_TimeStamp	_FileName	_FileType	_Complete	_ErrorOnExtract	Module_name_2
1	-1240325615	2017-01-30 00:11:55.000	D:\IBA\dat_files\dat-training\pda_training021.dat	real	1	0	MRG_N

❑ Store .dat file location as UNC path

This is necessary if .dat files are not stored locally

5.1.2 Option MC Extractor database library

If *ibaAnalyzer* is installed with the database support type "install the MC Extractor database library" the following window is displayed.



❑ Database connection
See chapter *Option Extractor Database Library (Standard)*, page 29

❑ Create default tables
See chapter *Option Extractor Database Library (Standard)*, page 29

❑ Use transactions
See chapter *Option Extractor Database Library (Standard)*, page 29

❑ Extract microseconds column
See chapter *Option Extractor Database Library (Standard)*, page 29

❑ Delete duplicate file rows
See chapter *Option Extractor Database Library (Standard)*, page 29

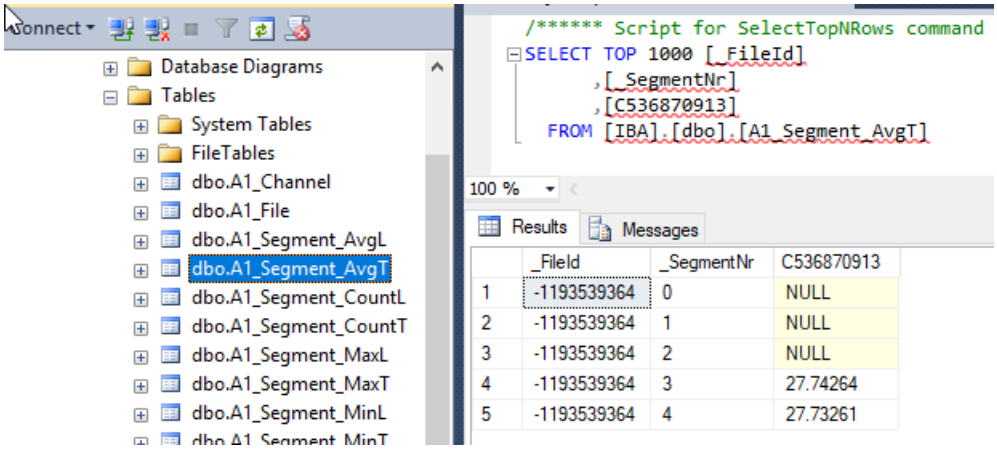
❑ Store .dat file location as UNC path
See chapter *Option Extractor Database Library (Standard)*, page 29

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

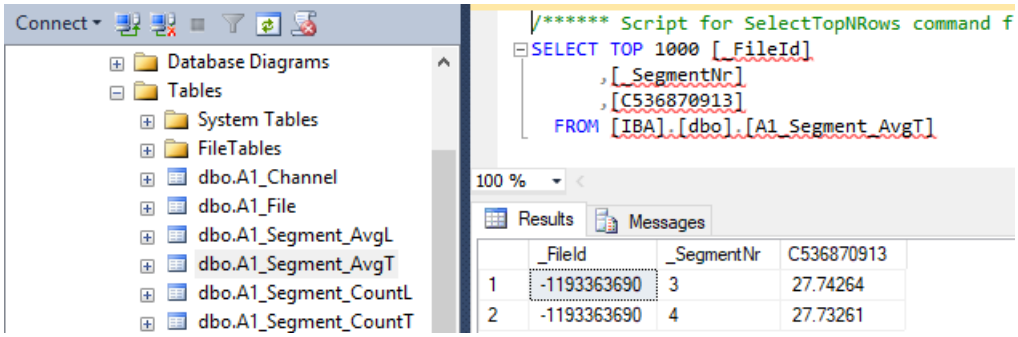
	_FileId	_SegmentNr	_ReverseSegmentNr	C33155	C33157
1	-374350715	0	5999	217.2715	190.2763
2	-374350715	1	5998	238.0183	196.4841
3	-374350715	2	5997	219.696	182.4169
4	-374350715	3	5996	226.9127	182.002
5	-374350715	4	5995	241.7527	185.6307
6	-374350715	5	5994	223.1701	192.2534

❑ Do not insert empty rows
Segments containing NULL are ignored during extraction

Deactivated:

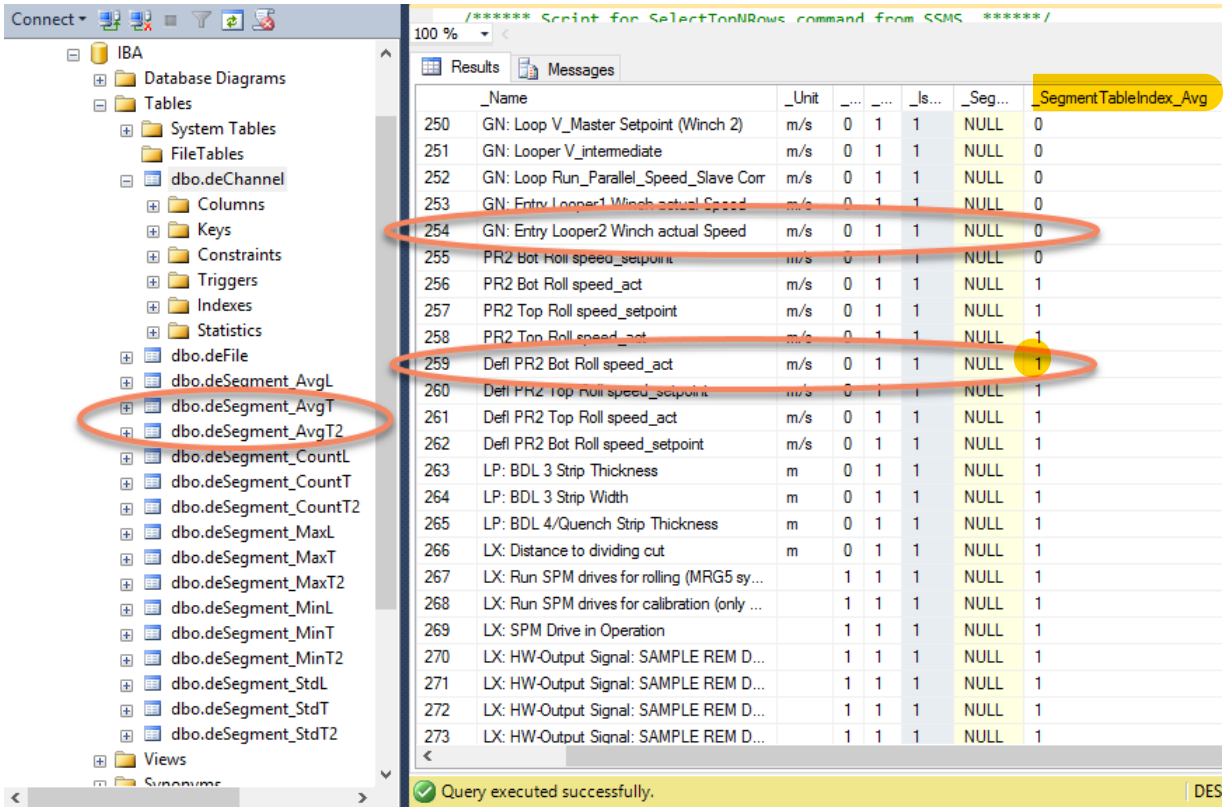


Activated:



❑ Start a new segment table every <N> columns

In MC-format, database restrictions may require further segment tables. For example, in Oracle the maximum number of columns is limited by 1,000. If more signals are to be extracted, it is necessary to create further segment tables. If the specified number (N >99) of columns is reached, the extractor automatically creates new segment tables. The channel table is also extended by this option to 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" and "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
266	LX: Distance to dividing cut	m	0	1	1	NULL	1
267	LX: Run SPM drives for rolling (MRG5 sy...		1	1	1	NULL	1
268	LX: Run SPM drives for calibration (only ...		1	1	1	NULL	1
269	LX: SPM Drive in Operation		1	1	1	NULL	1
270	LX: HW-Output Signal: SAMPLE REM D...		1	1	1	NULL	1
271	LX: HW-Output Signal: SAMPLE REM D...		1	1	1	NULL	1
272	LX: HW-Output Signal: SAMPLE REM D...		1	1	1	NULL	1
273	LX: HW-Output Signal: SAMPLE REM D...		1	1	1	NULL	1

Important information

"Create default tables" does not delete additional segment tables.

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 n > 0 represent a reference to segment tables with the number n+1 at the end of their names.

Important information

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

❑ Derive column names from

When using the ibaDataExtractorMC (multi column) library, the column names in the segment table can be derived either from the <Channel number>, the <Channel name> or the first or second comment of the extracted signals (see: ibaPDA manual chapter: IO manager).

Important information

If the selection is altered additional channels will be created and the previous channels will remain. No channel will be deleted!

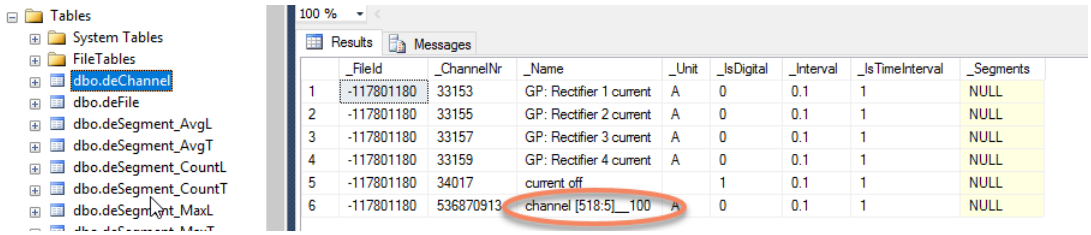
If <Comment 1> is selected and a signal does not have a first comment, the channel name is used instead. If <Comment 2> is selected for a signal where "Comment 2" is not available, "Comment 1" is used, if that is also missing, the channel name is used.

Important information

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

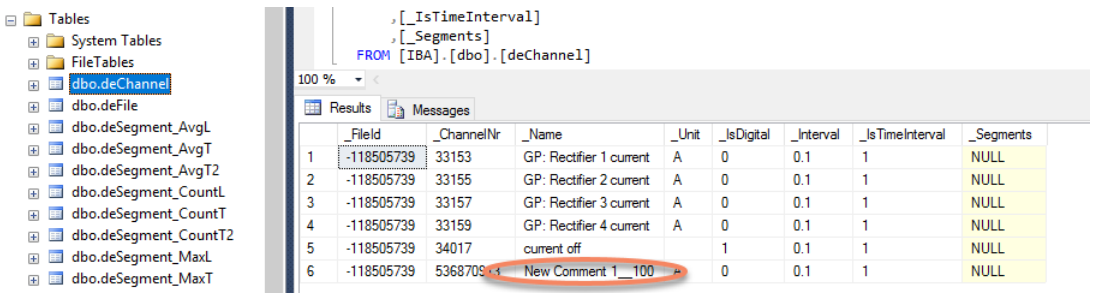
	Show	SignalName	Expression	Comment 1
1	☑	channel [518:5]	[18:7]	New Comment 1

"Channel number" or "Channel name" selected:



	_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

"Comments 1 or 2" selected:



	_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

When creating the default tables, a message will appear if tables with the specified names already exist. If the message is confirmed, all data stored in those tables will be lost ("drop table"). This is a quick way of deleting existing database tables when reinitializing.

Important information

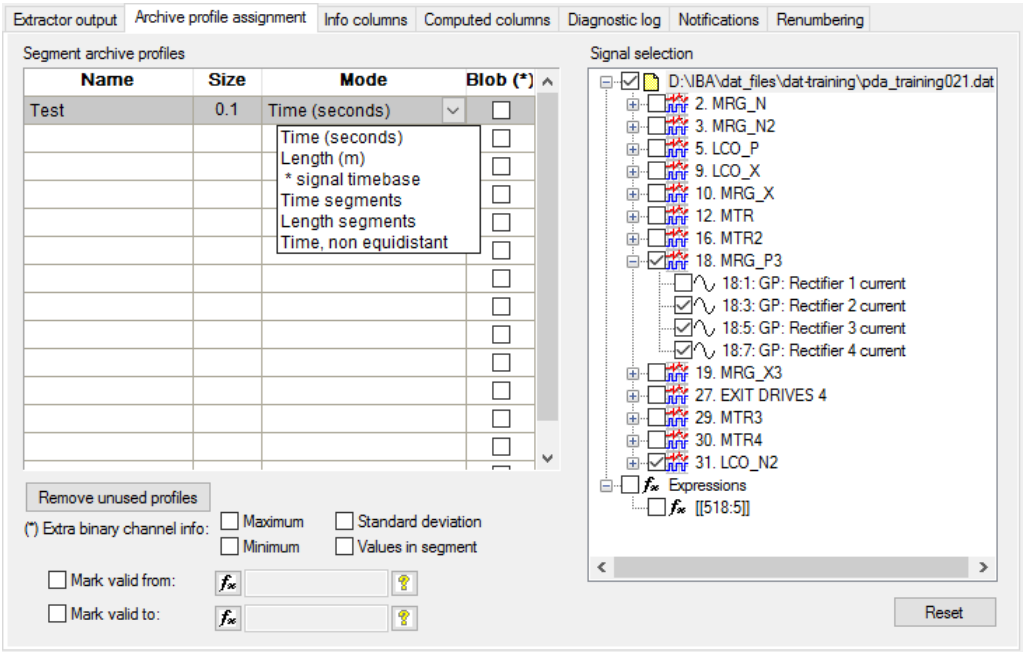
„Create default tables" does not delete any tables which were created when the option „Start a new segment table every <N> columns" was selected. These must be deleted manually in the database.

5.2 Archive profile assignment

In order to create an archive profile at least one data file must be opened for extraction.

Select the tab *Archive profile assignment*.

An archive profile defines the signals together with the required sampling cycle (time or length based) which will be included in the extracted data. More than one profile can be specified. However a signal can only be assigned to one profile.



❑ Column "Blob"

Data is stored as a Binary Large OBject (BLOB). For details see chapter *Using BLOBs*, page 59 .

Note

If the BLOB-option is used, the statistical values are only accessible by a user defined de-compress program - not by ibaAnalyzer!

Note

If the BLOB-option is used, the option „Values in segment" is unavailable.

❑ Column "Mode"

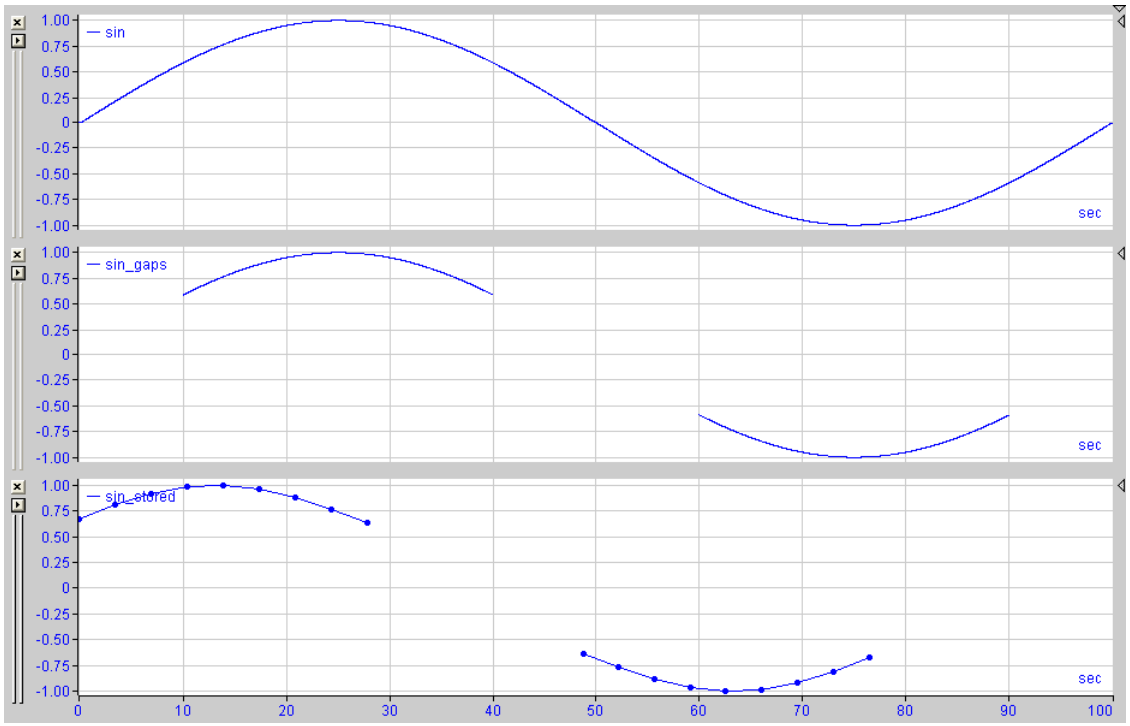
Use the drop down list to select a sampling mode in an empty cell in the "Mode" column of the "Segment archive profiles" window.

❑ Column "Size"

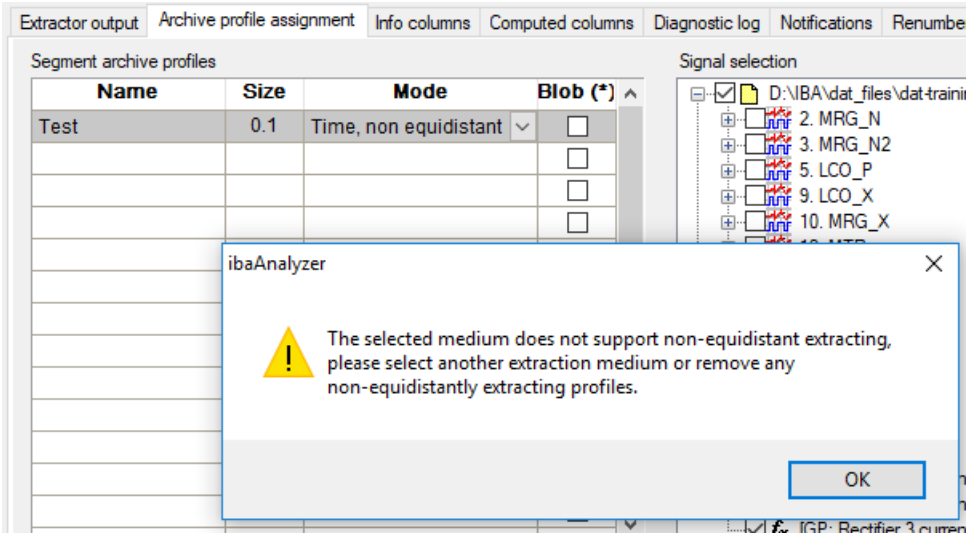
Select the adjacent cell in "Size" column and enter a multiplier to determine the sampling cycle for the extracted data (e.g. 10 s = "Size (10)" x "Mode (Time(Seconds))".

For the "Time" and "Length" modes, it's possible to enter a "Size" < 1 in order to obtain fractions of the "Mode" unit. Fractions of the mode "*signal time base", of course, make no sense, e. g. a temperature that was measured every minute need not be extracted every 100 ms. If the recording signal time base is 1 ms, the sample 100 ms profile could

also be defined by "Size" = 100 and "Mode" = "* signal time base".
By using the modes "Time segments" or "Length segments" 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 for the whole data channel is stored.
If parts of the source signal are invalid, then extracted segments containing these will be empty. If the empty segments are at the beginning or end of the signal, they will not be transferred to the extracted data. If the empty segments lie between valid segments, then the extracted data will contain corresponding gaps of invalid values.



The mode "Time, non-equidistant" is not supported and leads to the following error message.



❑ Column "Name"
Select the adjacent cell in the "Name" column and enter a profile name.

It is advisable to use a name which reflects the chosen sampling period for the extracted file (e.g. "10 s" = Size (10) * Mode (Time (Seconds)).

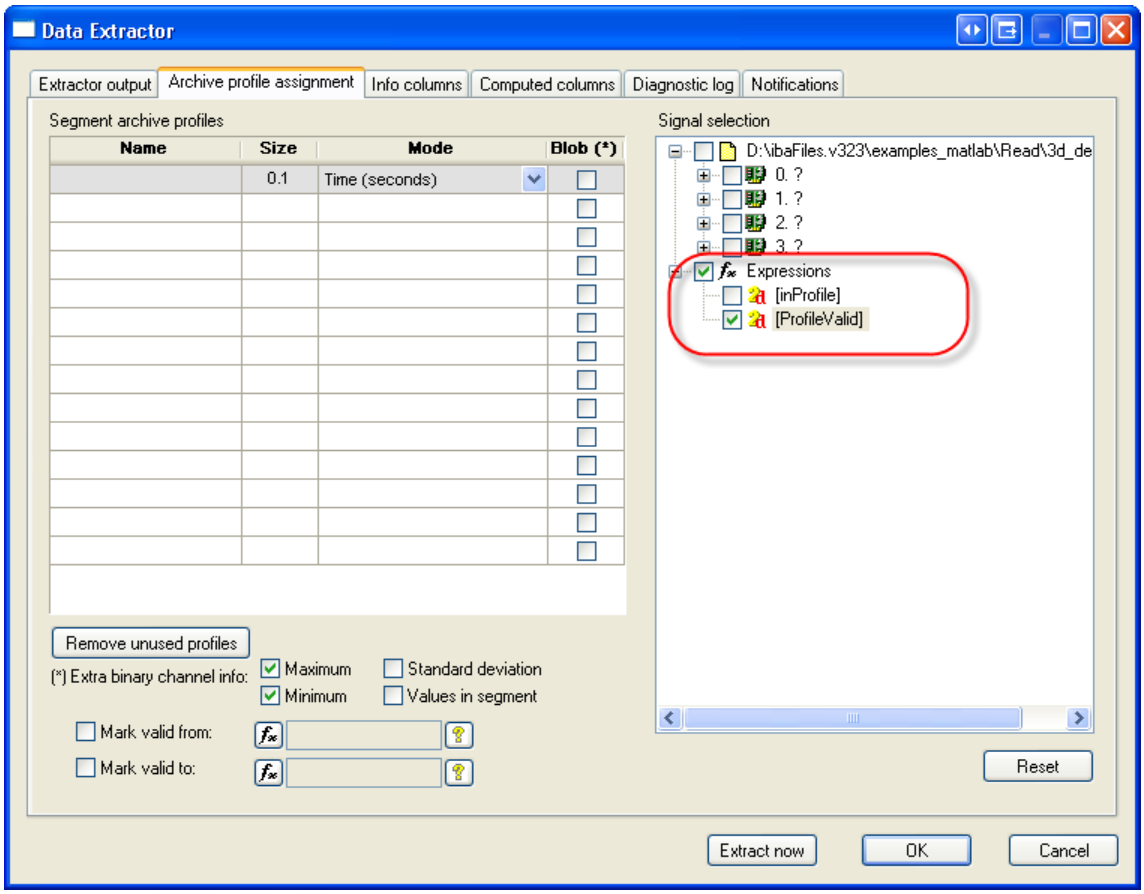
❑ Signal selection

Signals must be assigned to the created profiles.

Select any number of the signals to be included in the extracted data using the check boxes in the Signal selection window. The extracted signal values are the averages of the source signal values within the selected sample cycles.

When checking a module node in the "closed" tree view all signals belonging to that module will be checked, or unchecked. Before assigning individual signals the module tree has to be opened by clicking on the + symbol. The checked signals are always assigned to the marked profile (gray background color). When changing the profile, the checkmarks in the checkboxes disappear.

Multidimensional expressions (logicals, expressions on logicals and the new vectors from a data file) can be extracted to and queried back from the database. In the database channel table a new column named "_LogicalId" ("i_LogicalId" for Oracle and IBM DB2 databases) will be created to indicate which sub-channels belong to the same multidimensional channel.



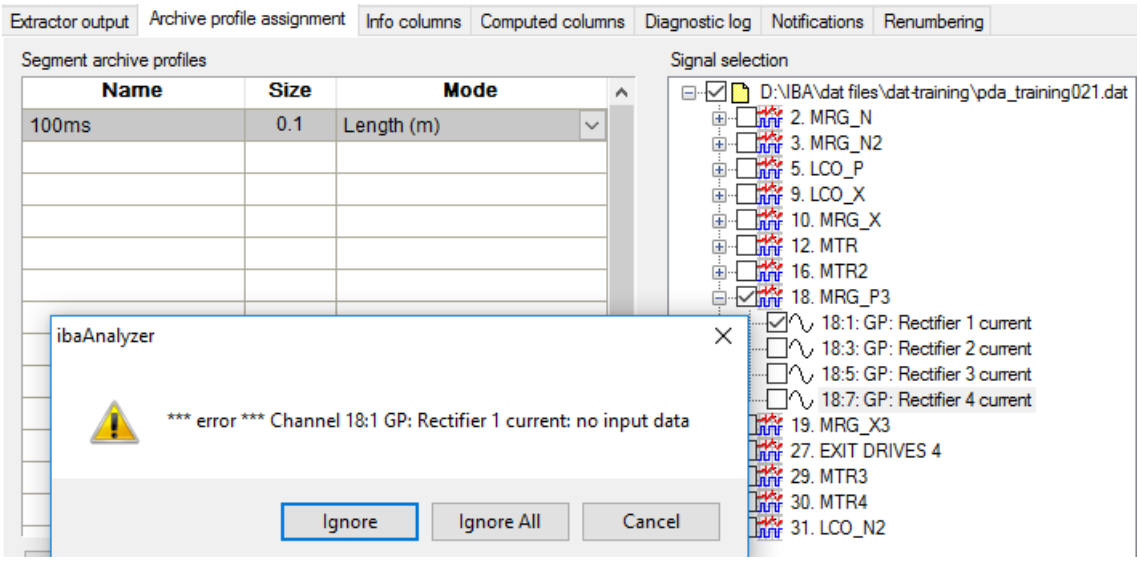
Important information

A signal can always be assigned to only 1 profile.

Important information

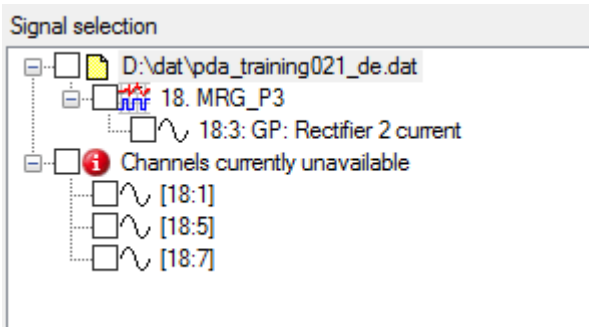
Assigning a length mode to a time based signal or a time mode to a length based signal

will produce an error on extraction.



Note

If the archive profile specifies signals which are not present in the existing data file, these will be listed under "Channels currently unavailable".



The options for <Maximum>, <Minimum>, <Standard deviation> and <Values in segment> are only available for the MC option (see chapter "MC-format segment tables"). In the standard option, these values are automatically extracted to the segment table.

❑ Maximum, Minimum or Standard deviation

Select any of these options. The options generate sub-channels which are respectively the maximum, minimum, or standard deviations of the source signal values within the selected sample cycles.

Note

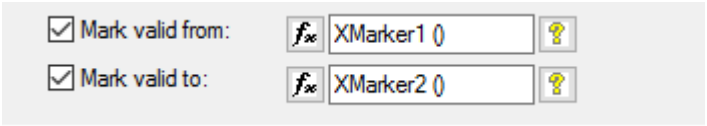
Obviously if the selected profile sample cycle is the same as that of the source signals, these values are meaningless.

❑ Values in segment

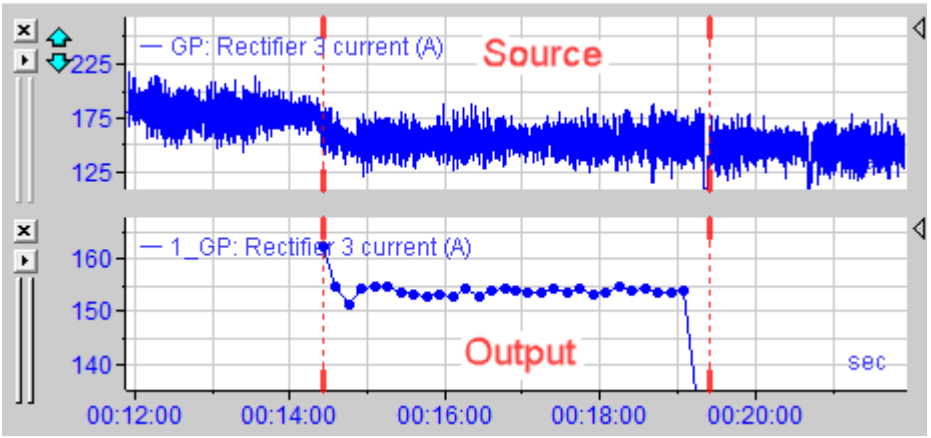
The column "_ValuesInSegment" is created in the database and contains the number of data points compressed into each segment stored in the database.

❑ Mark value from, Mark value to

Use "Mark value from" and/or "Mark value to" to define a beginning and end of the extracted profile range. If either of these is not defined, the corresponding end of the extracted profile range will be the same as that of the input signal range.



If the given expressions cannot be evaluated, an error message will occur during extraction.



Important information

You can use expressions to calculate the range from your data.

- <Remove unused profiles>

Profiles which have no signals assigned to them are deleted.

- <Reset>

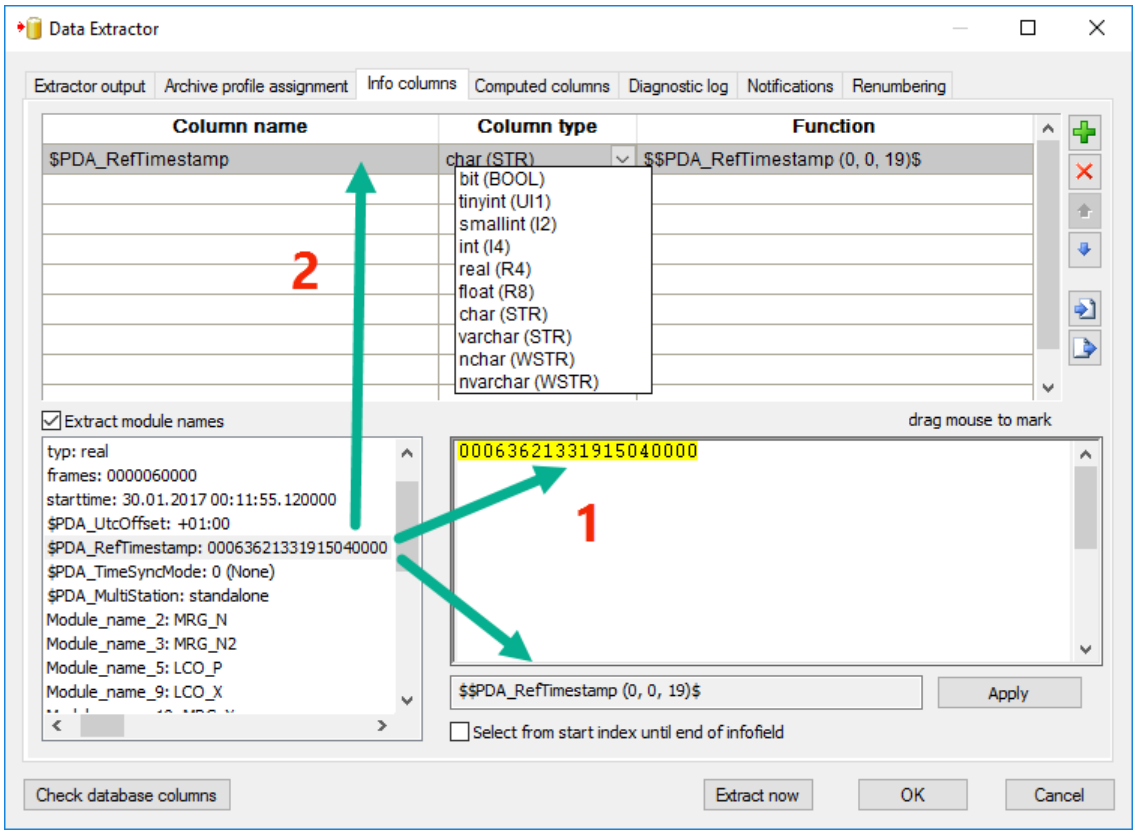
All signals assigned are removed from the selected profile.

5.3 Info columns

The *Info columns* tab permits the extraction of information from the "Info" part (e. g. techno string columns) of the source file. This data will be found in the "deFile" table in the database.

5.3.1 Info field assignment

There are two procedures for selecting an info field.



Click on the source field. The field information will be transferred to the processing field.

Double click on the source field. The column line will be filled in addition to the processing field. The "Column type" can be selected from the drop-down list, which contains the available data types on the currently active database system. The "Column name" can be renamed if required. Restrictions for database column names have to be considered. The range of characters within the "Function" can be altered in the processing window.

Note

If there is no active database connection, only "CHAR" will be displayed.

Caution

The names might not be compatible with the naming restrictions of the used database.

Avoid blanks, hyphens and special characters, etc. and check the permissible length!

When marking the characters in the technostring area, be sure that the correct row in the table above has been selected. The last marked excerpt before leaving the row or closing the window is valid and will be stored.

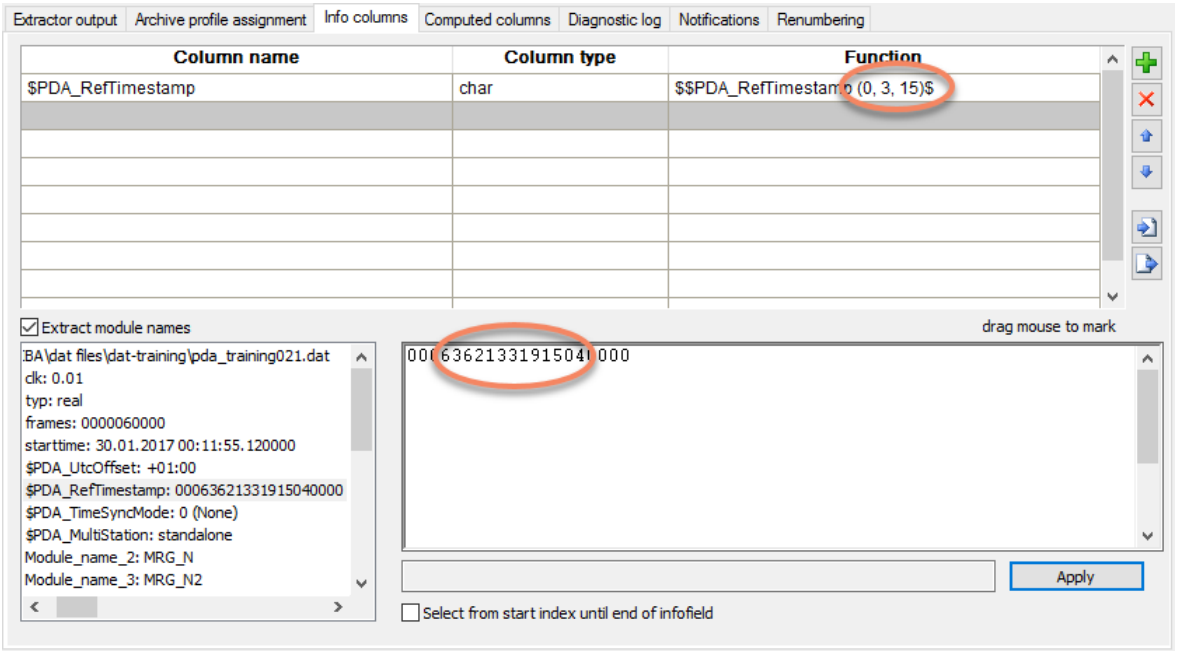
❑ Select group

It is also possible to select a group of characters from the contents of the info field.

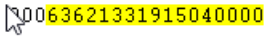
Click on the first of the required characters, drag the mouse to the last required character and release.

Click on <Apply>

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



❑ Select from start index to end of info field



All the characters from the marked starting point to the end of the info field are selected.

❑ Extract module names

If the option is enabled (default), the module names are extracted to the file table.

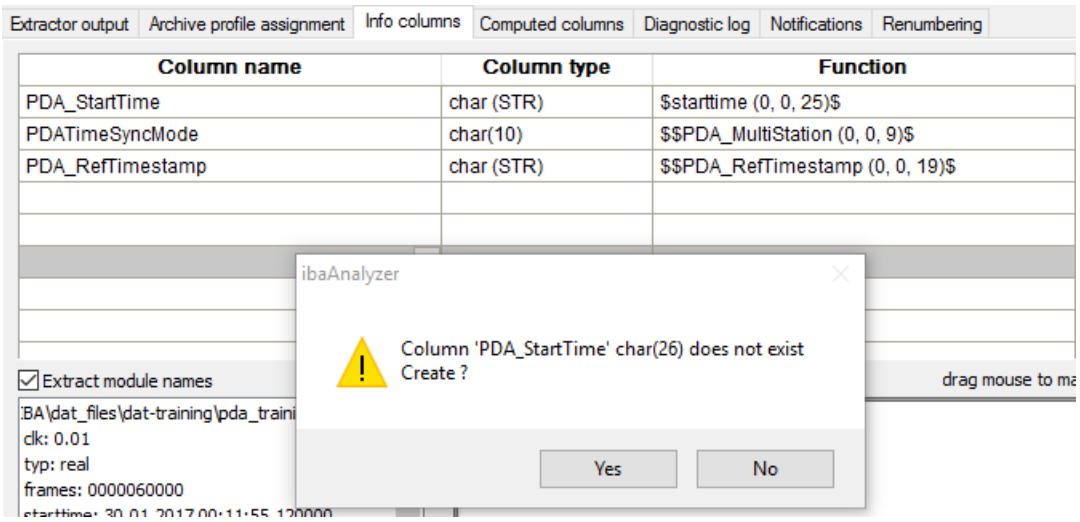
MinimumRectifier2current	_TimeStampMicroSecs	Module_name_2	Module_name_3	Module_name_5	Module_name_9	Module_name_10
103.815	120000	MRG_N	MRG_N2	LCO_P	LCO_X	MRG_X

Note

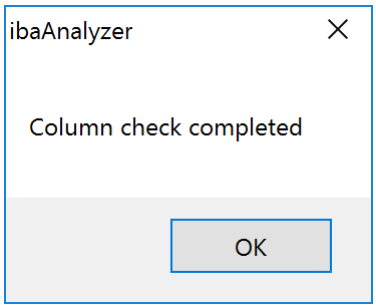
If there is no need for module names after the extraction, the number of columns in the file table can be reduced by deselecting this option.

❑ <Check database columns>

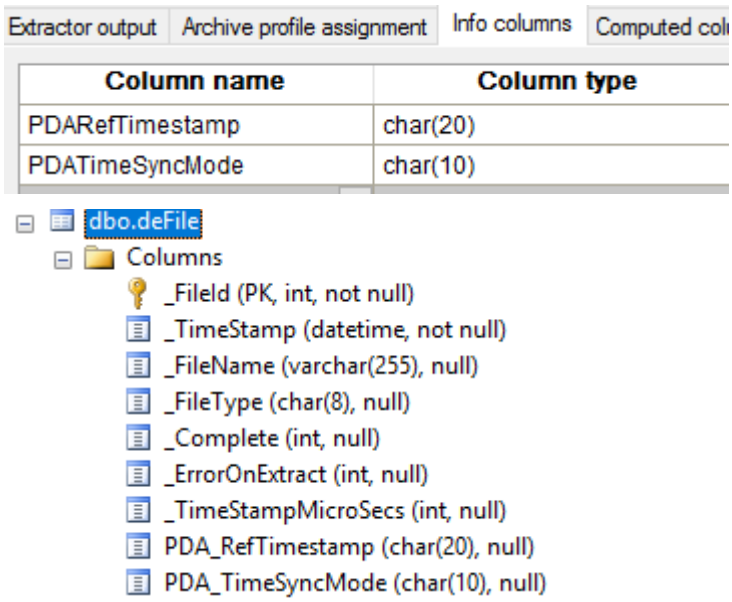
Once the "Info columns" and the option "Extract module names" have been configured, these must be synchronized with the database. If the requested column does not exist, its creation must be confirmed with <OK>. The columns for module names will be opened without the need for confirmation.



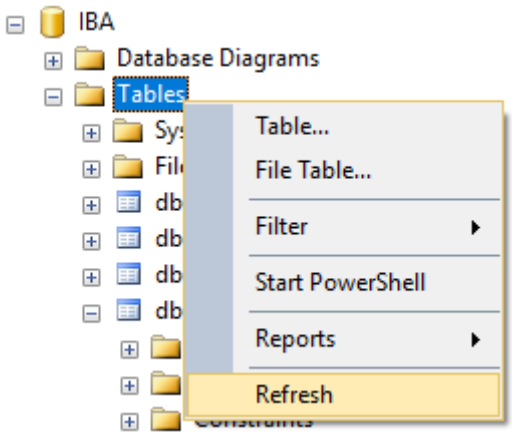
The following message will be shown after the successful creation of all required columns.



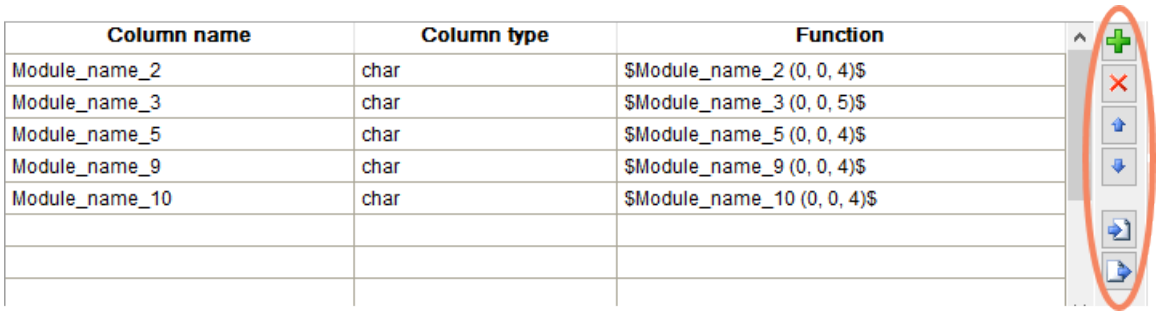
Location of Info data in database



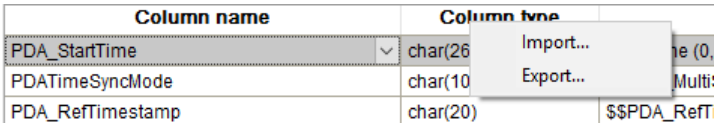
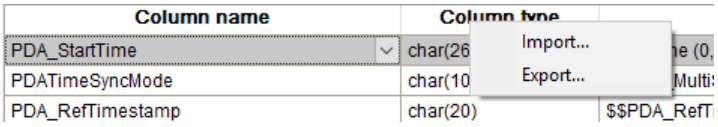
Remember to "Refresh" Tables in the database!



5.3.2 Procedures for manipulating Info columns

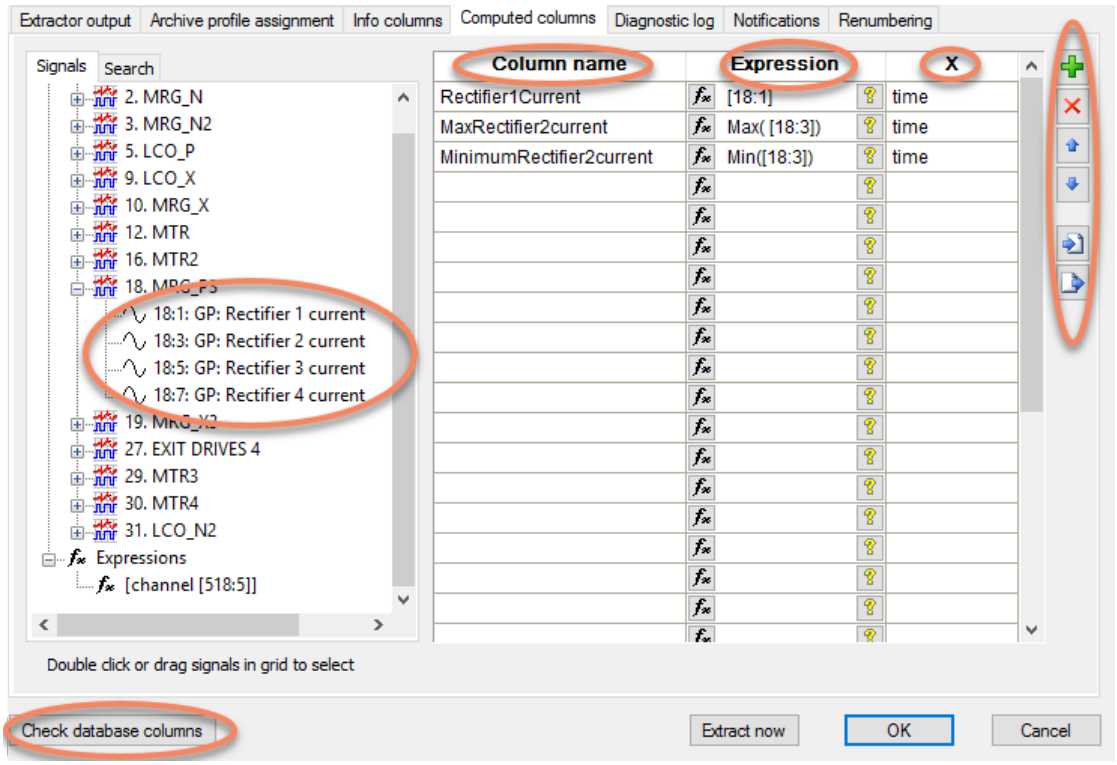


Column name	Column type	Function
Module_name_2	char	$Module_name_2 (0, 0, 4)$
Module_name_3	char	$Module_name_3 (0, 0, 5)$
Module_name_5	char	$Module_name_5 (0, 0, 4)$
Module_name_9	char	$Module_name_9 (0, 0, 4)$
Module_name_10	char	$Module_name_10 (0, 0, 4)$

	Add line	Adds a new empty line above the currently selected line.
	Delete line	Deletes the currently selected line.
	Move line up	Moves currently selected line up.
	Move line down	Moves currently selected line down.
	Import info file	Import Info Column information from text file. The "Open text file" dialog appears.
		Alternative: right click on header, select <Import> 
	Export info file	Export Info Column information to text file. The "Save text file" dialog appears.
		Alternative: right click on header, select <Export> 

5.4 Computed columns

The Computed columns tab is used to specify operations which can be applied to the individual signals. These operations will create one value for the associated signal (e.g. the average, maximum, or minimum of all the data in one signal). Values which are defined as "Computed columns" will be stored in the file table of the database (see: chapter: "Database Table deFile) and thus can be part of a query condition or can be used for long term trend analysis.



Select and prepare signal

Double click on required signal (drag and drop also works). The signal will be copied into the "Computed column" list.

The "Column name" initially takes the source signal name but can be altered as required.

The "Expression" initially takes the source signal channel number.

The expression can be altered by selecting <fx > the "Expression builder". The result of the expression should be one signal value. If no function is defined the average of the signal data will be calculated.

The "X" cell is automatically filled with the source signal cycle mode.

The list manipulation functions are described in section "Info Columns".

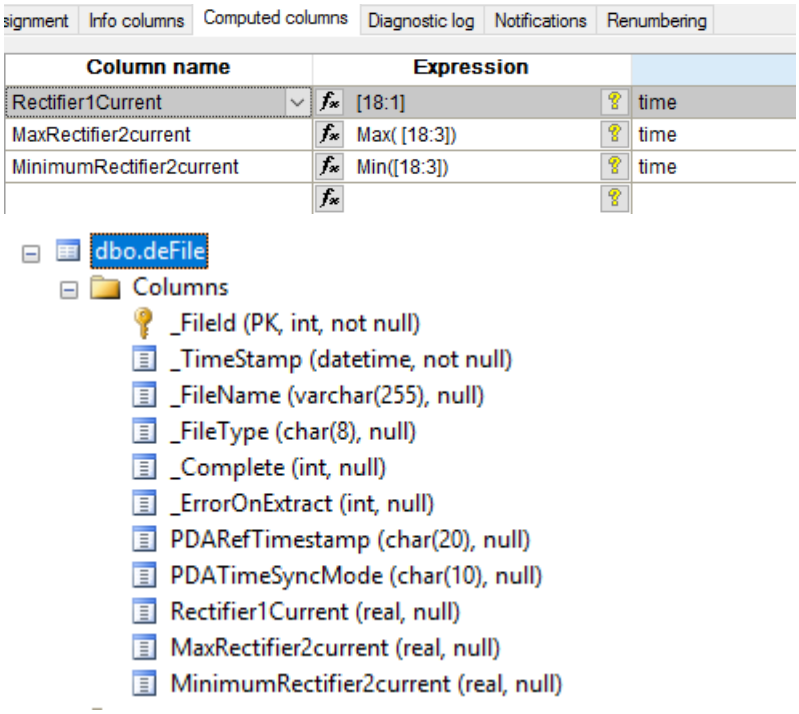
Note

If no group function is specified in the expression *ibaAnalyzer* automatically uses the average.

❑ <Check database columns>
After the computed columns are configured it is recommended to synchronize these definitions with the columns of the file table.

Location of "Computed column" data in database

Remember to Refresh Tables!



❑ Procedures for manipulating "Computed columns" column

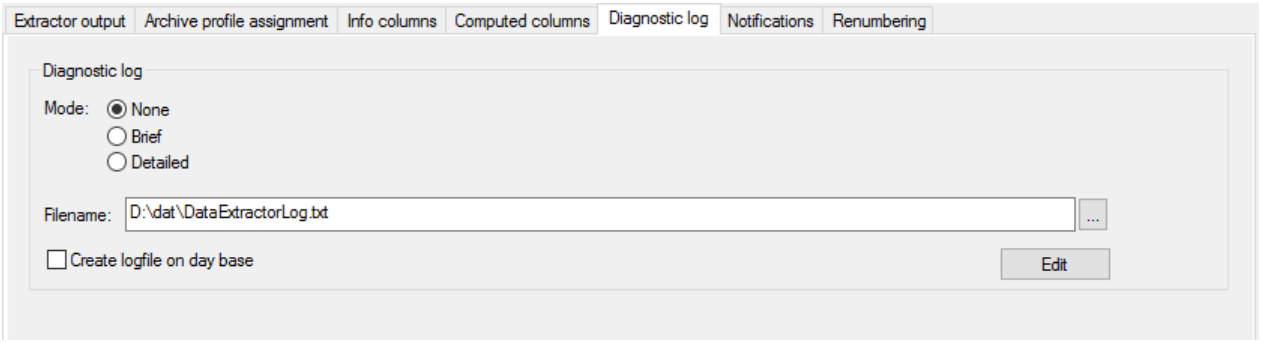
↗ See chapter: *Procedures for manipulating Info columns*, page 47

5.5 Diagnosis and notification

In a production environment it is necessary to monitor automated processes, to log processing information for diagnostic purpose and to generate notifications.

5.5.1 Diagnostic log

The diagnostic log is useful for checking progress when automatic extraction is operating (e.g. triggered by the ibaDatCoordinator). If a log mode is selected, the log file will be created and updated with each extraction.



- ❑ Mode
  - None: Log switched off.
  - Brief: Contents of log file

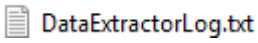
```
24-May-17 13:45:54: Start extract file 'D:\IBA\dat files\dat-training\pda_training021.dat'
24-May-17 13:46:57: Start extract file 'D:\IBA\dat files\dat-training\pda_training021.dat'
```

  - Detailed: Contents of log file:

```
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
```

- ❑ Filename

Enter name of diagnostic file. If daily log has not been selected this would be, e.g.:

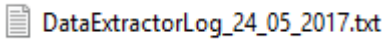


- ❑ <….>

Browse the network if required

- ❑ Create logfile on day base

One log file is created for each day.



- ❑ <Edit>

The selected log file can be opened in Notepad and changed as required.

Important information

If logging is activated permanently, a cleanup strategy for the log files has to be implemented externally (not part of ibaAnalyzer).

5.5.2 Notifications

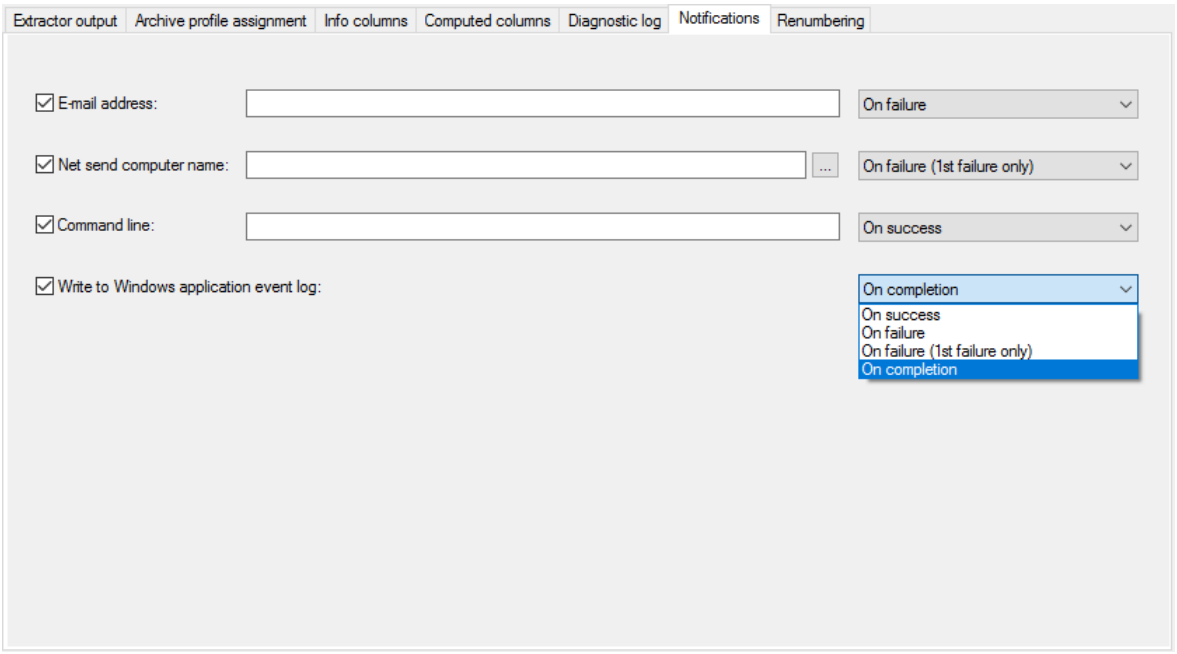
The *Notification* tab provides 4 means of communication triggered by a selected status of the extraction process. The following statuses are available:

❑ On completion
Notification made when the output procedure has been completed, success or failure is irrelevant.

❑ On success
Notification made when report output is successful.

❑ On failure
Notification made when any report output has failed.

❑ On failure (1st. failure only)
Notification made when first report output has failed



❑ E-Mail address
Enter address and select extraction mode.

❑ Net send computer name
Enter computer name and select extraction mode.

❑ <…>
Browse the network if required

❑ Command line

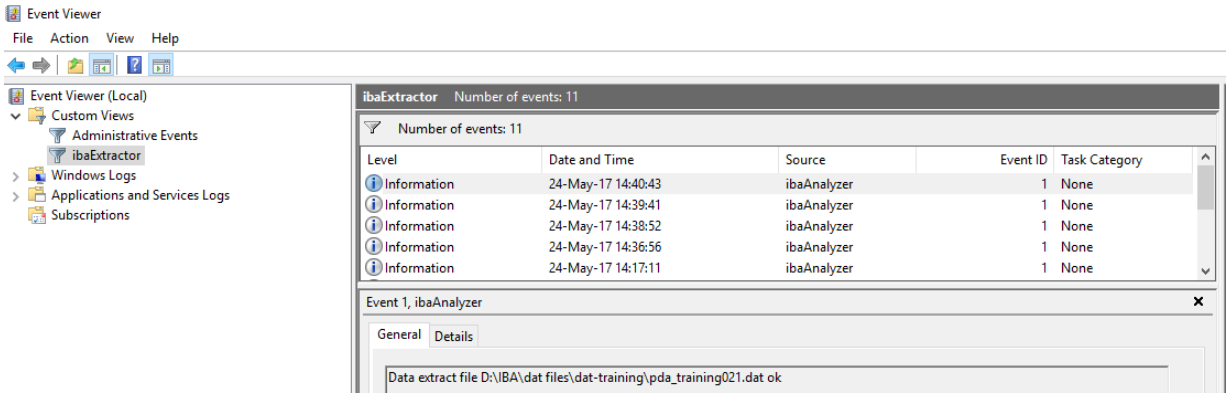
Enter command line script to be executed when extractor status achieved and select extraction mode.

Tip

The "Command line" option is not limited to notification tasks. Any application can be called to perform, for example, some post-processing.

❑ Write to Windows application event log

Select extraction mode. View in Windows Event Viewer

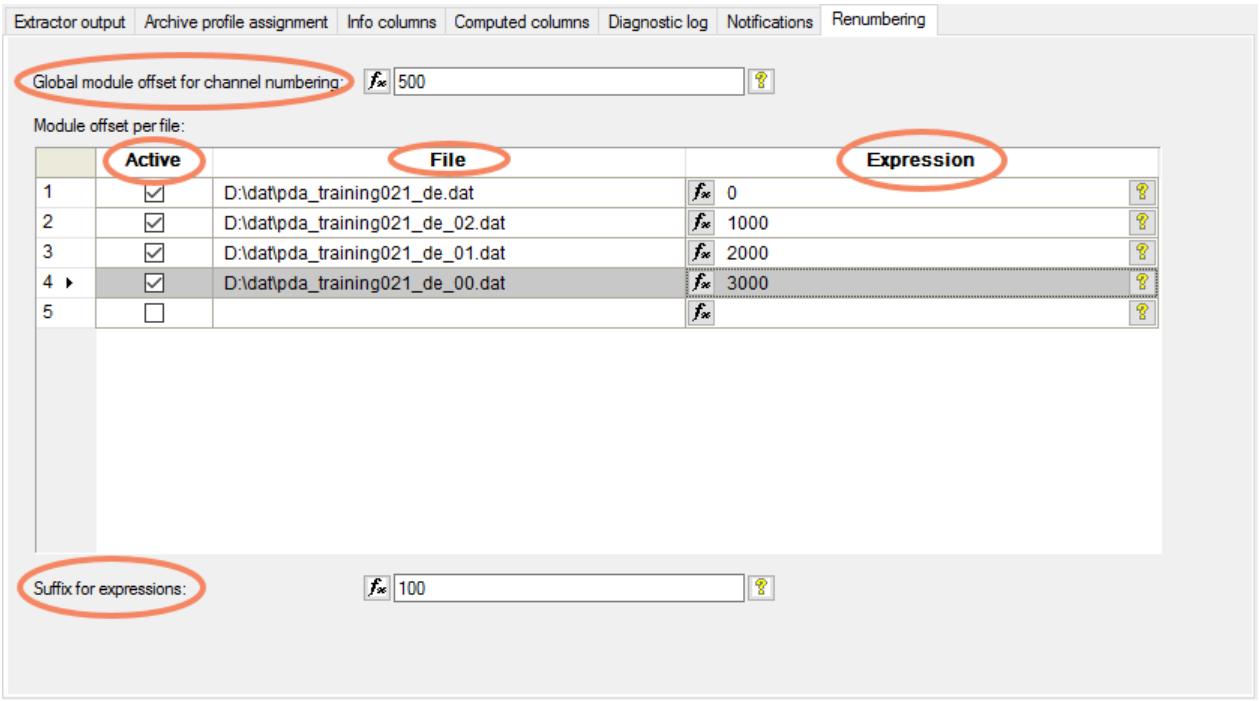


5.6 Renumbering

If multiple files are opened and several signals are to be extracted from several files, it is likely that multiple channels will have the same ID number and hence need to be renumbered or otherwise they will not be able to be uniquely identified in the exported media. *ibaAnalyzer* can do this renumbering automatically but will give a warning if it has to do so before proceeding with the extraction.

The channel IDs are comprised of a module number and a position within the module.

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.



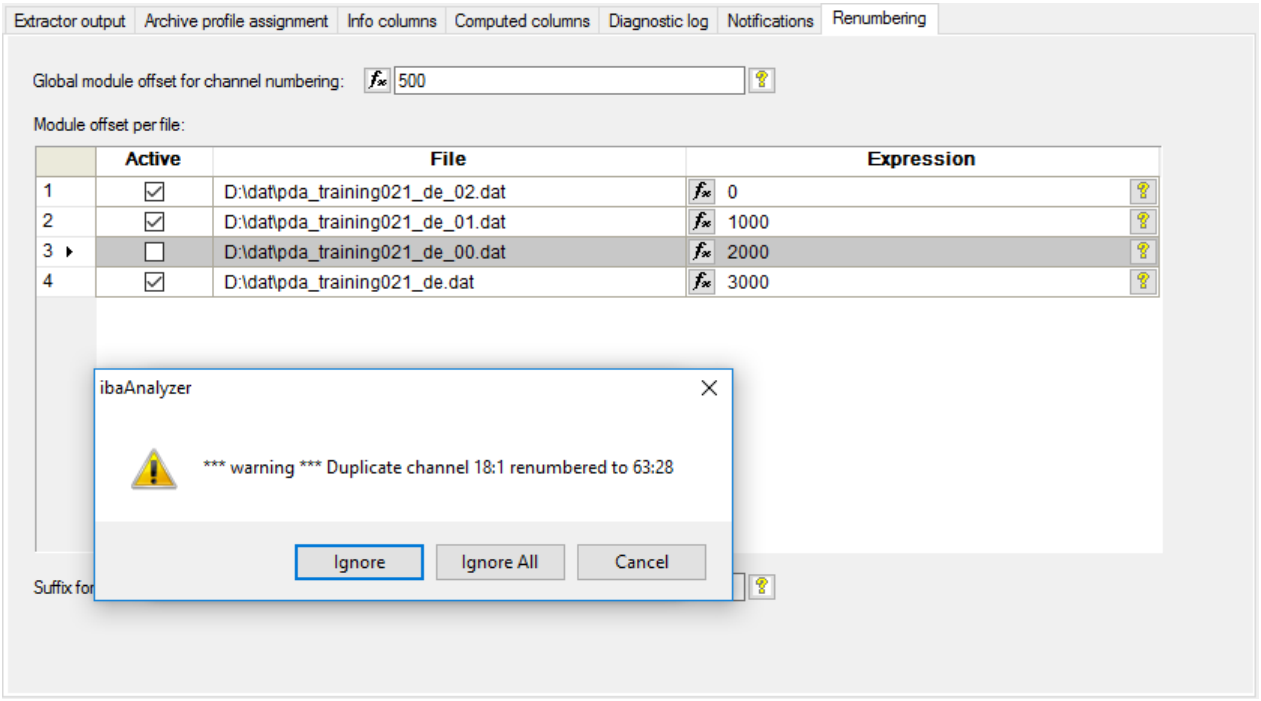
❑ Global module offset for channel numbering
This number will be added to data file channel numbers

❑ "File" column
The open .dat files are listed automatically in this column.

❑ "Expression" column
The offset for each data file is entered in this field

❑ "Active" column

Channel offsets will only be made for active files. Extraction of channels in inactive files leads to the following message.



Important information

The original and offset channel numbers are not visible in the database. They can only be seen *ibaAnalyzer* after database queries have been executed. See „Using *ibaAnalyzer-DB* for Analysis"

❑ Suffix for expressions

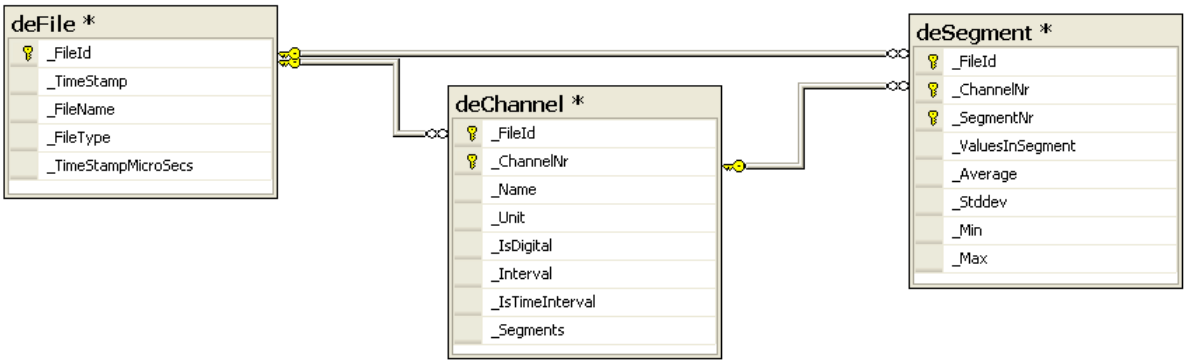
This applies only to the names and comments related to expressions created in ibaAnalyzer. The expression names are augmented by the specified suffix.

Example of extracted data based on above settings:

	_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

6 Database format options

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



6.1 Database table "deFile"

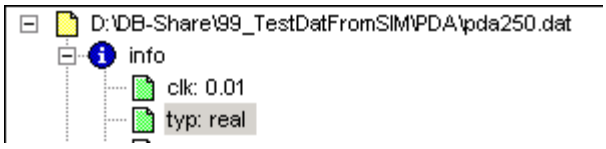
The table "deFile" contains the following header information for each extracted data record. A new data record is created for each extraction. The handling of identical source files is described in "Delete duplicate file rows"

❑ "_FileID"
This is a unique internal ID assigned to each record.

❑ "_TimeStamp"
Information on the date and time of the beginning of the acquisition is given in the second column.

❑ "_FileName"
The reference to the filename of the data file ("_FileName") is used for information or as a 'pathname' for the display of datasets in ibaAnalyzer

❑ "_FileType"
The file type ("_FileType") refers to the recording system, e. g. real => ibaPDA recording, QDR2 => ibaQDR-V6 recorder.



❑ "_Complete"
This shows the status of the extraction process. The first record of a data file extraction always is inserted into the file header table ("deFile") with status "_Complete = 0". After finishing all other inserts in the channel and segment tables the flag is updated to "_Complete = 1". When querying the "deFile" table the WHERE clause should check this "complete" flag to see whether the corresponding extraction was successful. For databases which support transaction encapsulation (e. g. Oracle, MSSQL, DB2-UDB) the option "Use transactions" can be selected (see chapter: *Option Extractor Database Library (Standard)*, page 29. In this case, the "_Complete" and the "_ErrorOnExtract" flags are obsolete.

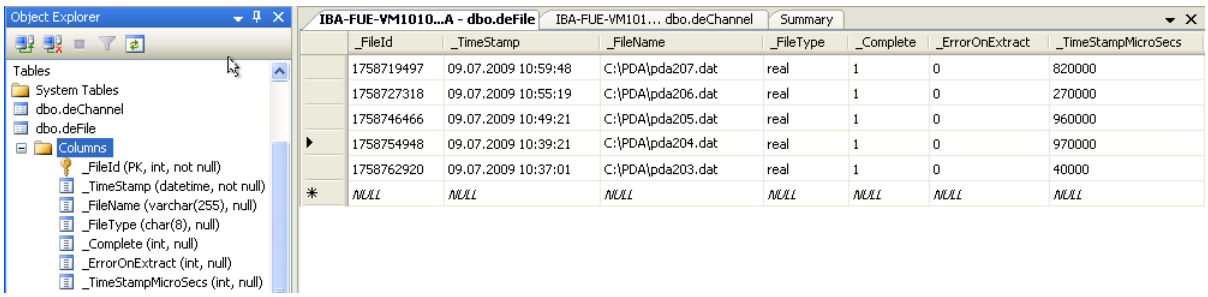
❑ _"ErrorOnExtract"
This is set if an extraction error occurs.

❑ "_TimeStampMicroSecs"
This function is enabled by selecting the "Extract microseconds column" option in the Extractor output tab (See chapter *Option Extractor Database Library (Standard)*, page 29).

Note

In many database systems there are data types available, which can handle microsecond resolution. However, a separate column has been introduced for this purpose to maintain backward compatibility.

The deFile is displayed in the database as shown below.



_FileId	_TimeStamp	_FileName	_FileType	_Complete	_ErrorOnExtract	_TimeStampMicroSecs
1758719497	09.07.2009 10:59:48	C:\PDA\pda207.dat	real	1	0	820000
1758727318	09.07.2009 10:55:19	C:\PDA\pda206.dat	real	1	0	270000
1758746466	09.07.2009 10:49:21	C:\PDA\pda205.dat	real	1	0	960000
1758754948	09.07.2009 10:39:21	C:\PDA\pda204.dat	real	1	0	970000
1758762920	09.07.2009 10:37:01	C:\PDA\pda203.dat	real	1	0	40000
NULL	NULL	NULL	NULL	NULL	NULL	NULL

6.2 Database table "deChannel"

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

❑ "_FileID"

↗ See chapter *Database table "deFile"*, page 56

❑ "_ChannelNr"

Unique internal channel ID assigned to the extracted signal

❑ "_Name "

The name of the extracted signal

❑ "_Unit"

The physical unit of the extracted signal

❑ "_IsDigital"

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

❑ "_Interval"

This is the sampling cycle of the extracted data which is described in the chapter *Archive profile assignment*, page 37

❑ "_IsTimeInterval".

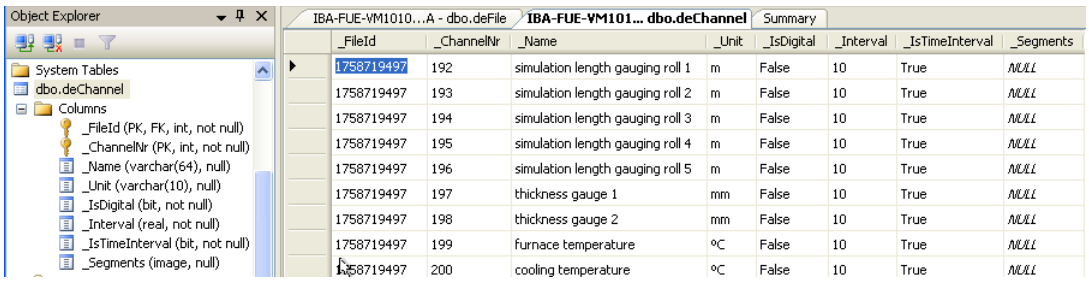
This flag is set for time based records

❑ "_Segments"

This column contains compressed data, which are extracted as "BLOBs" (Binary large objects), if the corresponding option is selected for extraction (see chapter *Using BLOBs*, page 59

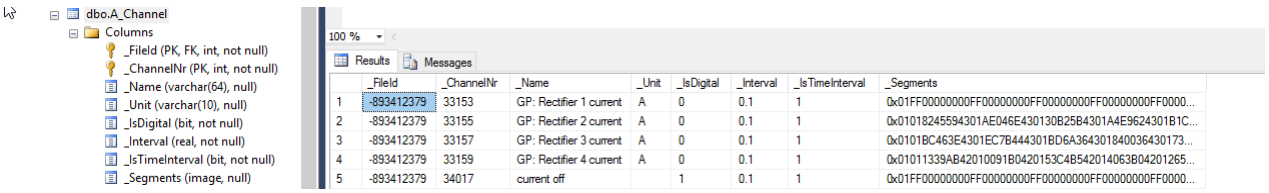
The "deChannel" table is displayed in the database as shown in the figures below.

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...

6.2.1 Using BLOBs

Instead of storing measured data in readable number columns in segment tables, it is also possible to keep the binary format when extracting data files into the database. The lossless compression (only changes are stored) saves, depending on signal characteristic, a huge amount of disc space and so enables database processing of high resolution measurement data. To access this compressed data with third party applications, a decompression function is required and has to be implemented individually.

Channel tables in Oracle also have a column "I_ISBLOB", which states whether the corresponding "I_SEGMENTS" column contains binary data.

If *ibaAnalyzer-DB* is used exclusively as an analysis tool, implementing BLOBs can be a very efficient storage solution. (This option can be enabled in the "*Archive profile assignment*, page 37" window.)

Note

BLOBs are not supported for SQLite databases.

6.3 Segment tables

6.3.1 Standard format segment table

The table "deSegment" contains the values of all signals which are stored in the database.

Important information

There is one table for all values!

Important information

This table is only filled when the "Blob" option has not been selected!

- "_FileID"

↗ See chapter *Database table "deFile"*, page 56

- "_ChannelNr"

↗ See chapter *Database table "deChannel"*, page 58

- "_SegmentNr "

Each compressed sample is assigned a segment number. The segment number is incremented automatically for each channel, starting from 0 (zero) until the end of the record.

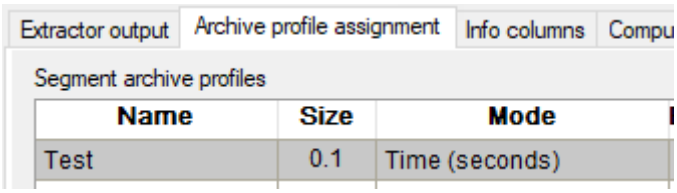
- "_ValuesInSegment"

This displays the number of values in the raw data file included in each segment (sample cycle) stored in the database.

Example

Raw-data in the data file: 1 value every 10 ms.

Data extract to database: 1 value every 100 ms (= sample cycle). Here "Size" is set to 0.1 based on a "Mode" of "Time (seconds)" set in "*Archive profile assignment*, page 37"



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

- "_Average"

Average value of signal values within each segment.

- "_Stddev"

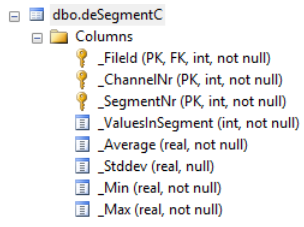
Standard deviation of signal values within each segment.

- "_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 below.



Results | Messages

	_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

6.3.2 MC-format segment tables

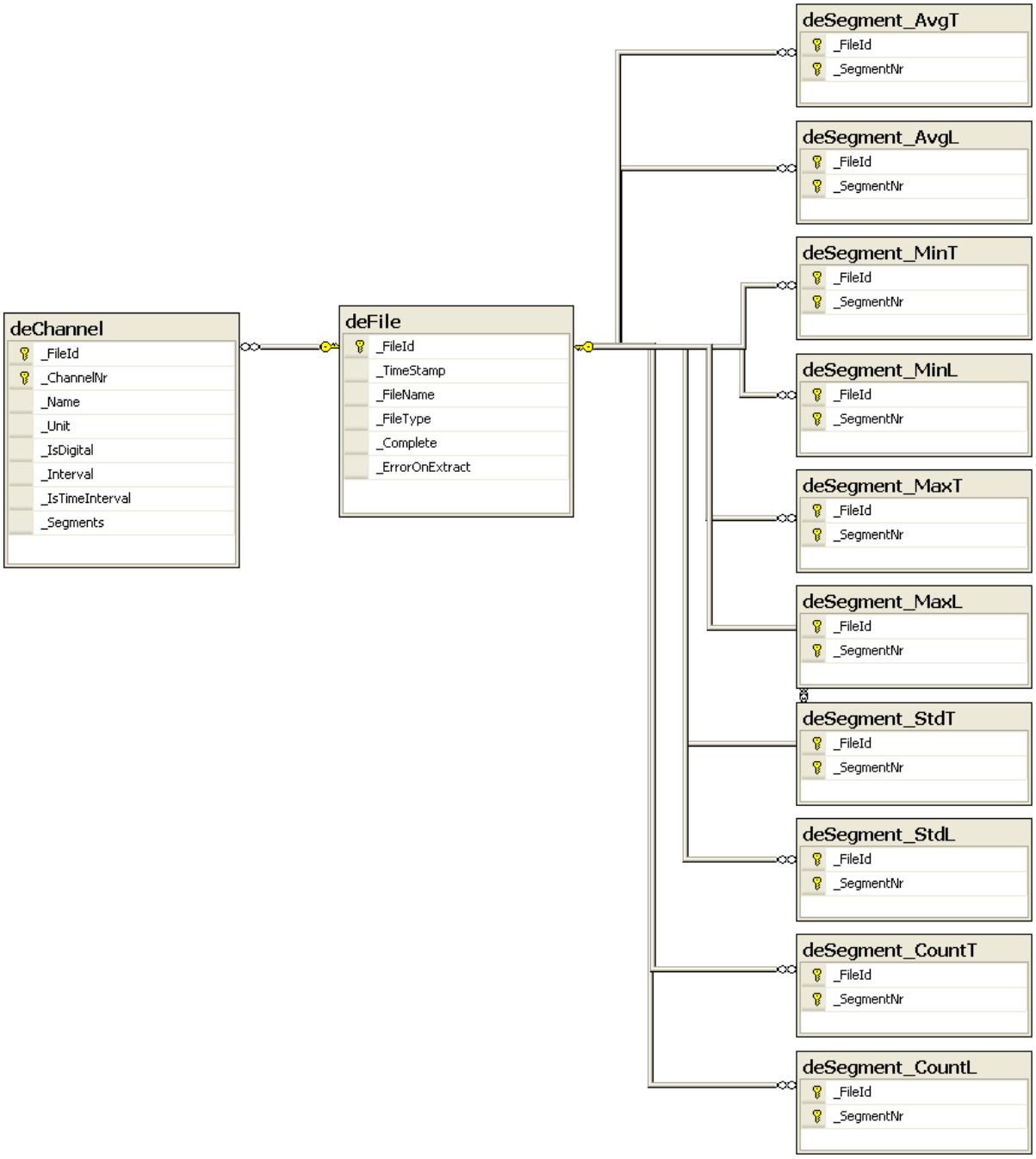
The Multi Column (MC)-format provides five identically structured tables for time(T)- and length(L)-based extracted data.

- Average (table deSegment_AverageT/L),
- Minimum (table deSegment_MinT/L),

Maximum (table deSegment_MaxT/L),

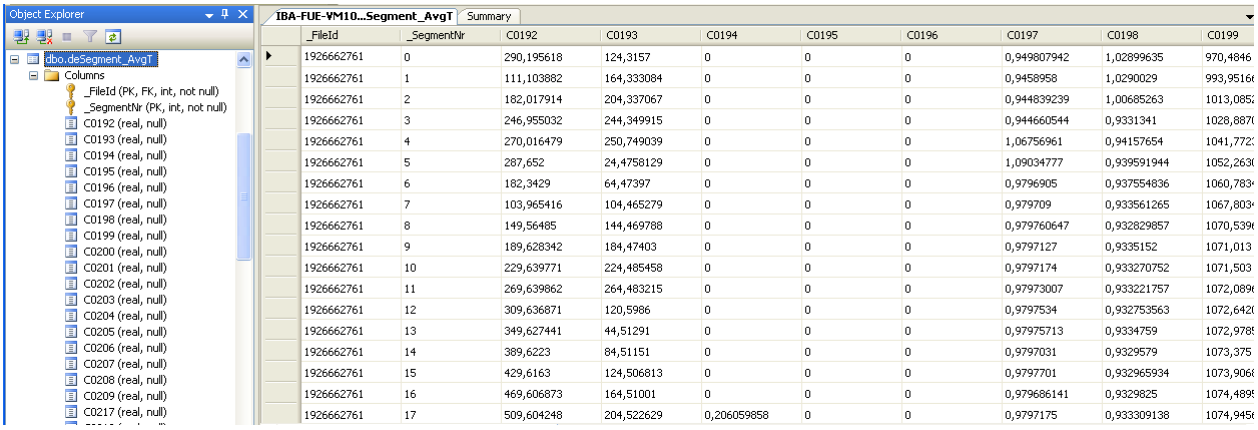
- Standard deviation (table deSegment_StdT/L) and
- SampleCount (table deSegment_CountT/L).

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 N columns*, additional segment tables may be created automatically. This option is explained in section "Start a new segment table every <…..> columns". This can be used to avoid database table column number restrictions.

Before inserting the segment records *ibaAnalyzer-DB* checks if columns for all the channels exist. If not, the segment tables will be altered. By default, the column names for the channels are derived from the channel number. Format and contents of MC database table "deSegment_AvgT" in MSSQL-format:



_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

Important information

These tables are only filled when the "Blob" option has not been selected!

6.4 Differences between used database products

There are some differences between the databases supported by OLE-DB API (SQL-Server, Oracle, DB2-UDB) and by ODBC (MySQL, PostgreSQL, MS Access, etc.). For example, the column names of *ibaAnalyzer-DB* for Oracle and MSSQL are not identical. With large database systems that have their history based on mainframe, VMS or UNIX platforms and architectures (DB2-UDB, Oracle), the restrictions, especially for object and column naming, are very strong. Thus some features of *ibaAnalyzer-DB* cannot work with all database systems (e.g. BLOBs or cascading constraints see chapter: "*Customizing and integration*, page 64").

6.5 Customizing and integration

ibaAnalyzer-DB provides a basic framework for database integration which means that it creates standard tables, constraints and indexes by mouse click. In small standalone environments this may be sufficient. But even in these environments some implementations, such as data deletion or archiving, are recommended to guarantee availability.

If high resolution data is to be loaded into a database there will be requirements for archiving and, in particular, delete functions. These delete functions have to be derived from the individual data management strategy. A very simple strategy could be for example:

Every night at 02:00 AM run an automated job, that deletes all data older than 100 days

To do this, a simple SQL command (MSSQL syntax) has to be processed every night and this command could be put in a short SQL script ("del_old_recs.sql"):

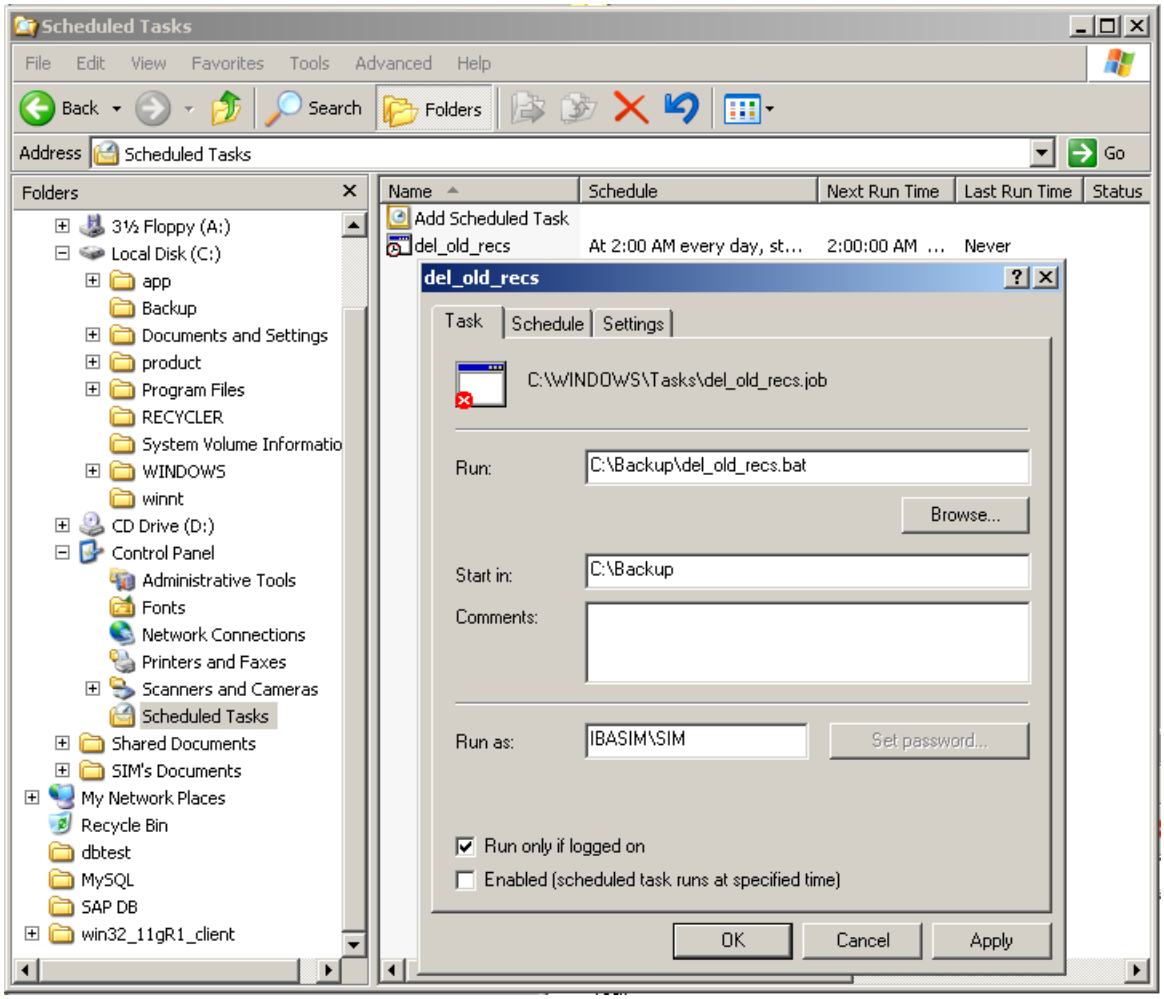
```
1 USE IBADB
2 GO
3 delete deFile where _TimeStamp < (GetDate() - 100);
4 GO
5 COMMIT
6 GO
```

The "cascading constraints" in MySql, DB2-UDB, MSSQL, PostgreSQL and Oracle databases, ensure that when a row in the file table is deleted, the corresponding rows in the segment and channel tables are deleted as well (referential integrity).

In a windows environment it is possible to create a batch file ("del_old_recs.bat")

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

and define (Add) a "Scheduled Task" in the control panel "del old recs" (see figure below).



Important information

Deleting records does not always ensure an increase of free disk space. Depending on the database system, additional maintenance commands or procedures may have to be executed.

Data management strategies may also depend on related third party applications.
For example a MES or QM application gives the triggers for deletion or archiving of records that come from product related data files. In this case a product-id reference is required in the file table

Another part of the data management strategy covers requirements for the disk storage. How much hard disk space (e.g. to store data for 100 days) is required to implement *ibaAnalyzer-DB*? By using information from the database supplier it is possible to calculate the different record lengths, the index space requirements and so on. In reality it is better to simulate typical data processing and monitor the space consumption. It is very easy to configure a typical database extraction process using some test data.

Using the extracted measurement data together with further data in a production or information system environment sometimes requires additional INDEXES for performance tuning or information columns that will be filled from external procedures. All the "insert" statements of *ibaAnalyzer-DB* are "fully qualified". Therefore adding

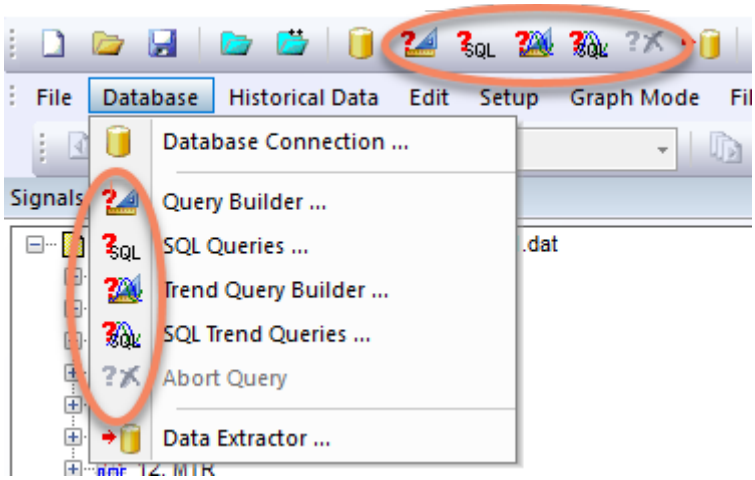
NULL-columns won't be a problem. Adding INDEXES is also possible and recommended. If database triggers are necessary, especially on the long segment table they may slow down the load performance significantly.

A very efficient way to link *ibaAnalyzer-DB* tables with third party tables is by defining database views that hide complexity in JOIN conditions.

7 Using ibaAnalyzer-DB for analysis

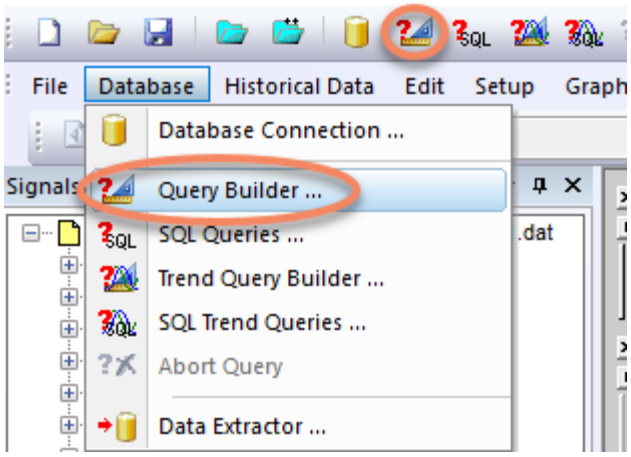
ibaAnalyzer-DB not only uses *ibaAnalyzer* in the ETL (Extract Transform Load) process, it also provides powerful functions to query data from databases for standard or trend analysis in ibaAnalyzer.

The query definition forms can be started from the *Database* menu or directly from the toolbar.

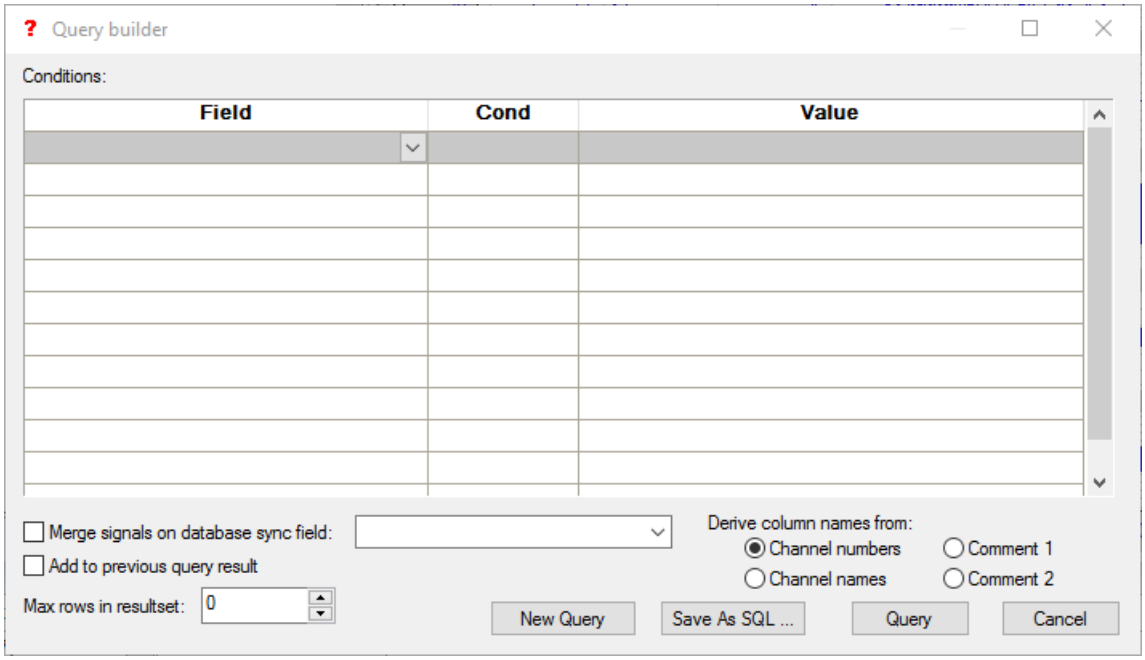


7.1 Standard queries

7.1.1 Query builder

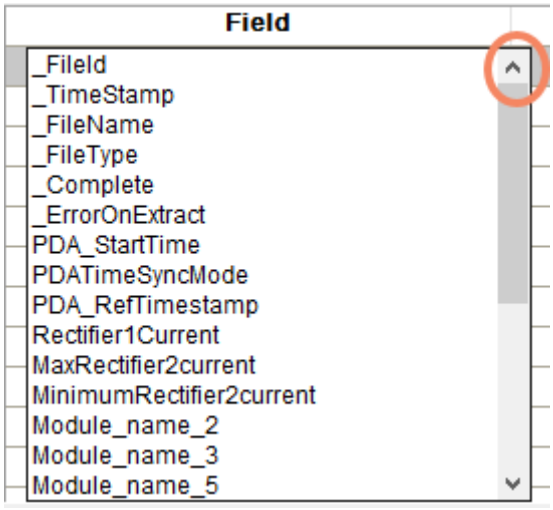


It is possible to create standard queries without any knowledge of SQL (Structured Query Language). *Query Builder* opens the following window.



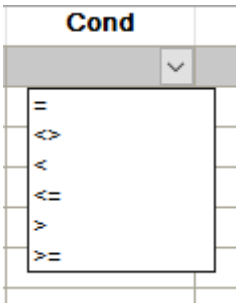
❑ Conditions

In the "Query Builder", conditions can be defined that are based on the contents of the main fields in a file table. When clicking on the drop down button in the "Field" column a selection of available fields pops up.



These fields are the same as shown in the database file table. Apart from the standard fields such as _FileID, _TimeStamp, _FileName or _FileType you can also select self-created fields from the technostring field assignment, e.g. 'STRIP_ID'.

A condition can be selected from the options made available by clicking on a cell in the "Cond" column.

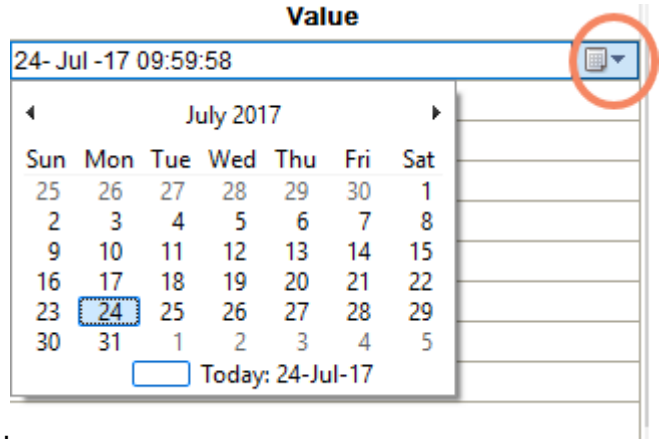


The comparator, which must be compatible with the database format, is entered in the "Value" column.

Several rows of queries can be entered which are then automatically combined by AND keywords.

Field	Cond	Value
_TimeStampMicroSecs	>	100000
_TimeStampMicroSecs	<	140000

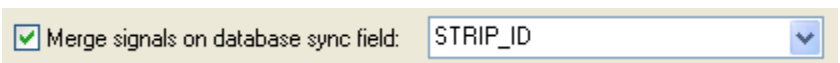
If the database field type of the chosen field is "Date" or "DateTime", a calendar pops up automatically when the drop down button is selected.



.

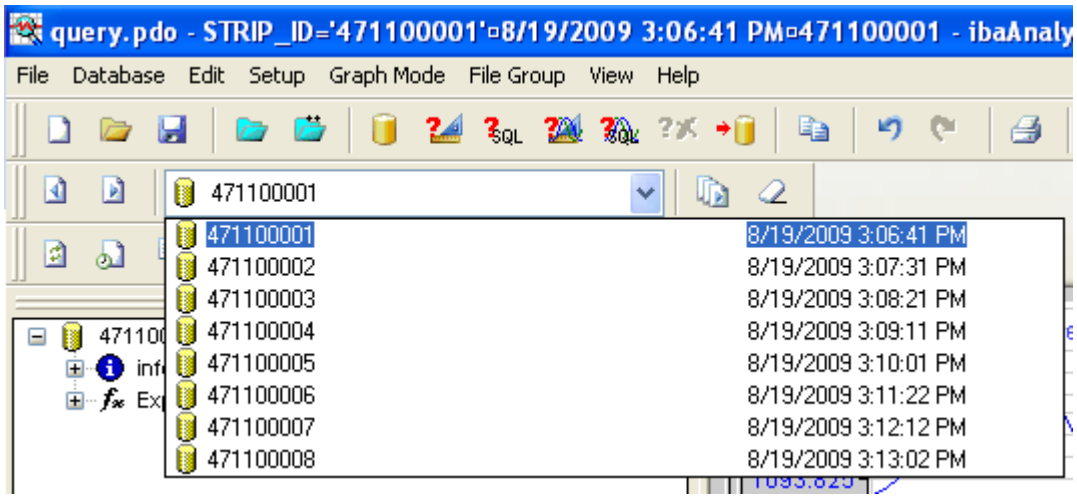
❑ Merge signals on database sync field

If this option is selected then the database files will be selected and displayed according to the sync field value.



If data extracted from different data files (different ibaPDAs or recorders) are associated by a common identifier, such as for example, the material-id, and these are to be merged, a "sync field" can be defined (see chapter: *MC-format segment tables*, page 61).

If the "sync field" option is used, all queried records (data files) that belong to one field (e. g. STRIP_ID) are merged and only the sync field's content is shown in the dropdown list.



❑ <Add to previous query result>
If this is activated the result of the query is added to previous query results. The option is equivalent to the "Add new data file…" function when working with data files. If this option is not selected all previous query results will be deleted.

❑ "Max rows in resultset <N>"
To prevent uncontrolled allocation of system resources (database server, network,…) the "Max rows in resultset" should have an appropriate value. A setting of 10 selects data from 10 data files. The value 0 (Zero) disables this restriction.

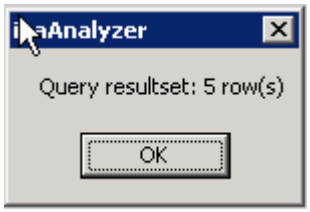
❑ "Derive column names from"

↗ See chapter: *Extractor output*, page 29

❑ <New Query>
Select this to clear the window and define a new query.

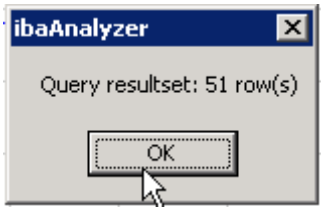
❑ <Save as SQL>
The SQL command which was generated by the query builder can be saved as a text file. This option is very useful for getting a basic statement that can be customized or extended and then executed in the "SQL queries…" window.

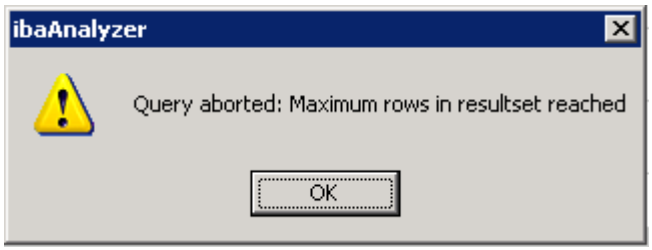
❑ <Query>
This executes the query to the configured database connection. If successfully executed, the number of fetched records is displayed:



Important information

If the "Max rows in resultset" parameter was exceeded the following message boxes are displayed (Example: Max rows in resultset = 50). These have to be confirmed and the query reconfigured:

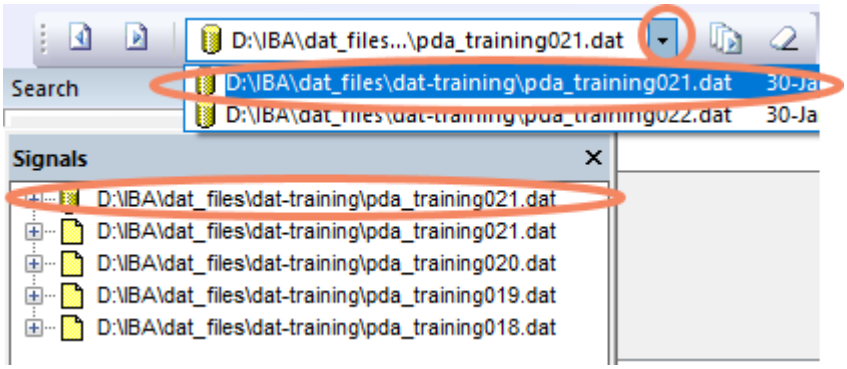




❑ <Cancel>
The window is closed without saving the query and any currently running query is terminated.

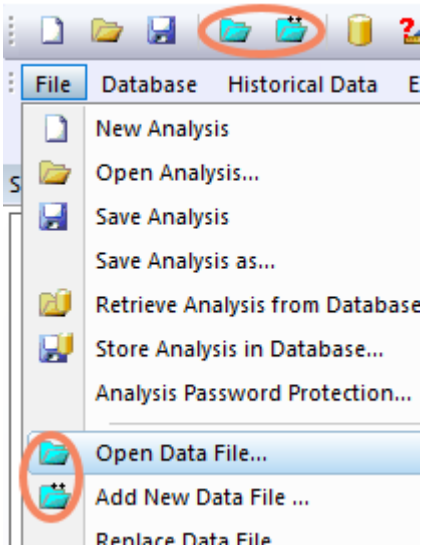
7.1.1.1 Handling queried records

The queried records (data files) can now be selected from a dropdown list (multiselect bar).

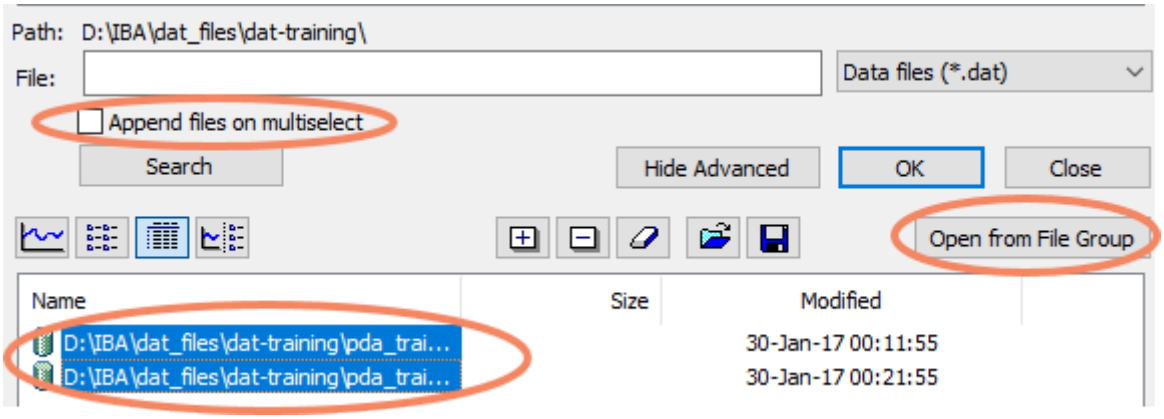


A record can be selected by clicking on the drop down button and then on the required record. The selected result will always be shown at the top of the tree view in the "Signals" window.

The queried records can be handled like data files in the advanced window of the "Open Data File …" or "Add New Data File… " dialogs.

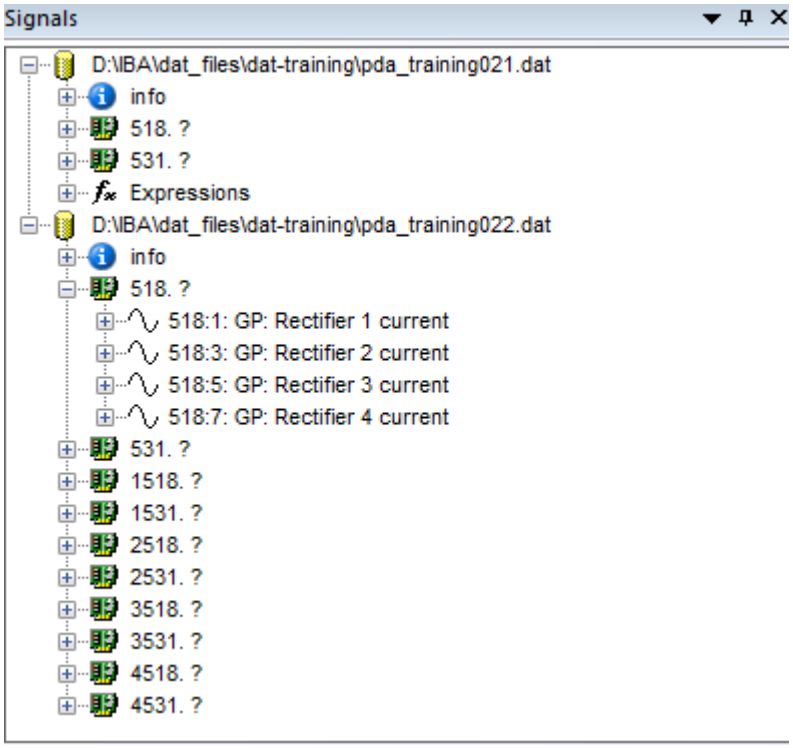


Select the required database files, decide on the option *Append files on multiselect* and select <Open from File Group>

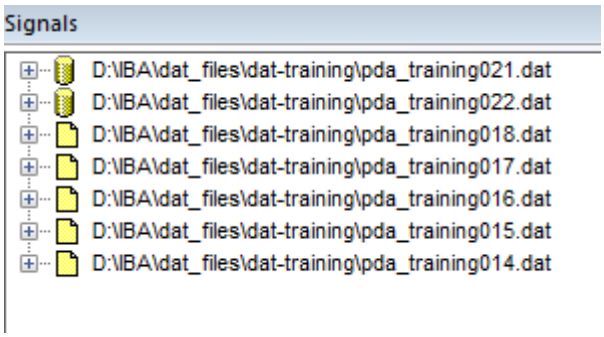


Using the *Append files on multiselect* option is also possible for queried data.

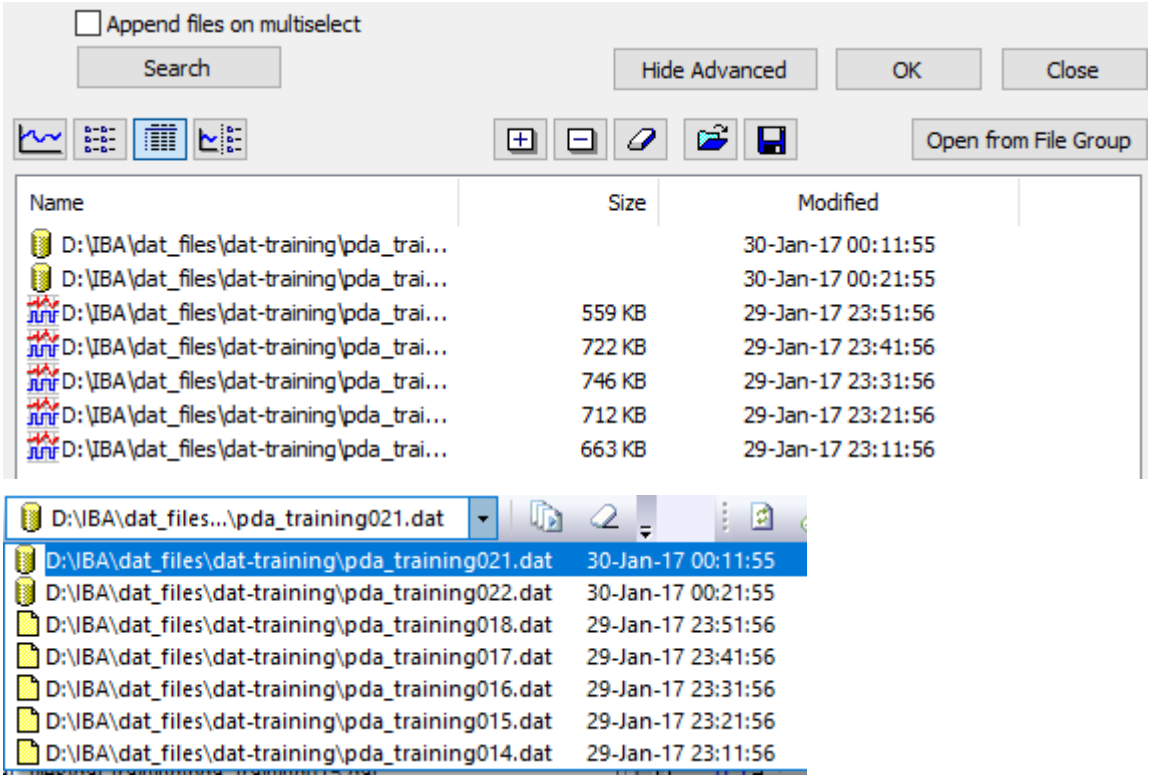
The selected database file will be displayed in the "Signals" window.



It is also possible to mix database queries and data files. The different data sources are distinguished by the icons for database and for data file.

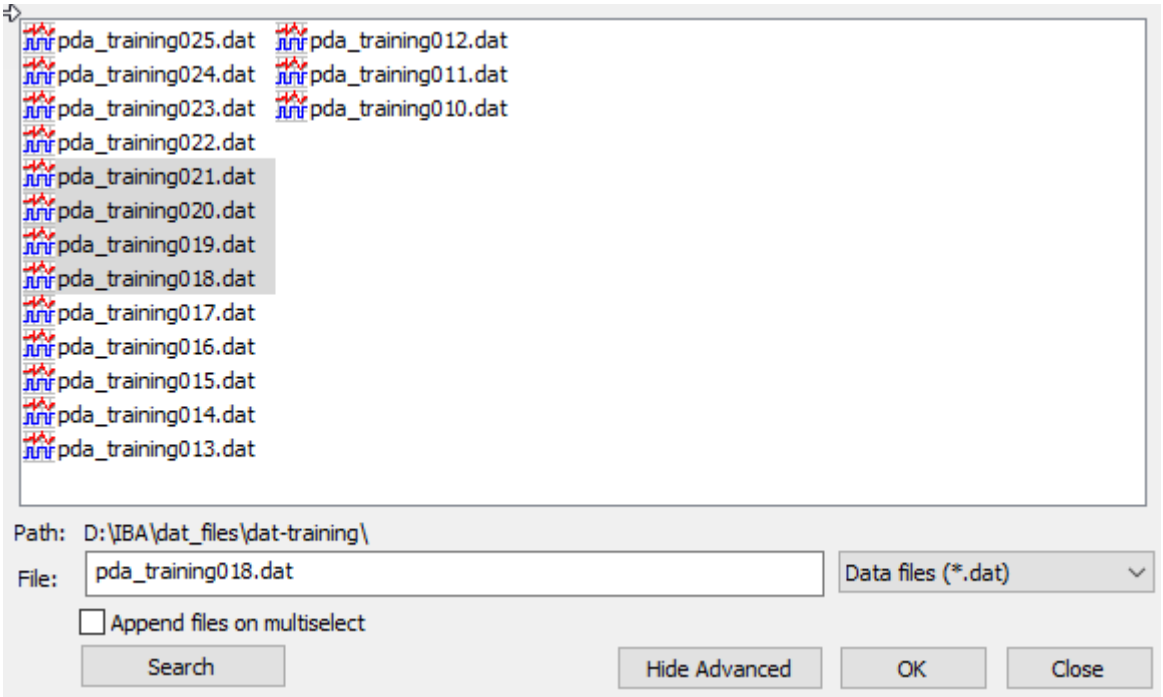


If the data files are opened and added in the "Advanced" window of the "Open/Add data file" dialogs they will be part of the group in the multiselect bar as well as being shown in the "Signals" window.

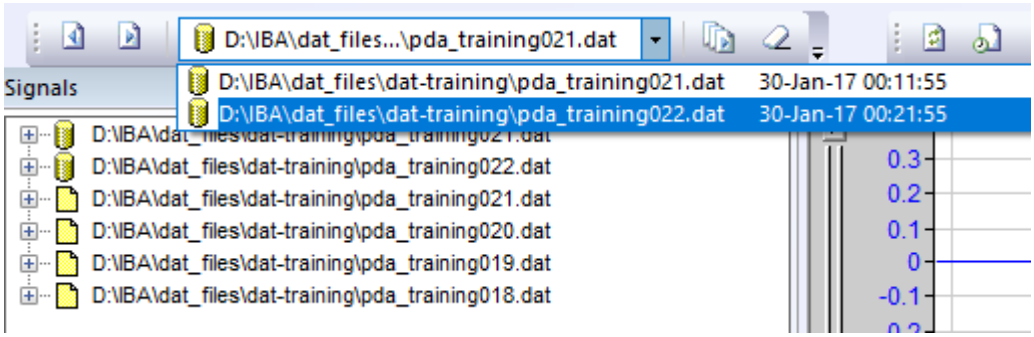


Added query results can always be found in the multiselect bar.

If data files are not opened via the "Advanced" window of the "Open data file" window,

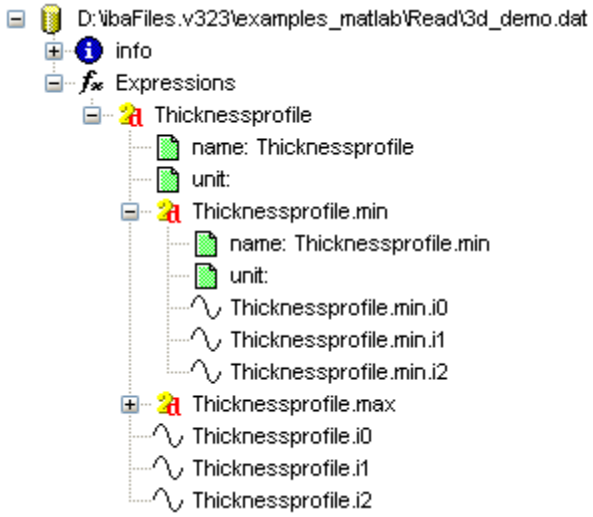


they will appear in the tree structure shown in the "Signals" window but not in the multiselect drop down menu.



❑ Multidimensional channel expressions

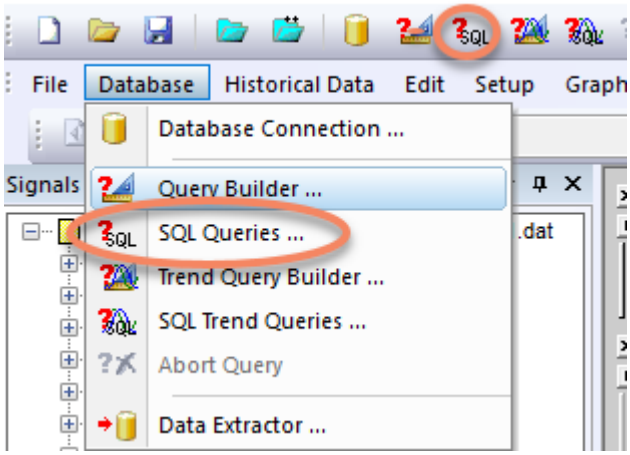
Queried multidimensional channel expressions are displayed under the group "Expressions" where they can be selected. The sub-signals can also be selected and referred to in derived expressions.



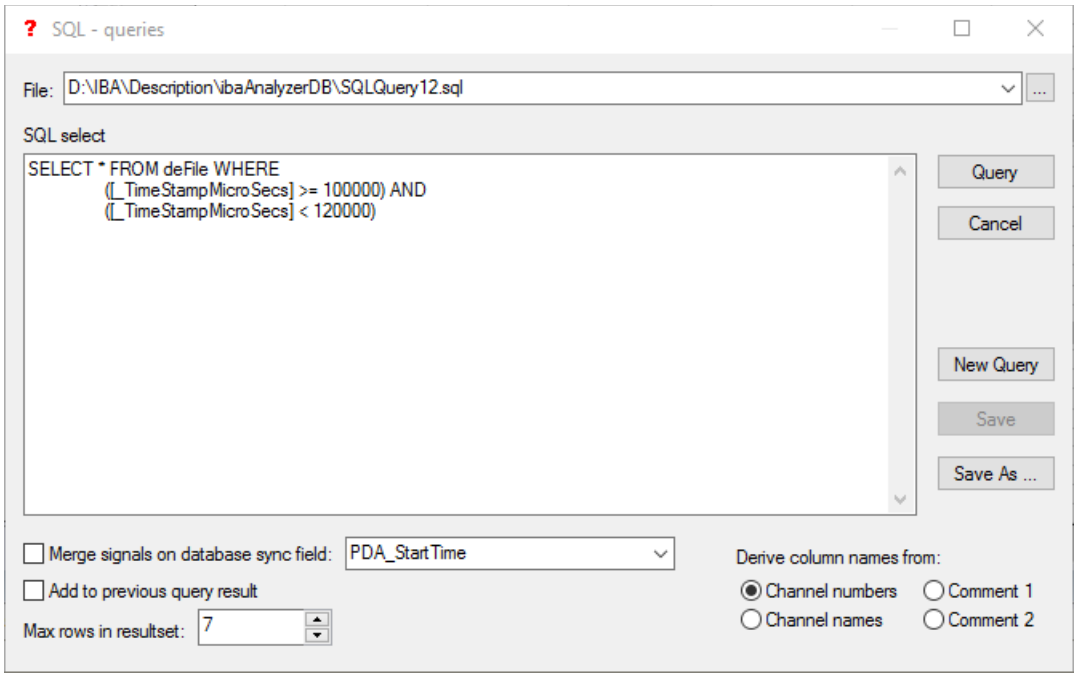
The signals can be selected as usual for display and further evaluation.

7.1.2 SQL-queries

It is also possible to write queries directly in SQL syntax. This option provides the full flexibility that is available within the used database/application environment. Any table that is accessible from the configured database connection can be used in the WHERE condition.



<SQL Queries > opens the following window.



❑ File
A previously created query file can be selected

❑ SQL select
SQL text can be written in this field.

❑ Merge signals on database sync field

↗ See chapter: *Query builder*, page 67

❑ Add to previous query result

↗ See chapter: *Query builder*, page 67

- Max rows in resultset
- See chapter: *Query builder*, page 67
- "Derive column names from:"
- See chapter: *Query builder*, page 67
- < Query>
- See chapter: *Query builder*, page 67
- <Cancel>
- See chapter: *Query builder*, page 67
- <New Query>
- See chapter: *Query builder*, page 67
- <Save>

Save changes to currently opened query file.

- <Save As>

Save as new query file.

Note

Only column names specified by the selected "Derive column names from:" option can be used in the SQL query.

Note

The SQL syntax may differ depending on the used database system and connection type (ODBC, OLEDB).

Example:

Querying Data of the last two days in SQL-Server:

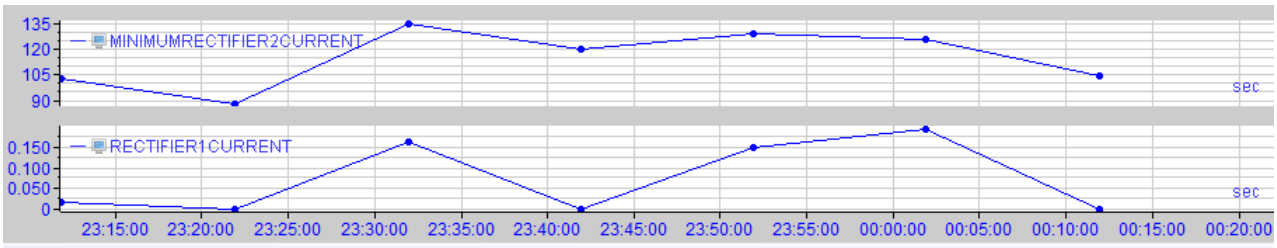
```
SELECT * FROM PDA_File WHERE _Timestamp > (getdate()- 2)
```

Equivalent query in Oracle syntax:

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

7.2 Trend queries

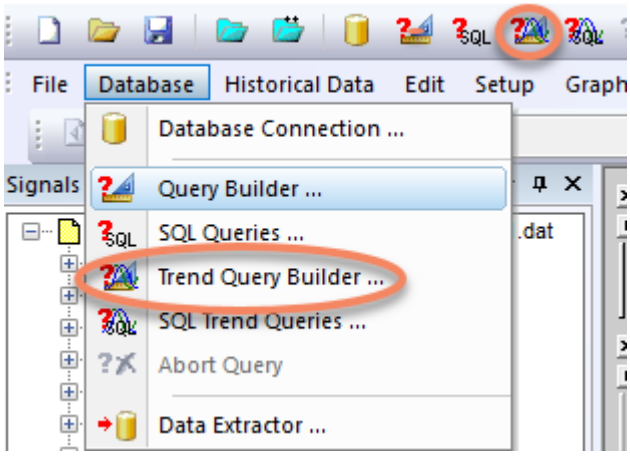
Instead of querying detailed measured data by standard queries it is also possible to query the info fields and/or computed columns which are aggregations of values in data files (i.e. one value per data file). This resolution permits long term analysis over days, weeks, months or years without generating excessive amounts of data. The file table created by *ibaAnalyzer-DB* is used to display a trend graph of that data. Theoretically data from other database tables or views can also be queried, provided that these tables contain a timestamp column and numeric or string data. The results of a trend query are signals containing non-equidistant data samples, similar to the XY view.



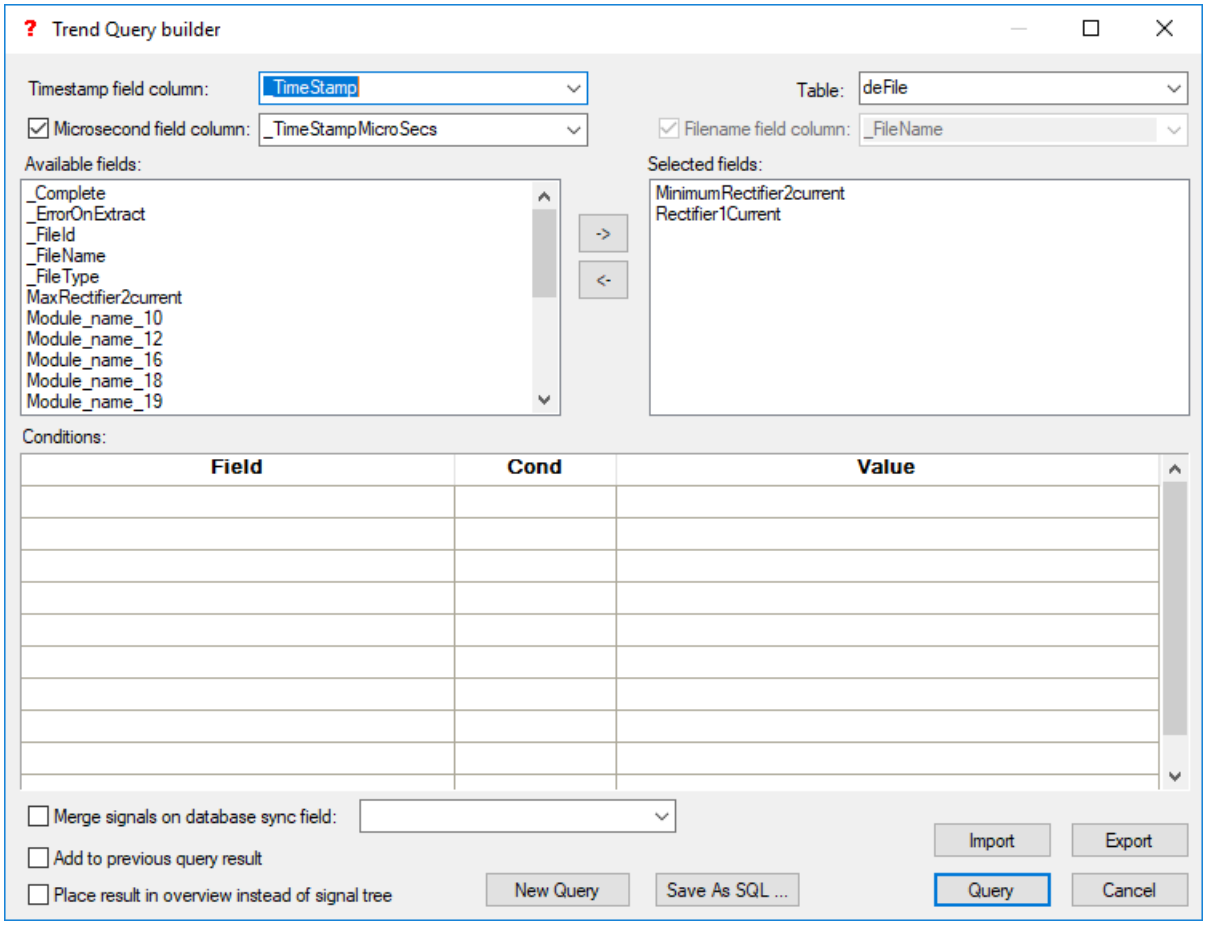
Each record in the query result is displayed by a point having an x-coordinate equal to the time stamp of the source file and a y-coordinate equal to the aggregate value of the sample. These generated signals can be used in further computations. However in this case they are first converted to equidistant signals by linear interpolation before being used in the given expression. The sample separation of the resampled signal will be the time interval between the closest two sample points in the original, non-equidistant signal.

7.2.1 Trend query builder

Similarly to the standard query builder, the trend query builder is available to assist in the creation of SQL instructions to perform a trend queries.



Trend Query Builder opens the following window.

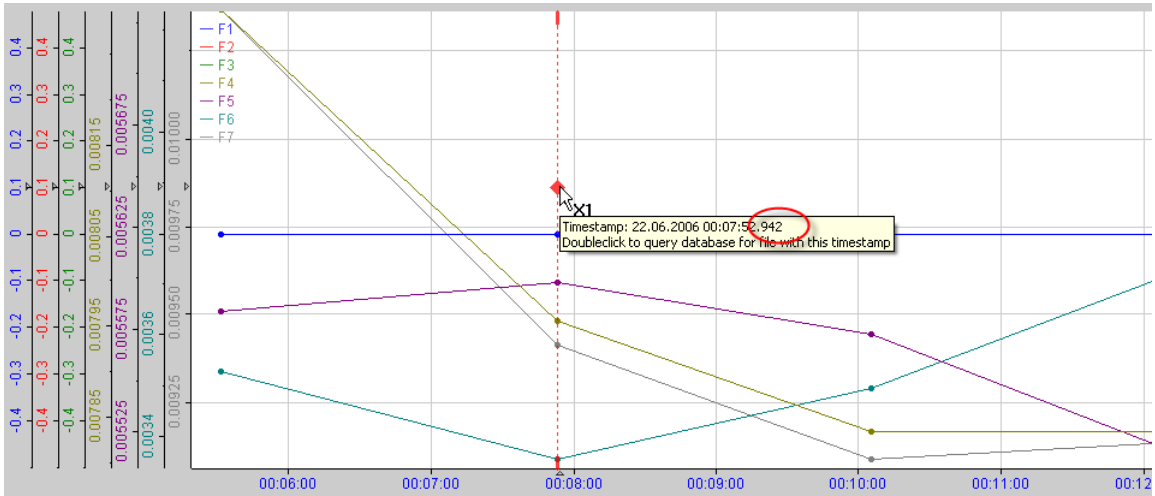


❑ Timestamp field column

Trend queries require a time stamp. The relevant column is selected here.

❑ Microsecond field column

This option is selected if microsecond precision is required (see chapter: "Database Table deFile",) and the column where the data is stored is selected here.



❑ Table

Select the table to be queried.

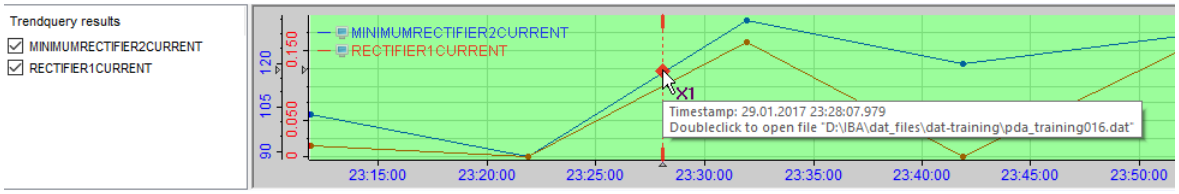
❑ Filename field column

This column contains the filenames.

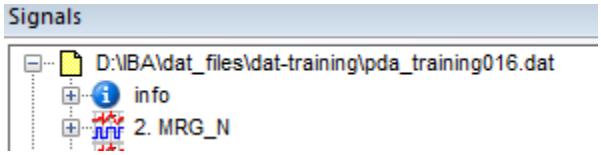
_FileName
D:\IBA\dat_files\dat-training\pda_training018.dat
D:\IBA\dat_files\dat-training\pda_training016.dat
D:\IBA\dat_files\dat-training\pda_training019.dat
D:\IBA\dat_files\dat-training\pda_training017.dat
D:\IBA\dat_files\dat-training\pda_training015.dat

Selecting this option will enable the opening of the original files corresponding to the extracted files in the database. If the table to be queried has the iba database layout, the correct column will be preselected.

In the "Overview" window move the cursor onto the diamond to view the name of the source file and double click.



The source file will appear in the "Signals" window.

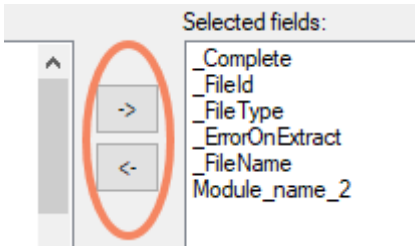


This option can be disabled if the original files are not required or no longer present as the query speed is then increased. This option is not available if the option "Place result in overview instead of signal tree" has not been selected.

❑ Available fields

All the fields which are available in the selected table (type "defile") are displayed

❑ Selected fields



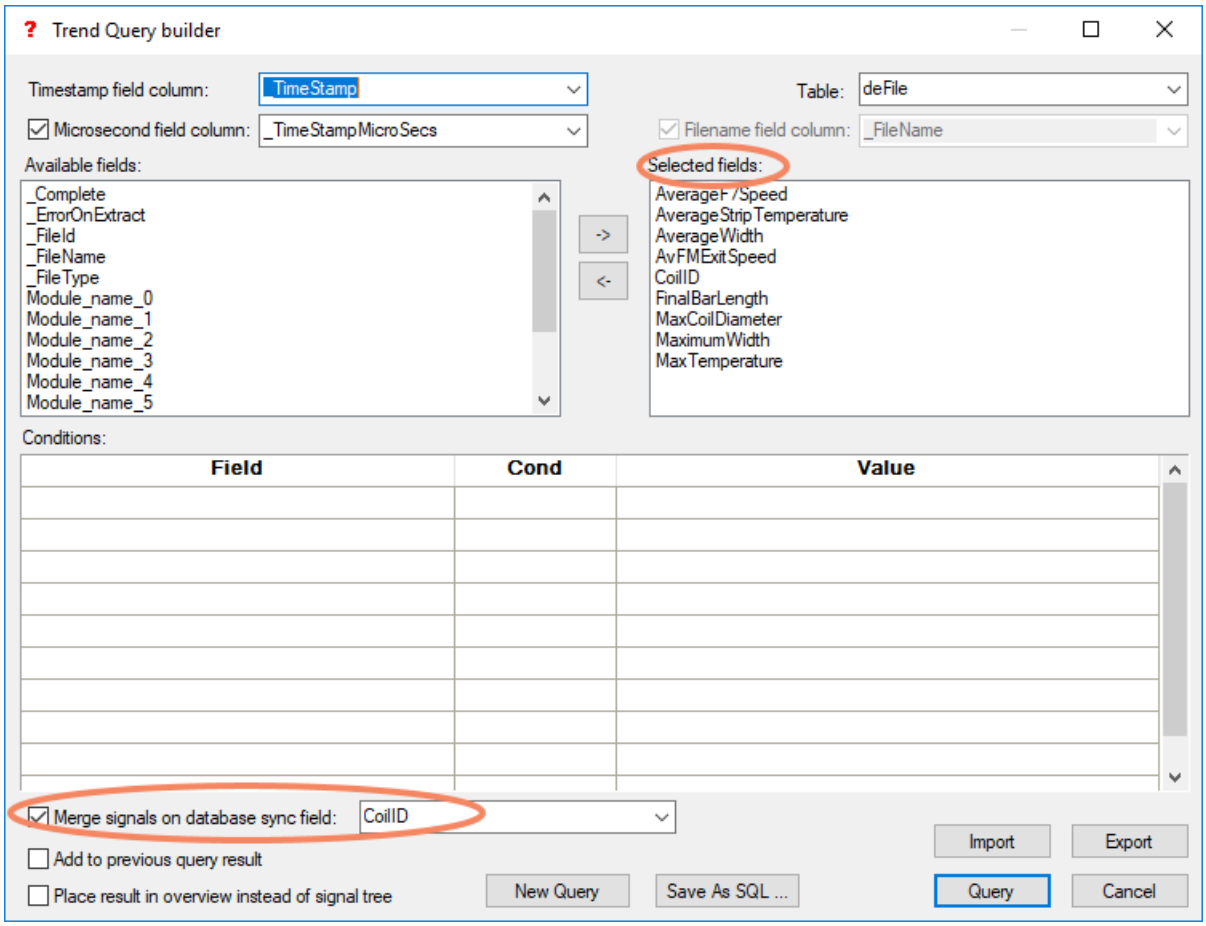
The fields required for the query can be selected / deselected.

❑ Conditions

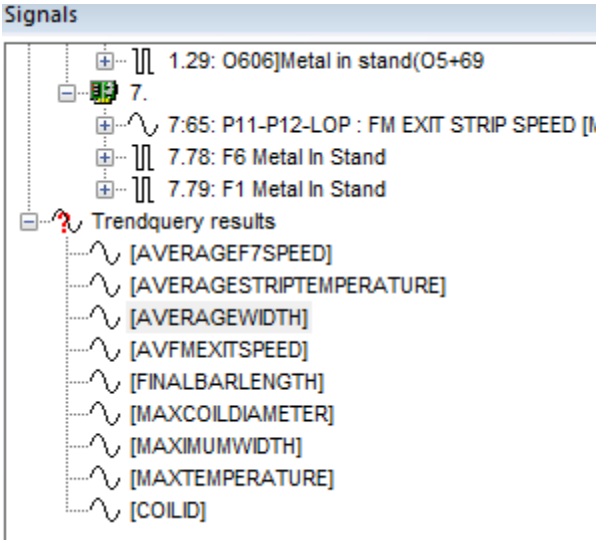
↗ See chapter: *Query builder*, page 67

❑ Merge signals on database sync field

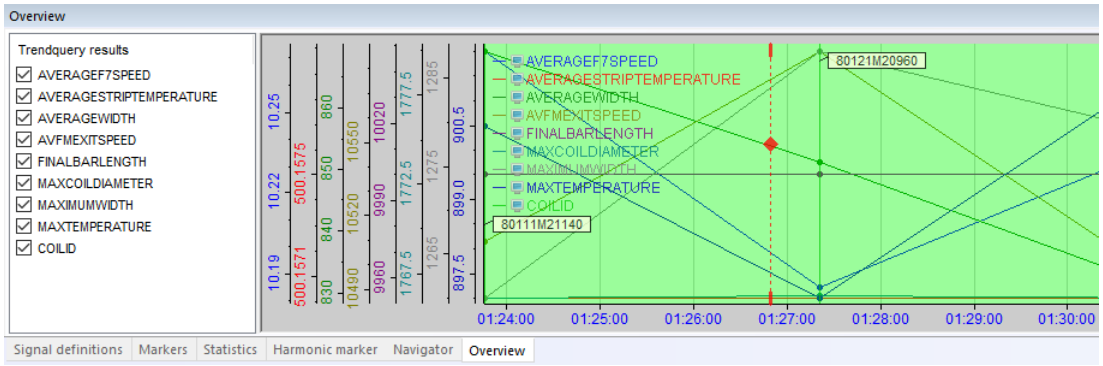
The Merge sync field function is also similar to the Merge function previously described for standard queries (see chapter: *Query builder*, page 67).



When selecting this option and selecting the name of the appropriate sync field column in the combo box next to it (e.g. CoilID), all rows with the same value in the sync field column will be merged and presented as a single data point in the trend query. The "Selected fields" will be displayed under "Trendquery results" in the "Signals" window



or in the "Overview" window.



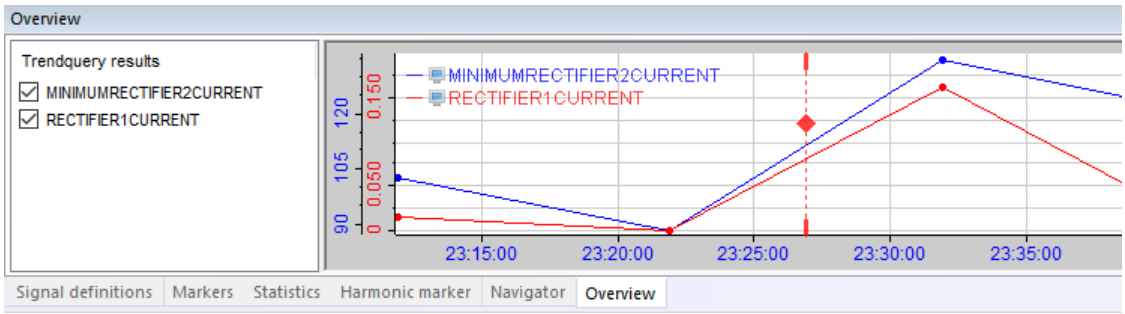
The timestamp of the data point will be the earliest in the merged rows and the values of the data point will be the first non 0 (Zero) values in the selected numeric or string columns when the merged rows are sorted by the timestamp (or the values will be 0 (Zero) if 0 (Zero) or NULL values are present in the merged rows for the selected numeric or string columns).

❑ Add to previous query result

↗ See chapter: *Query builder*, page 67

❑ Place result in overview instead of signal tree

This option puts the trend query result in the "Overview" window instead of the signal tree.



Note

Note that the option "Add to previous query result" is grayed out when this option is selected. This is because there can only be one query result (possibly comprised out of several signals) present at a time in the overview. Any previously executed trend query result in the overview will be replaced when continuing with the query.

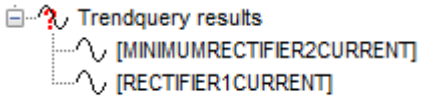
❑ <New Query>

↗ See chapter: *Query builder*, page 67

❑ <Save as SQL>

↗ See chapter: *Query builder*, page 67

❑ <Query>
Each field in the result set will generate a signal with a name which is the same as the fieldname. The signal will be added to the signal tree under the "Trend query results" node.



❑ <Cancel>

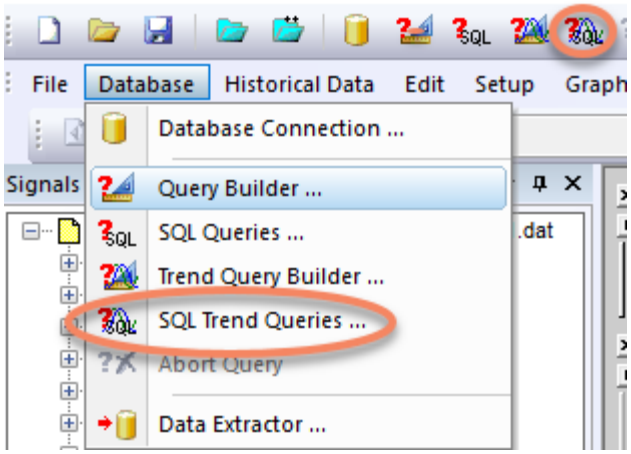
↗ See chapter: *Query builder*, page 67

❑ <Import>
Exported Trend Query Builder settings can be imported. The current settings are overwritten.

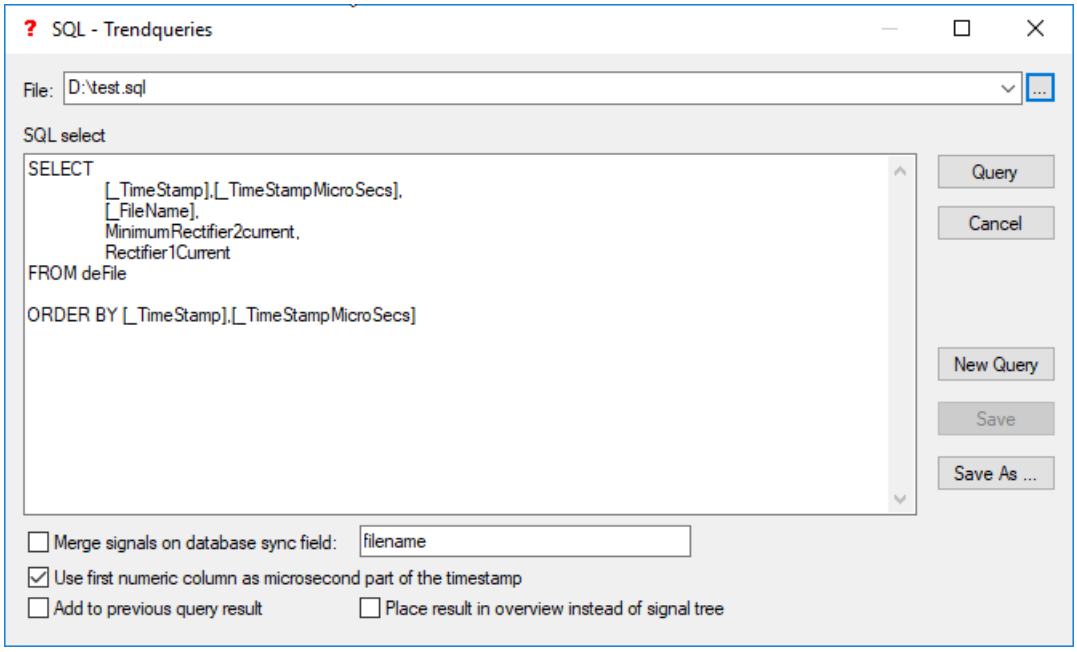
❑ <Export>
Trend Query Builder settings are exported as an .ini file.

7.2.2 SQL trend queries

Trend queries can be written in the same way as described in the chapter: *SQL-queries*, page 76.



SQL Trend Queries opens the following window.



SQL instructions can be entered into the "SQL-Trendqueries" window in order to acquire the trend graph data.

- File

↗ See chapter: *SQL-queries*, page 76

- SQL select

↗ See chapter: *SQL-queries*, page 76

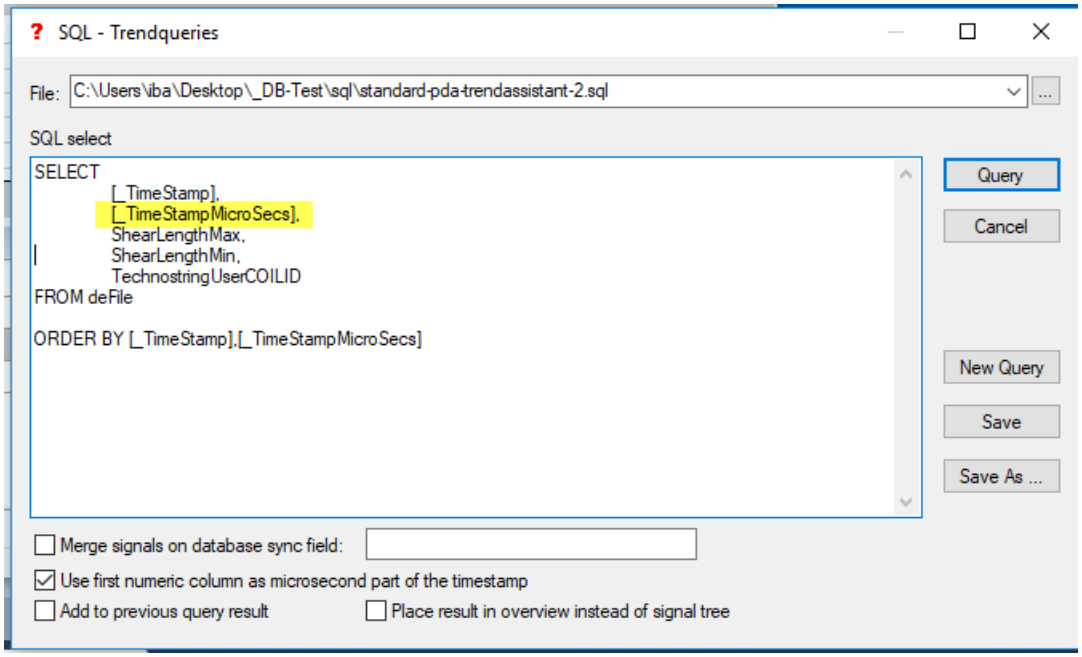
- Merge signals on database sync field

↗ See chapter: *Query builder*, page 67

- Use first numeric column as microsecond part of timestamp

Note

Obviously this column should contain the correct data type, otherwise the result will be meaningless



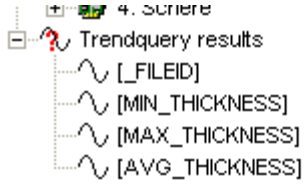
- <Add to previous query result>
  - ↗ See chapter: *Query builder*, page 67
- <Place result in overview instead of signal tree>
  - ↗ See chapter: *Query builder*, page 67
- < Query>
  - ↗ See chapter: *Trend query builder*, page 78
- <Cancel>
  - ↗ See chapter: *Query builder*, page 67
- <New Query>
  - ↗ See chapter: *Query builder*, page 67
- <Save>
  - ↗ See chapter: *SQL-queries*, page 76
- <Save As>
  - ↗ See chapter: *SQL-queries*, page 76

When executed, the instruction must have a result set with a time stamp field (if there are more than one time stamp columns only the first one will be referenced, the others will be displayed) and at least one numeric field. The instruction must also contain an "Order by" clause on the timestamp field.

7.3 Trend query results

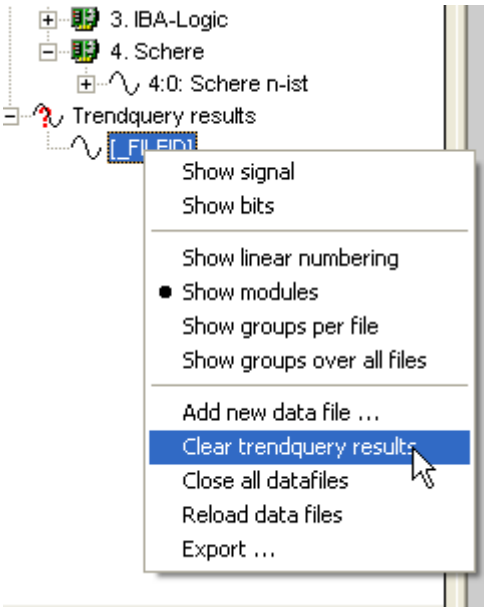
In addition to the standard query results, trend query results can also be displayed and evaluated using new functions such as overview display and drill down.

7.3.1 Trend query signal tree

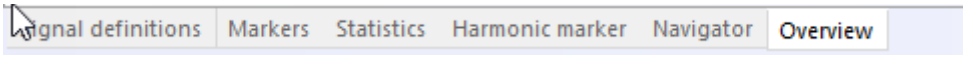


The results generated by the queries can be seen in the signal tree under the "Trend query results" node. These can be dragged or double-clicked to display them in the graph view.

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 drop down menu. The query results can also be cleared by selecting "Close all data files". This can also be done from the "File" drop down menu. A query result in the "Overview" window can only be erased by a new query.



7.3.2 Trend queries in overview



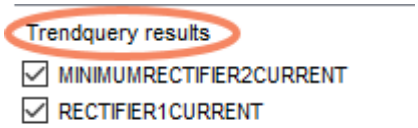
The trend queries can be shown in the "Overview" window. Calculations cannot be performed on trend queries placed in this window. However the "Overview" window has functions which are not available for trend queries placed in the main graph window. Specifically, the overview can be used to query the database for the extracted or source files that correspond to the data points of the trend query result in the overview and open the original data files corresponding to these extracted files (drill down functionality).

7.3.2.1 Trend query result list and graph

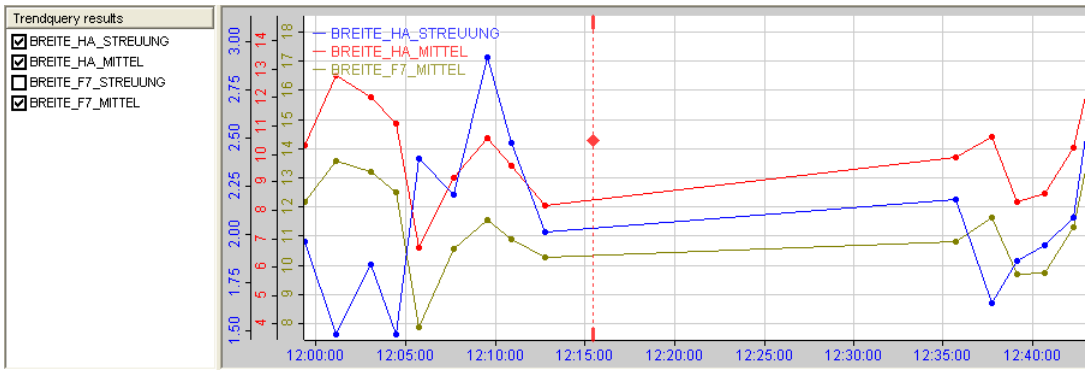
The overview contains two parts:

- A graph on the right side to depict the trend query results
- A list of the signal names available in the trend query result on the left side.

This list contains all column names resulting from the trend query. If the sync field was numeric, it will also be available here. Unchecking the checkboxes next to the names will mask the signals in the graph on the right. Selecting the result at the top of the list and clicking on "Trendquery results" causes the first checkbox state to be inherited by the remaining results. In this way multiple signals can be masked or unmasked together.



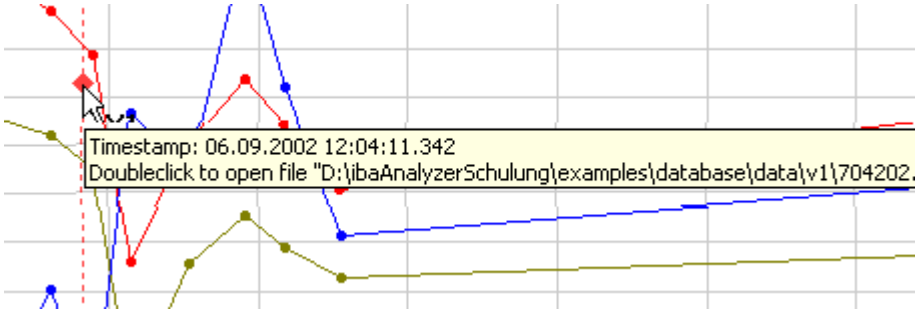
The graph that depicts the signals resulting from the trend query result is very similar to the standard *ibaAnalyzer* graph. Signals can be dragged and dropped on to the same Y-axis within the graph (but not out of the graph). The zoom buttons in the *ibaAnalyzer* toolbar will also work, provided that the overview is the active window (i. e. the last clicked window).



7.3.2.2 Markers

A pair of markers is available in the overview graph. These are like the normal markers except they have a diamond located at roughly 2/3rd of their height. These markers can be dragged by either the thick outer ends or the diamond. Pressing the <Ctrl> key while dragging the markers causes them to jump to the next data point.

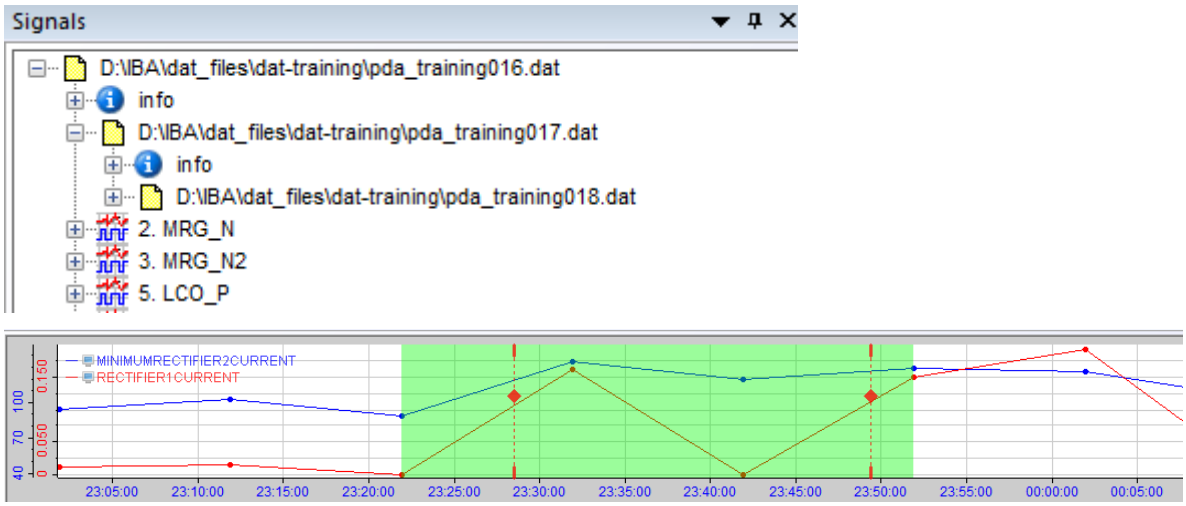
A tool-tip will appear when the mouse hovers over a marker's diamond containing the exact timestamp of the data point it is on and the action it will perform if you double click on the diamond. Depending on the settings this will either open the original file or query the database for the extracted file.



Pressing the <Ctrl> key while double clicking opens or queries all the files encompassed by the markers and appends them.

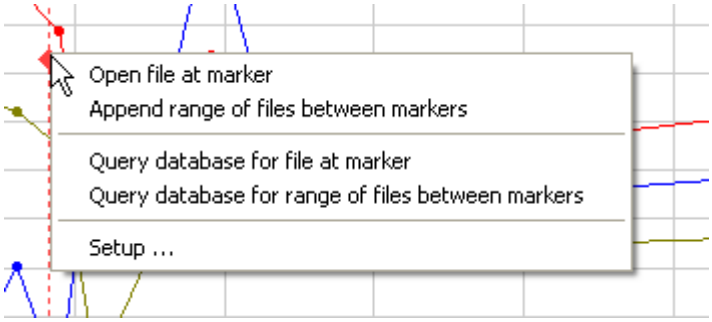
Note

The area to the right of a data point "belongs" to the file of that data point.



The time range visible in the graph window is highlighted by the green rectangle in the overview window.

A right click on the diamond opens a drop down menu of drill down options.



❑ Open file at marker

Open the original file corresponding with the data point, this option will be grayed out if the "Filename field column" was not selected 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) encompassed by the markers. Again this option will be grayed out if the "Filename field column" was not selected in the query dialog.

Note

The area to the right of a data point "belongs" to the file of that data point.

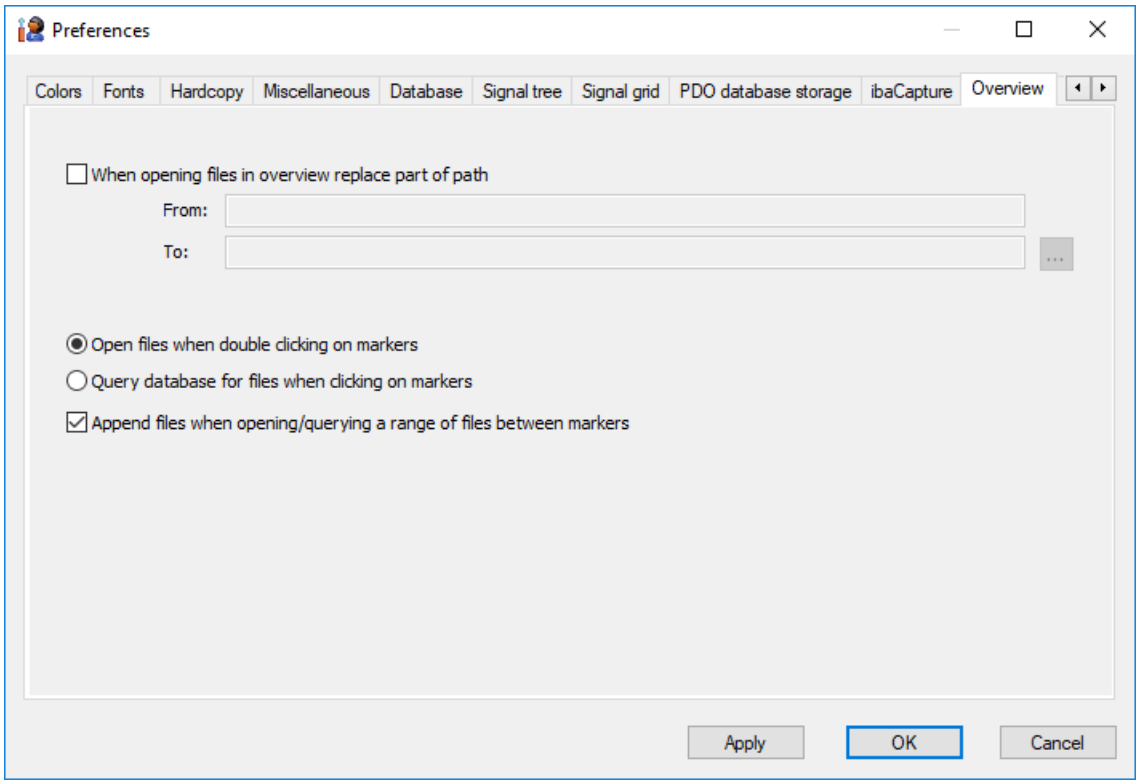
❑ <Query database for file at marker>

Instead of opening the original file, the extracted file in the database will be queried.

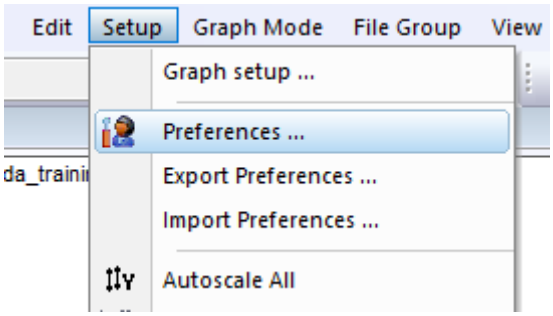
❑ <Query database for range of files between markers>

Similar to "Append range of files between markers" but will query and append the extracted files instead.

❑ <Setup …>
Will open the "Preferences" dialog with the tab showing the options for the overview selected.



This tab can also be opened by



7.3.2.3 Overview options

The following options are available in the *Setup – Preferences - Overview* window.

❑ When opening files in overview replace part of path

If the original data files have been moved to another location, the entire path or part of the path (e.g. from local pda computer) can be replaced with the full address of a file server from the filename column. A browse button is available to select the replacement path.

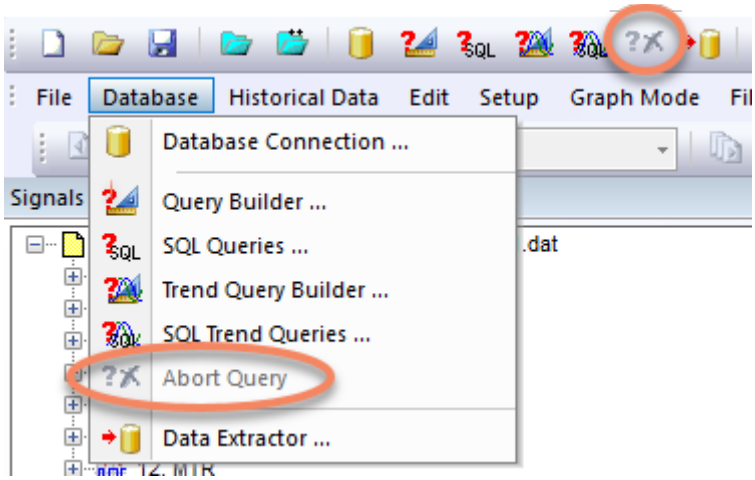
❑ Open files when double clicking on markers> or <Query database for files when double clicking on markers

The default behavior when double clicking on the marker diamond can be selected. You can select either the original files will be opened or the database will be queried for the extracted files.

❑ Append files when opening/querying a range of files between markers

If this is enabled all the selected files are opened immediately and appended to each other. If disabled only, the first file will be opened.

7.4 Abort query



Abort Query terminates any currently running queries.

8 Database introduction and basic handling techniques

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

8.1 Overview

A database is an organized collection of data, typically stored in electronic format.

- Data can be input, managed, organized, and retrieved quickly.
- Traditional databases are organized by records (rows), fields (columns) located in tables which are stored in the database files.

Relational Database Server Goals

A database table is a collection of rows and columns that is used to organize information about a topic. Each row within a table corresponds to a single record and contains several attributes that describe the record.

These tables are stored in databases

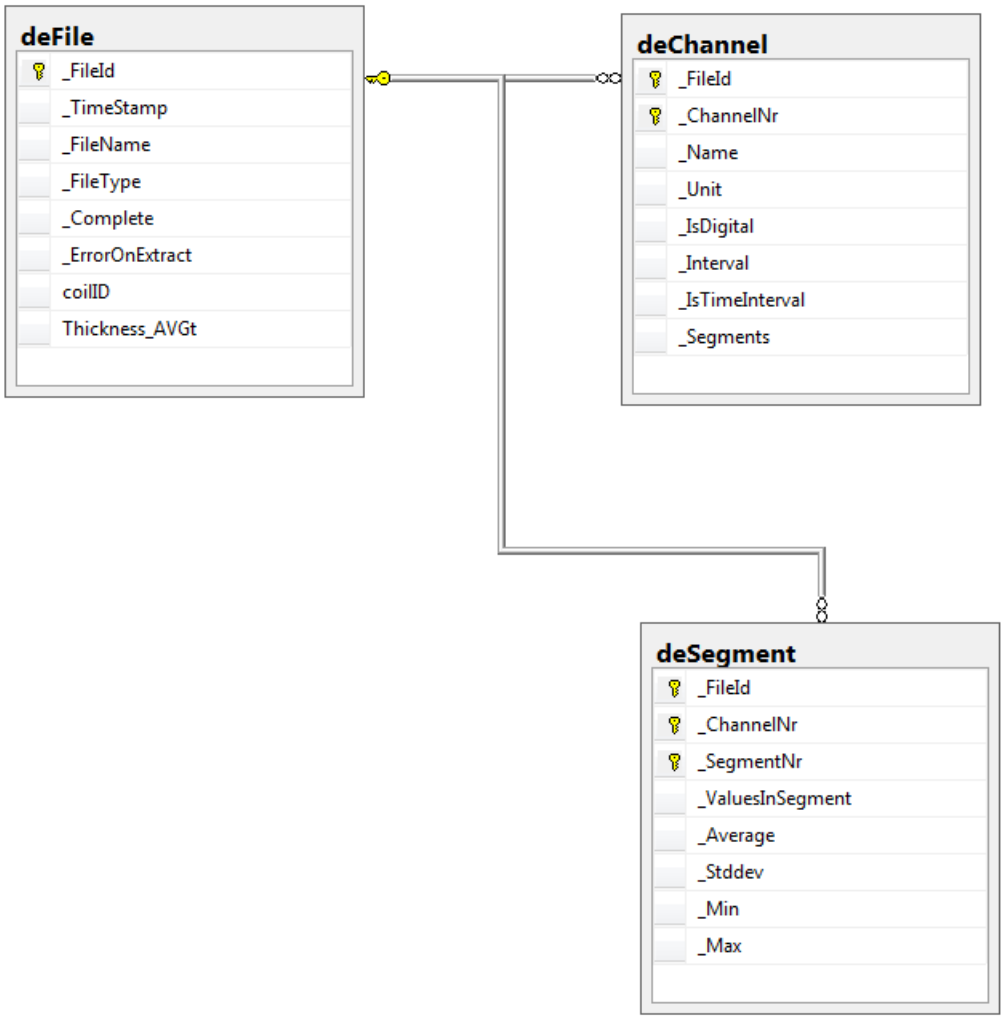
	_FileId	_TimeStamp	_FileName	_FileType	_Complete	_ErrorOnExtract	coilID	Thickness_AVGt
1	1218089036	2008-12-17 10:52:40.000	C:\iba\training\dat\QDR\471100017.dat	QDR2	1	0	471100017	1.08266
2	1218354195	2008-12-17 10:47:40.000	C:\iba\training\dat\QDR\471100015.dat	QDR2	1	0	471100015	0.959364
3	1218732091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

Data types - built-in data type categories

- Exact numerics – (bigint, bit, decimal, int, money, numeric, smallint)
- Approximate numerics (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, timestamp, uniqueidentifier, table)
- Large valued data types (varchar(max), nvarchar(max))
- Large object data types (text, ntext, image, xml)

A relational database is a collection of data tables all of which are formally described and organized according to the relational model. Each table must identify a column or group of columns, by means of the PRIMARY KEY, in order to uniquely identify each row.

ibaAnalyzerDB creates the following types of tables (see chapter: *Database format options*, page 56)



Referential Integrity (RI) is a database concept used to ensure that the relationships between the database tables remains synchronized during data modifications. RI can be used to ensure the data is consistent, may be helpful in optimizing the database environment and can assist in 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 referencing a primary key constraint, a foreign key can also reference a UNIQUE constraint to help maintain referential integrity.

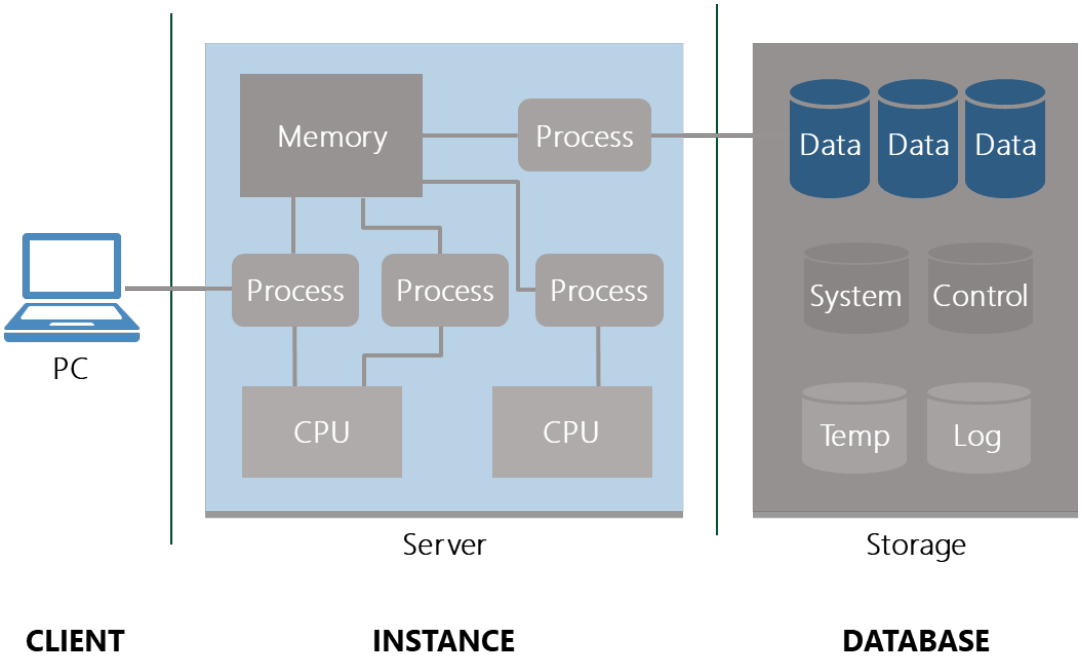
Normalizing a database

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

General structure in database management systems

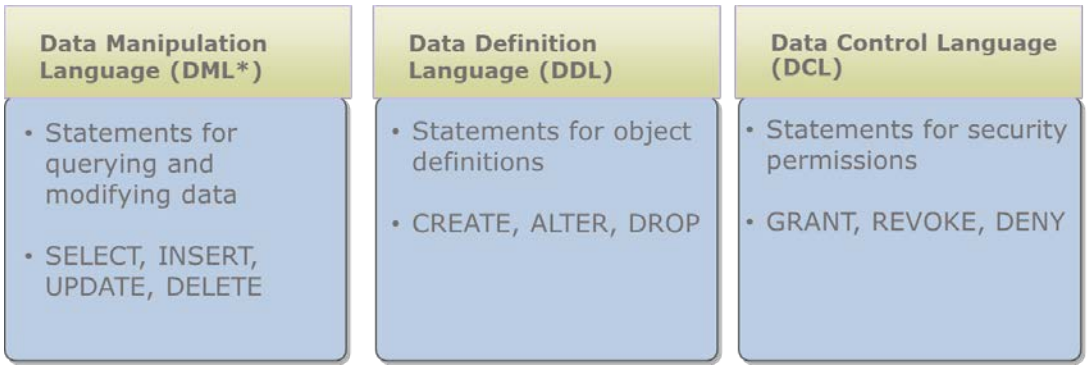


8.2 Introduction to SQL language

SQL; Structured Query Language is a special-purpose programming language designed for managing data held in a relational database management system (RDBMS).

- SQL was initially developed at IBM in the early 1970s.
- In the late 1970s, Relational Software, Inc. (now Oracle Corporation) developed its own SQL-based RDBMS with the purpose of selling it to the U.S. Navy, Central Intelligence Agency, and other U.S. government agencies.
- In June 1979, Relational Software, Inc. introduced the first commercially available implementation of SQL, Oracle V2 (Version2) for VAX computers.

Categories of SQL statements



*DML with SELECT is the primary focus of this description

Elements of the SELECT statement

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

The following examples are based on the MS SQL server syntax.
Other databases (Oracle, IBM-DB2, etc.) may have different SQL syntaxes.

8.2.1 SELECT statements

Values:

```
SELECT coilID, Thickness_AVGt
  FROM deFile
```

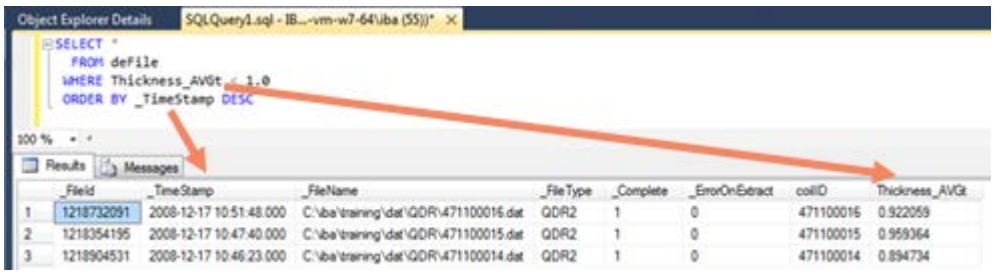
100 %

Results

```
coilID     Thickness_AVGt
---------  --------------
471100017  1.08266
471100015  0.959364
471100016  0.922059
471100014  0.894734

(4 row(s) affected)
```

Conditions and sort:



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

8.2.2 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

8.2.3 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

8.2.4 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 n: Select top n files from a table and sort them ASCending or DESCending

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

```
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: display results according to set conditions

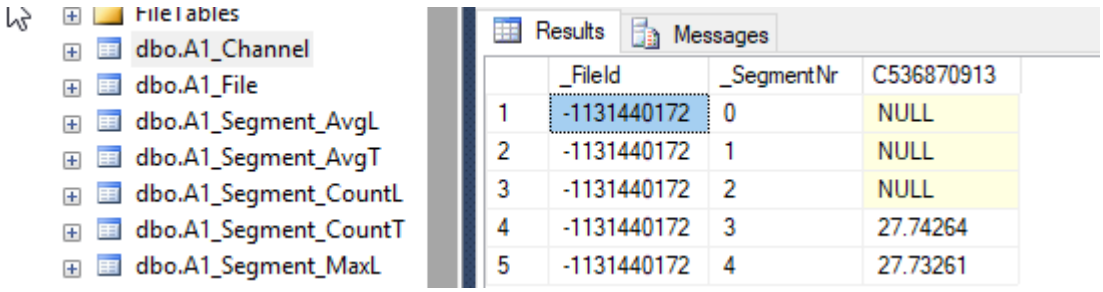
```
select [_FileName],
       [MinimumRectifier2current],
       CASE WHEN  [MinimumRectifier2current] < 105
            THEN 0
            ELSE [MinimumRectifier2current]
        END AS "Current less than 105"
   from [IBA].[dbo].[deFile]
```

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

8.2.5 NULL handling



Select top 2 "NULL" segments:

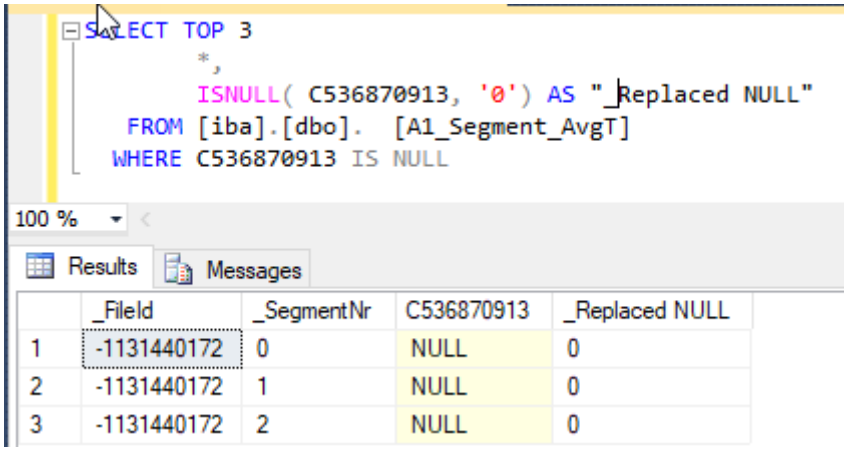
```
SELECT TOP 2
       *
  FROM [iba].[dbo].[A1_Segment_AvgT]
 WHERE C536870913 IS NULL
```

100 %

Results Messages

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

Replace NULL with 0:



```
SELECT TOP 3
       *,
       ISNULL( C536870913, '0') AS "_Replaced NULL"
  FROM [iba].[dbo].  [A1_Segment_AvgT]
 WHERE C536870913 IS NULL
```

100 %

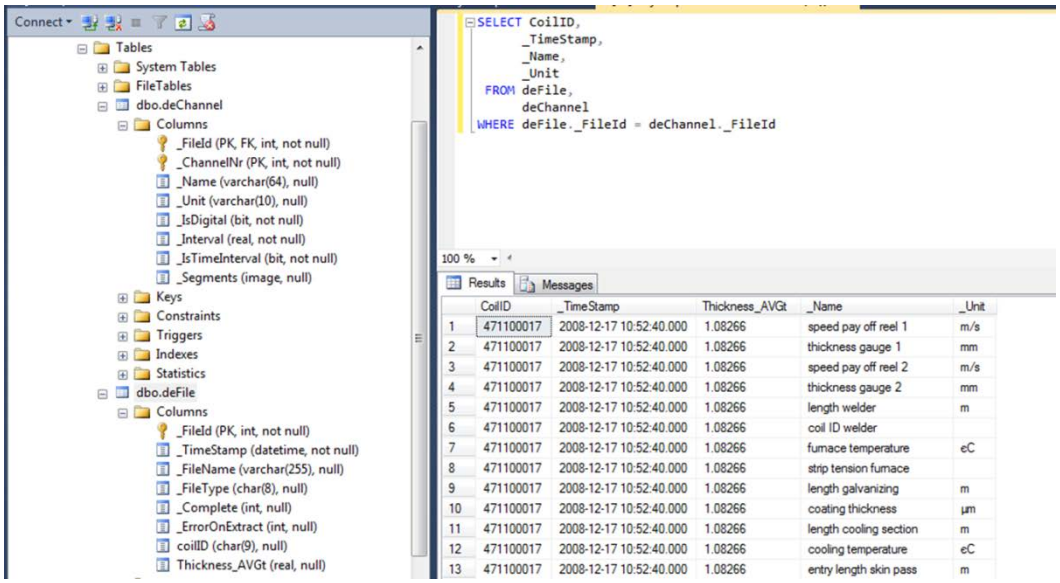
Results | Messages

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

8.2.6 JOIN clause

The JOIN clause the combination of related data from multiple tables into one result set

Select data from two tables with a common _FileId.



Connect

- Tables
  - System Tables
  - FileTables
  - dbo.deChannel
    - Columns
      - _FileId (PK, FK, int, not null)
      - _ChannelNr (PK, int, not null)
      - _Name (varchar(64), null)
      - _Unit (varchar(10), null)
      - _IsDigital (bit, not null)
      - _Interval (real, not null)
      - _IsTimeInterval (bit, not null)
      - _Segments (image, null)
    - Keys
    - Constraints
    - Triggers
    - Indexes
    - Statistics
  - dbo.deFile
    - Columns
      - _FileId (PK, int, not null)
      - _TimeStamp (datetime, not null)
      - _FileName (varchar(255), null)
      - _FileType (char(8), null)
      - _Complete (int, null)
      - _ErrorOnExtract (int, null)
      - coilID (char(9), null)
      - Thickness_AVGt (real, null)

```
SELECT CoilID,
       _TimeStamp,
       _Name,
       _Unit
  FROM deFile,
       deChannel
 WHERE deFile._FileId = deChannel._FileId
```

100 %

Results | Messages

	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

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

```
SELECT TOP 50000
       deFile.CoilID,
       deFile._TimeStamp,
       deChannel._Name,
       deChannel._Unit,
       deSegment._SegmentNr,
       deSegment._Average,
       deSegment._Min,
       deSegment._Max
  FROM deFile,
       deChannel,
       deSegment
 WHERE deFile._FileId = deChannel._FileId
   AND deFile._FileId = deSegment._FileId
   AND deChannel._ChannelNr = deSegment._ChannelNr
   AND coilID = '471100017'
```

100 %

Results | Messages

	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

8.2.7 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

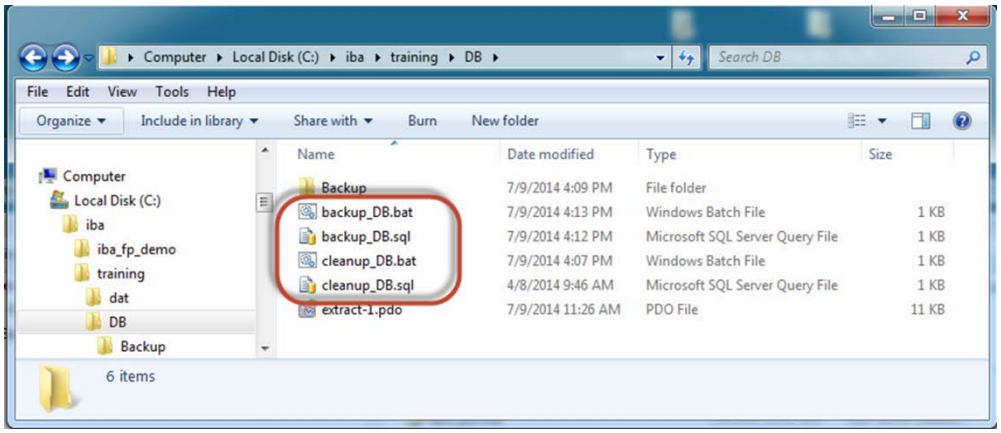
	CoilID	_TimeStamp	_Name	_Unit	_ChannelNr	# Segments	Avg Avg	Min Min	Max Max
1	471100014	2008-12-17 10:46:23.000	rolling force AS	N	-2142896007	72	23.6523003458149	0.001651668	29.51965
2	471100014	2008-12-17 10:46:23.000	rolling force BS	N	-2142896006	72	23.8890460899307	0.001166938	29.93768
3	471100015	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

8.3 SQL-scripting, 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 and are terminated with the GO clause.

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

8.3.1 Useful applications (backup & cleanup):



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
```

9 Command Line Options

9.1 Starting ibaAnalyzer

If *ibaAnalyzer* has been installed using the installation wizard, the simplest method of starting the program is by double-clicking the icon on the desktop or opening a data or analysis file.

ibaAnalyzer can also be started using a command line. This means that the program can also be started via batch/windows scripts or from within other programs, such as ibaDatCoordinator, ibaPDA, ibaLogic or customer applications.

A special option is available when the program is started via the command line. This permits different parameters to be added which instruct *ibaAnalyzer* to perform particular analyses, print reports, cyclically refresh the display with every new data file or write into/read from a database.

The ibaDatCoordinator was developed for automated processing in distributed environments. ibaDatCoordinator enables easy configuration of standard tasks such as database extraction, report generation or data file archiving and can handle several typical fault or maintenance events such as network problems or server maintenance stops. Advanced processing tasks such as automated report generation based on overview and trend queries can also be implemented using ibaDatCoordinator by means of so called "script tasks". In these scripts (e. g. *.bat or *.vbs), the behavior of *ibaAnalyzer* is configured by command line switches.

(Please refer to your operating system’s documentation to learn more about the usage and functions of the command line)

9.2 Command Line Syntax

```
ibaAnalyzer.exe dat1 […datn] [pdo] [/switch1] … [/switchm]
```

One or more data files (data file name), an analysis (pdo file name) and one or more switch parameters (switches) can be included in the program call. Although it is possible to use the relative path, it is recommended to always use the complete path for every file.

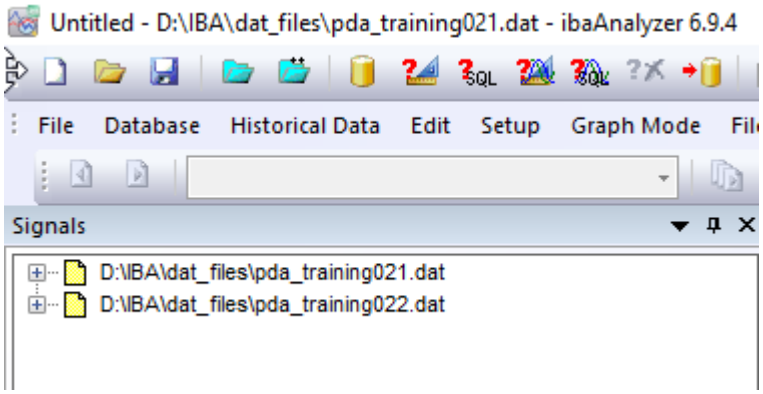
Note

Any paths containing blanks need to be in quotation marks.

If only data files are specified, *ibaAnalyzer* opens all the data files and shows them in the signal tree.

Example:

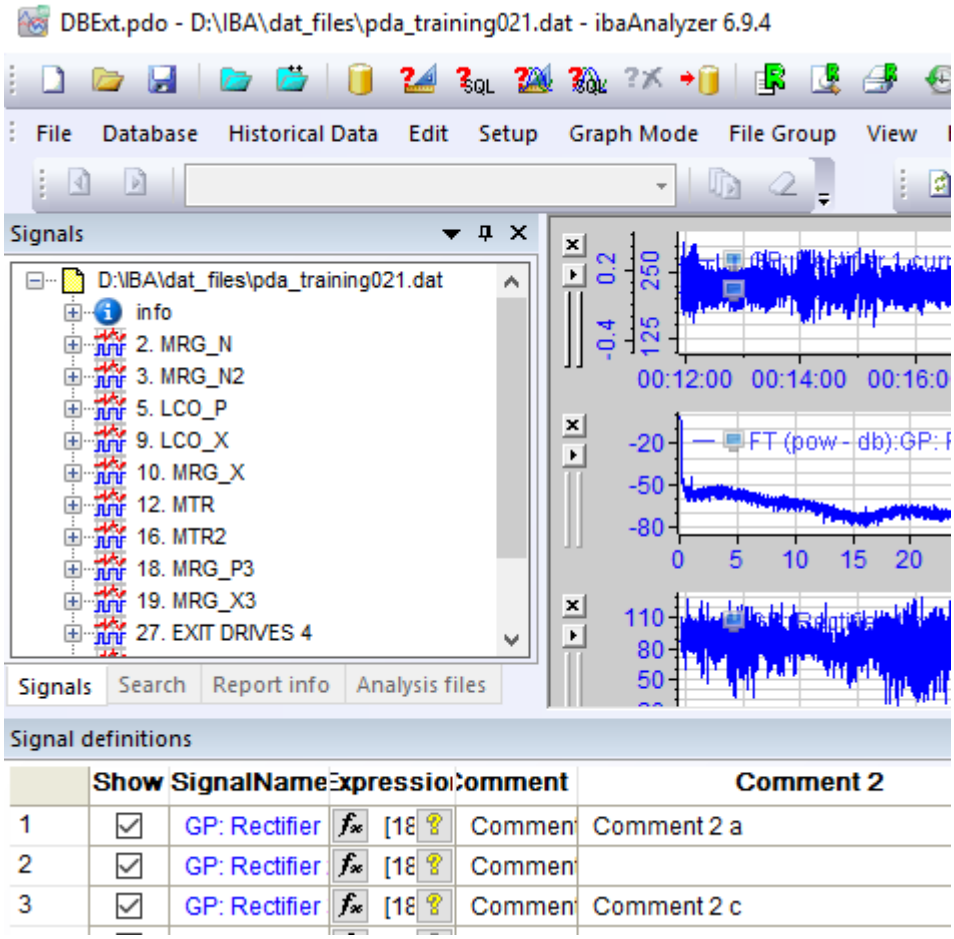
"C:\Program Files (x86)\iba\ibaAnalyzer\ibaAnalyzer.exe"
D:\IBA\dat_files\pda_training021.dat D:\IBA\dat_files\pda_training022.dat



If an analysis file is also specified, *ibaAnalyzer* opens the data files and shows the signals as defined in the analysis file. If more than one analysis file is specified, only the last analysis file in the command-line is loaded.

Example:

"C:\Program Files (x86)\iba\ibaAnalyzer\ibaAnalyzer.exe" D:\IBA\dat_files\pda_training021.dat D:\IBA\DBExt.pdo



9.3 Command line switches

The switches are particularly important in conjunction with further actions because they can be used to automatically complete analysis tasks. It is, however, also possible to use the switches in conjunction with a manual program start.

9.3.1 "/reuse" switch

If this switch is included in the program call, *ibaAnalyzer* starts, loads the specified data files and, if applicable, displays the results as determined by an analysis file.

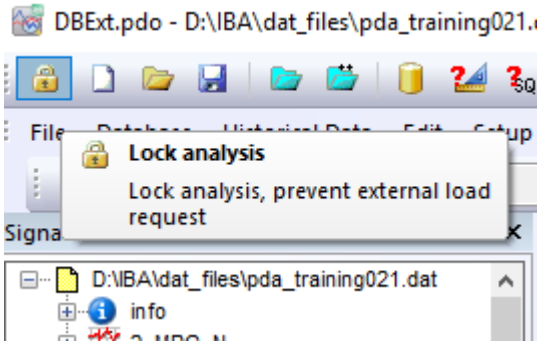
Every new program call containing the /reuse switch keeps the currently running instance of *ibaAnalyzer* open, but closes previously opened data files before opening the files specified in the current call.

Example:

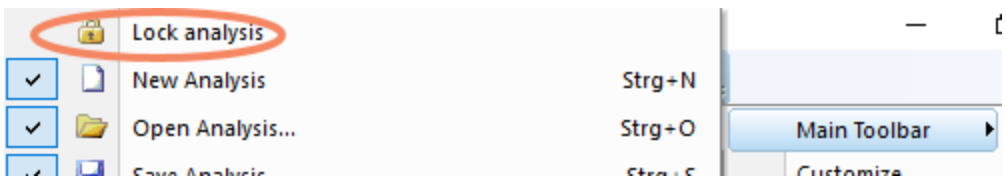
"C:\Program Files (x86)\iba\ibaAnalyzer\ibaAnalyzer.exe"
D:\IBA\dat_files\pda_training021.dat D:\IBA\DBExt.pdo /reuse

A "Lock analysis" icon in the upper left corner of the tool bar indicates that *ibaAnalyzer* has been started with the /reuse switch.

Clicking this button disables updates by further program calls containing the /reuse switch. By automating this process, for example, using the post-processing command, it is possible to permanently update an analysis display with the latest .dat file.



As this icon is not displayed by default it must be activated via the Main Toolbar selection.

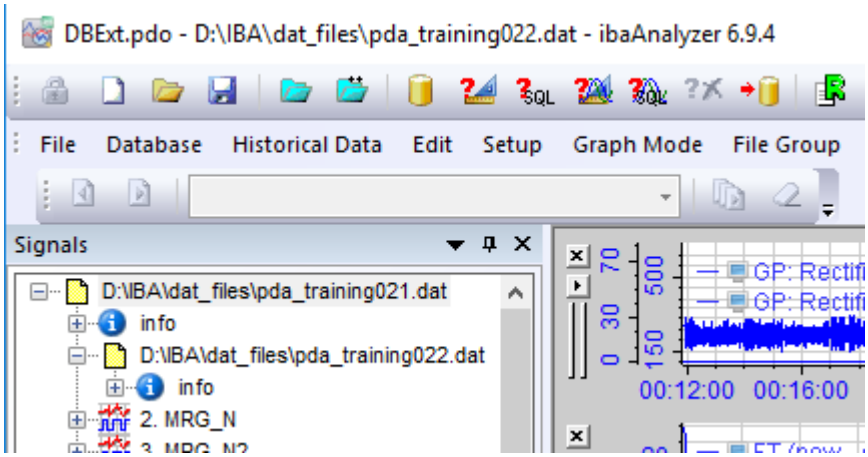


9.3.2 "/append" switch

This switch enables the appending of several data files specified in the call. These files are then displayed contiguously in the graph's X direction. An example is given below.

Example:

"C:\Program Files (x86)\iba\ibaAnalyzer\ibaAnalyzer.exe" D:\IBA\dat_files\pda_training021.dat D:\IBA\dat_files\pda_training022.dat D:\IBA\DBExt.pdo /append



9.3.3 "/print" switch

This switch causes the measurement data to be printed as a record or log in the format defined in the selected analysis. The Windows default printer is used.

Example: C:\ibaAnalyzer.exe C:\dat\pda040.dat C:\iba\speed.pdo /print

ibaAnalyzer is closed after the printing process has been completed or after the print job has been triggered,
In the case of an error, however, *ibaAnalyzer* remains open in order to display the error message.

9.3.4 "/extract[:"filename"]" switch

The extract switch means that *ibaAnalyzer* starts and loads the specified data file. Thereafter, the measurement data is processed in accordance with the specified analysis and extracted into a database. This has the same effect as pressing the <Extract now> button on the Data Extractor dialog. (see: chapter: Extractor output).

During this process, no *ibaAnalyzer* window is opened on the screen, i. e. the extraction process runs in the background. The database connection must have been configured previously in the analysis.

Examples:

```
C:\ibaAnalyzer.exe C:\dat\pda040.dat C:\iba\dbextract.pdo /extract
```

```
C:\ibaAnalyzer.exe C:\dat\pda040.dat C:\iba\txt.pdo
/extract:"c:\output.txt"
```

If "Extract to file" is selected in the analysis file then the name of the output file can be specified after the extract switch. The second example will extract the data specified in

txt.pdo from **pda040.dat** into a text file called: "**c:\output.txt**".

This switch can only be used in conjunction with the license for the database interface (ibaAnalyzer-DB-Extractor). You may also extract the data into a file by using the filename as parameter to the switch. For extracting data into a file the license *ibaAnalyzer-DAT-Extractor* is required.

9.3.5 "/sql:filename.sql[;sync:"syncFieldName"]" switch

This switch is used for database queries. The "file name.sql" argument can be used to transfer SQL statements as a basis for the database query. The additional, optional [;sync:...] parameter can be used to specify a merge criterion for the query data.

Examples:

```
C:\ibaAnalyzer.exe C:\iba\dbquery.pdo /sql:"C:\iba\query.sql"
```

```
C:\ibaAnalyzer.exe C:\iba\dbmerge.pdo /sql:"C:\iba\merge.sql";
sync:"STRIP_ID"
```

The "dbquery.sql" must be a text file conforming to the SQL-language supported by the database system specified in the PDO file (i. e. Oracle, SQL-Server, DB2, ODBC)

For instance the query.sql specified in first example could be something like:

```
SELECT * FROM PDA_File order by [_Timestamp] DESC;
```

The second example shows the use of the /-sql switch together with the sync field argument.

The sync field name (here "STRIP_ID") corresponds to the database sync field specified in the SQL-Queries dialog. (See chapter: *Query builder*, page 67).

When specifying data files and also using the /sql parameter, all data files are placed in the signal tree starting from the second line. If the query was successful the first position will contain the first file in the query result, otherwise the first line will be empty.

9.3.6 "/trendsql:filename.sql" switch

Examples:

```
C:\ibaAnalyzer.exe C:\iba\dbtrend.pdo /trendsql:"C:\iba\trend.sql"
```

```
C:\ibaAnalyzer.exe C:\iba\dbtrend.pdo /trendsql:"C:\iba\trend.sql";
sync:"SID"
```

This switch is similar to the previous one, but it queries infofields and/or computed columns from the file table instead of data files. (See chapter *Trend queries*, page 78

ibaAnalyzer starts and queries the database specified in the PDO with the SQL statement from the "trend.sql" file. This query needs to have a TimeStamp column selected. All other selected columns will be added under the "trend query results" node in the signal tree and can be used in the analysis.

The file trend.sql must be a text file conform to the SQL-language supported by the database system specified in the analysis file (i. e. Oracle, SQL-Server, DB2, ODBC).

The SQL instruction in it must have a result set with a timestamp field (you can have more than one, but only the first one will be referenced) and at least one numeric field. The instruction must also contain an "Order by" clause on the timestamp field.

9.3.7 "/overviewsql:filename.sql" switch

This switch is similar to the previous one. The difference is that it displays the query results in the overview window. (See chapter: *Trend queries*, page 78) Trend QueriesThe /sql switch can be combined with the /overviewsql switch. This enables starting *ibaAnalyzer* with an overview while also simultaneously querying files from the database. You need to specify separate SQL instructions for the overview and the file query in different text files in this case.

Example:

```
C:\ibaAnalyzer.exe /overviewsql:"C:\iba\overview.sql"
/sql:"C:\iba\query.sql"
```

9.3.8 "/report[:filename]" switch

With this switch, *ibaAnalyzer* starts, loads a specified data file and performs an analysis in accordance with the specified analysis rules. Thereafter, the integrated report generator is started and the data is printed on the Windows default printer using a report layout specified in the analysis rule if the [:file name] option was not used with the switch.

If the [:file name] switch option is used, the report can be written into a file rather than being printed. The desired file type is determined by the file name extension.

Many customary formats are supported, including, for example, .pdf, .htm, .rtf, .tiff, .jpg, .xls, etc.

Examples:

```
C:\ibaAnalyzer.exe C:\dat\pda040.dat C:\iba\rep.pdo /report
```

```
C:\ibaAnalyzer.exe C:\dat\pda040.dat C:\iba\rep.pdo
/report:"c:\repout.pdf"
```

Reports can also be based on database queries. After an standard or trend query it is possible to build a report with the selected data. The according *.sql and *.pdo can be used for a command line script.

Example:

```
C:\ibaAnalyzer.exe /sql:"C:\iba\merge.sql";sync:"STRIP_ID" 8
```

```
C:\iba\strip_rep.pdo /report:"c:\tmp\repout.pdf"
```

9.4 Combination of command line switches

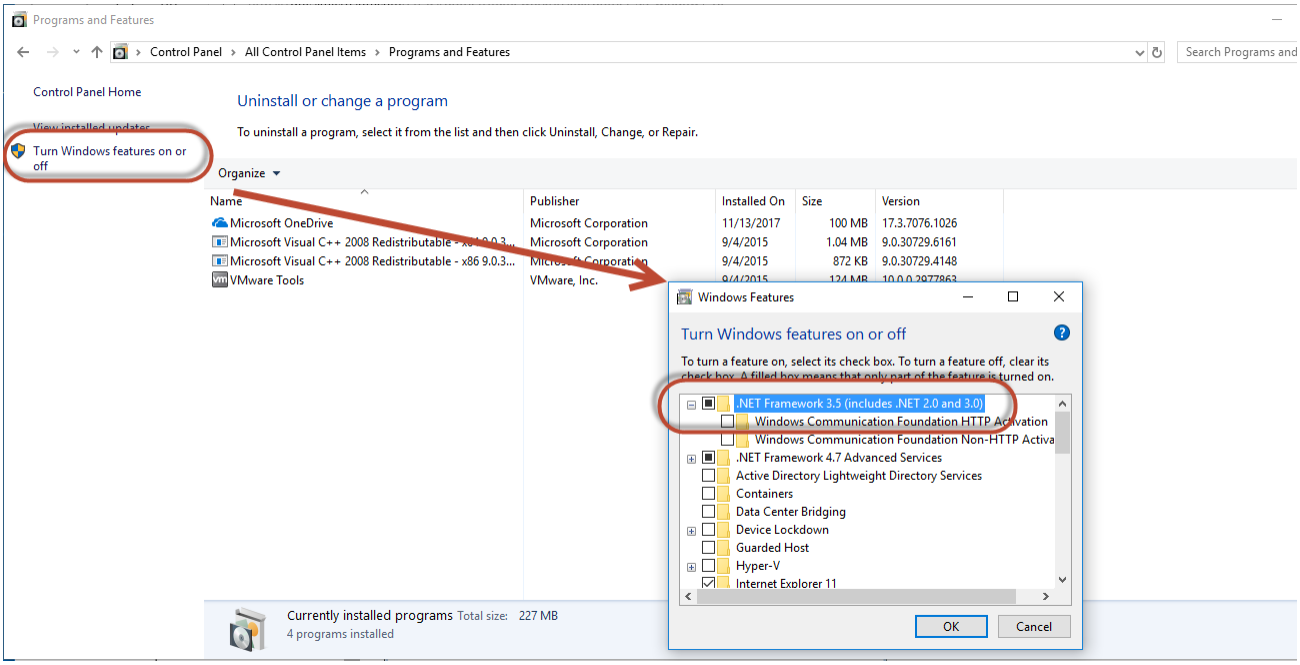
Not all command line switches can be combined. The following table gives an overview which switches can be combined:

Combination permissible or useful ?	/sql	/reuse	/append	/print	/extract	/report	/trendsql /overviewsql
/sql							
/reuse	YES						
/append	YES	YES					
/print	YES	NO	YES				
/extract	NO	NO	YES	YES			
/report	YES	NO	YES	YES	YES		
/trendsql /overviewsql	YES	YES	NO	YES	NO	YES	

10 Setting up the database

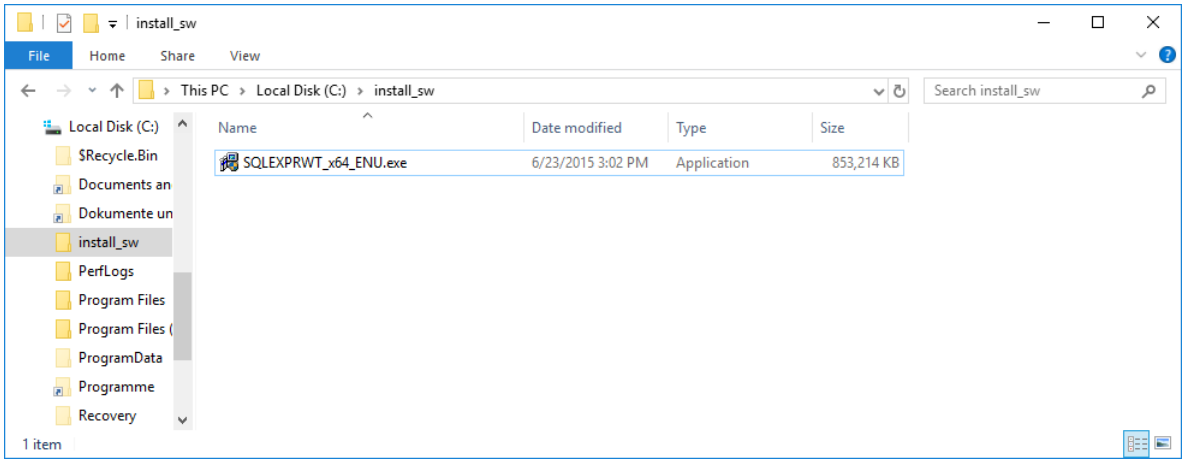
Several databases are supported by Microsoft OLE-DB API: SQL-Server, Oracle, DB2-UDB or by ODBC: MySQL, PostgreSQL, SQLite, MS Access, etc. The following describes the steps required to set up the required database software for MS SQL Server Express 2014 with Tools 64bit (SQLEXPRESSWT_x64) which could be downloaded from Microsoft website.

Before starting the Installation, ensure that the required .Net framework 3.5 is available:

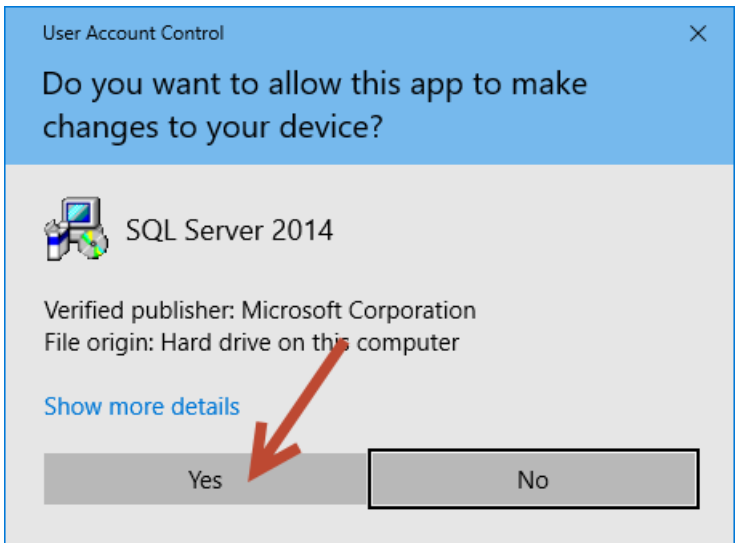


10.1 Start software installation

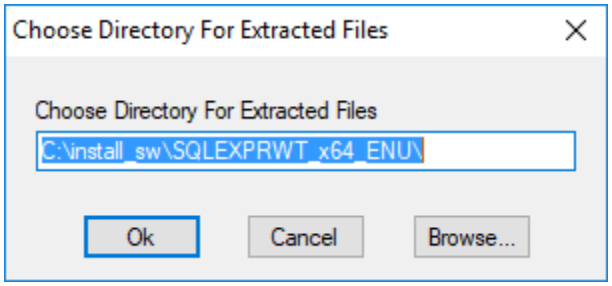
Start Install package:



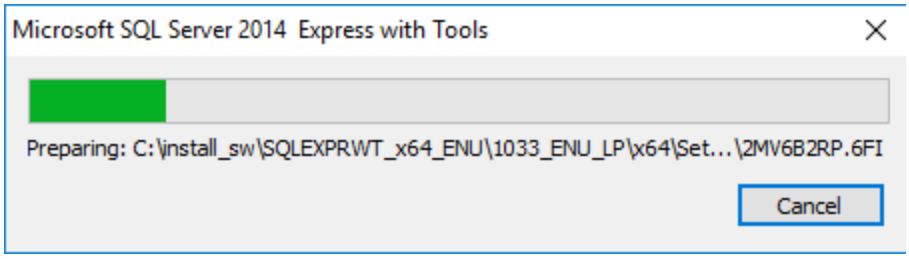
Allow installation and system changes:



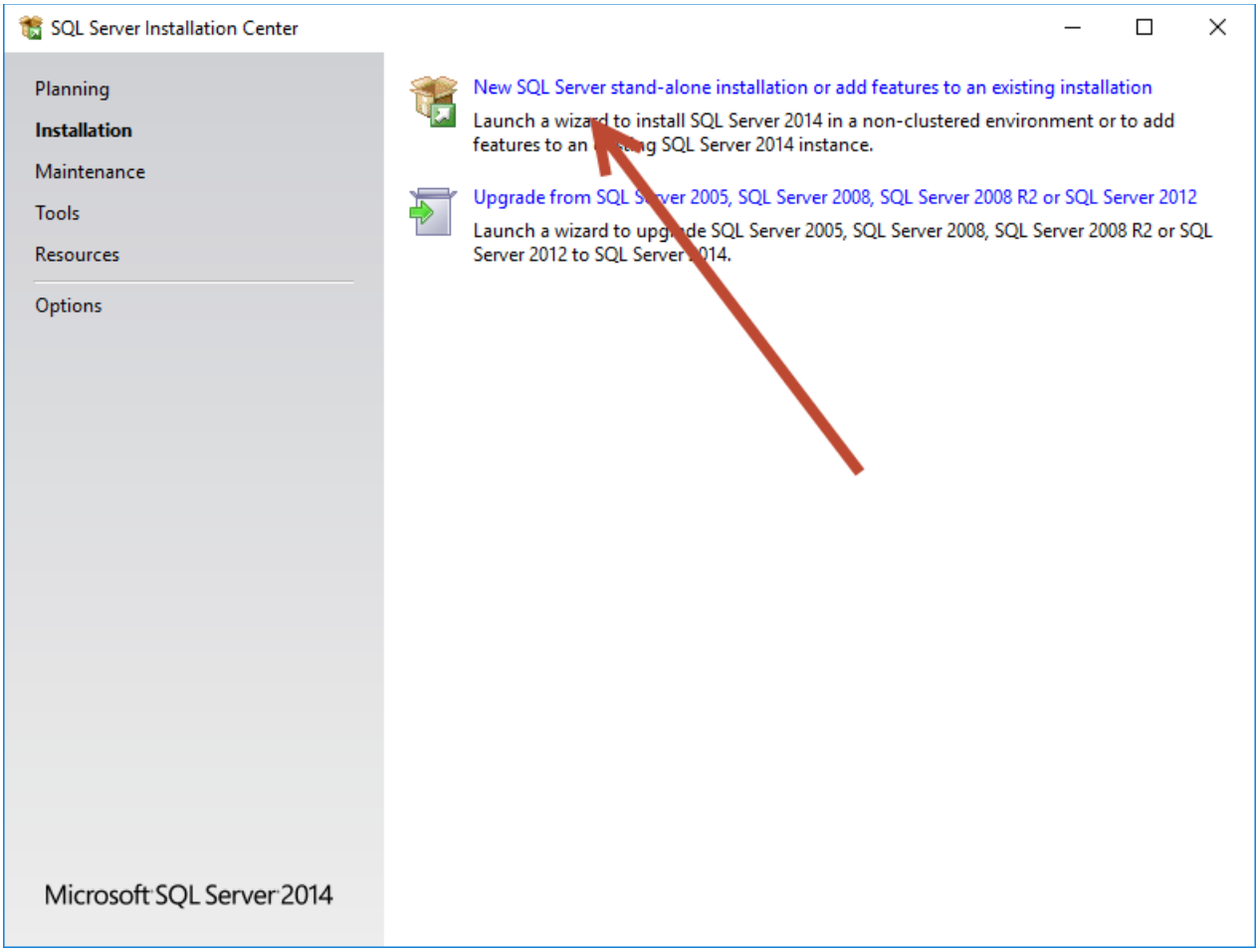
Select directory for temporary sw extract:



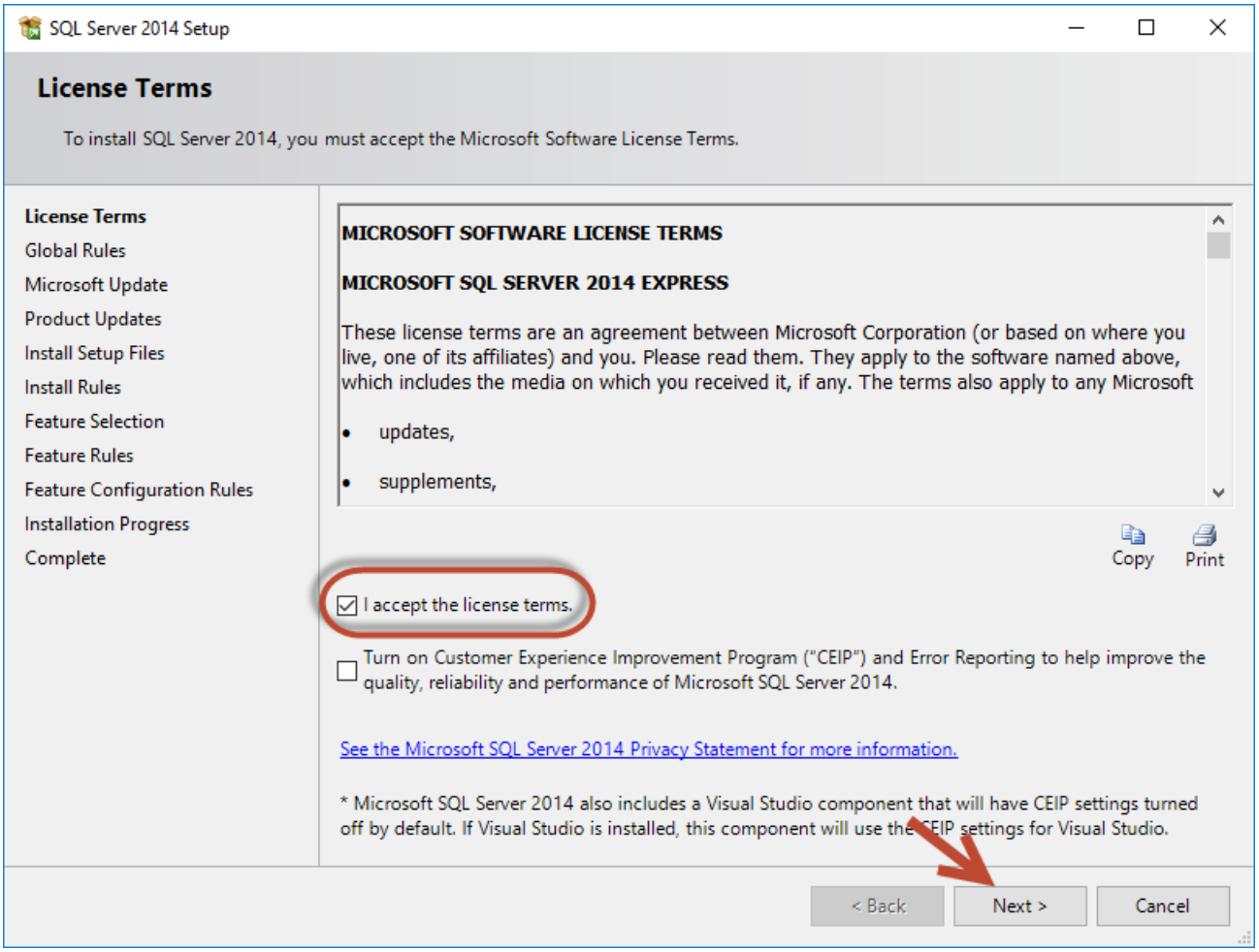
… extract process…

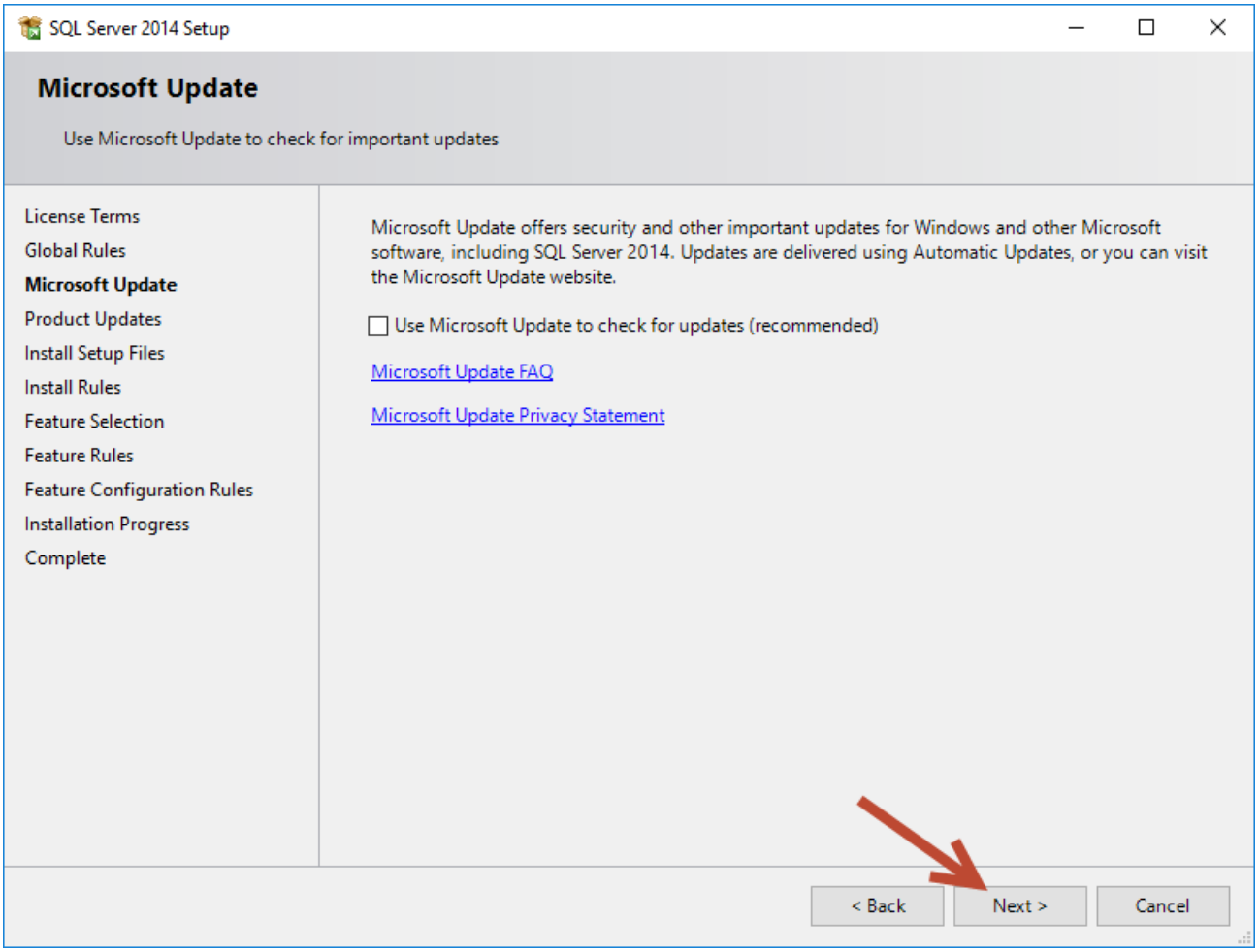


Select new installation:

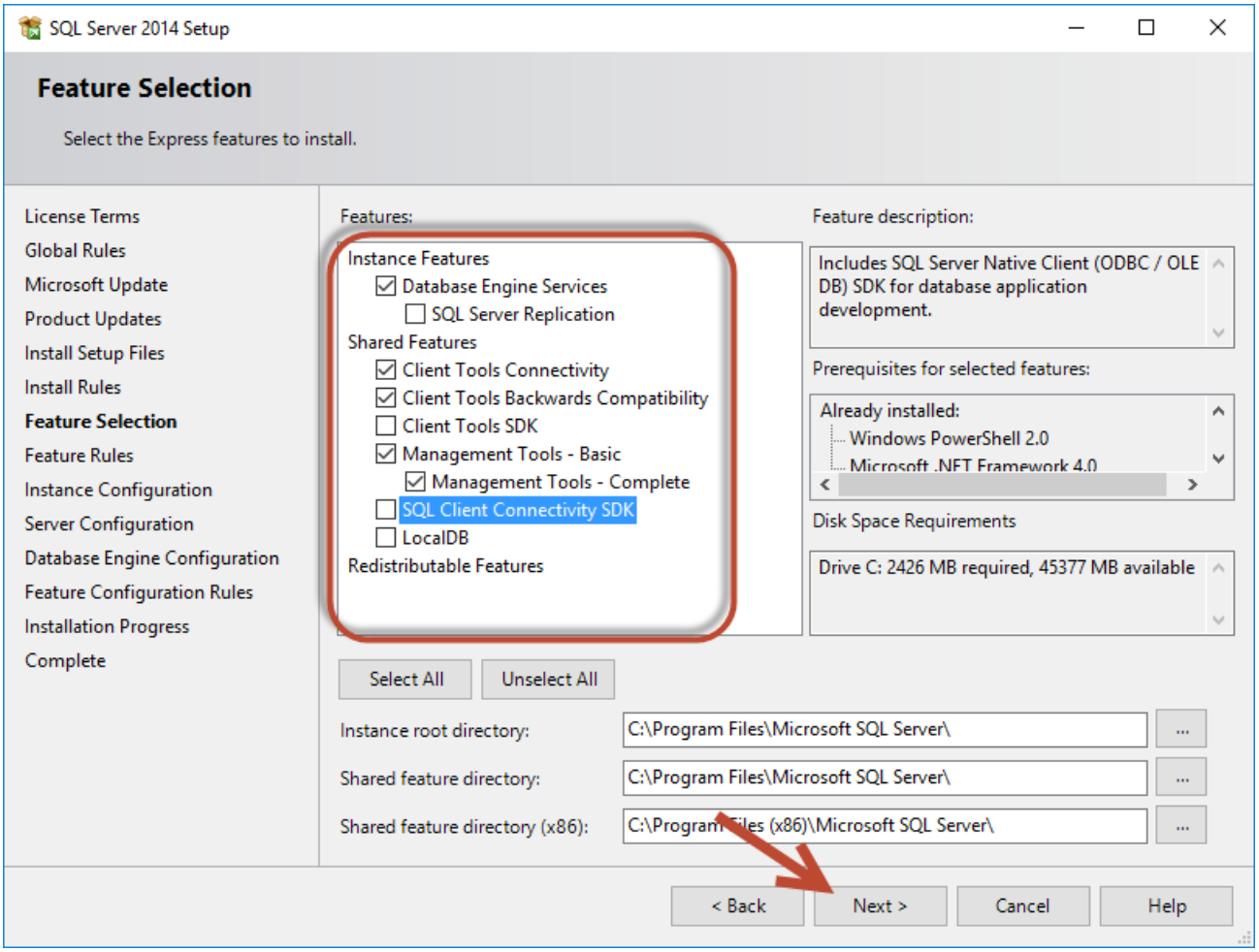


Accept license terms:

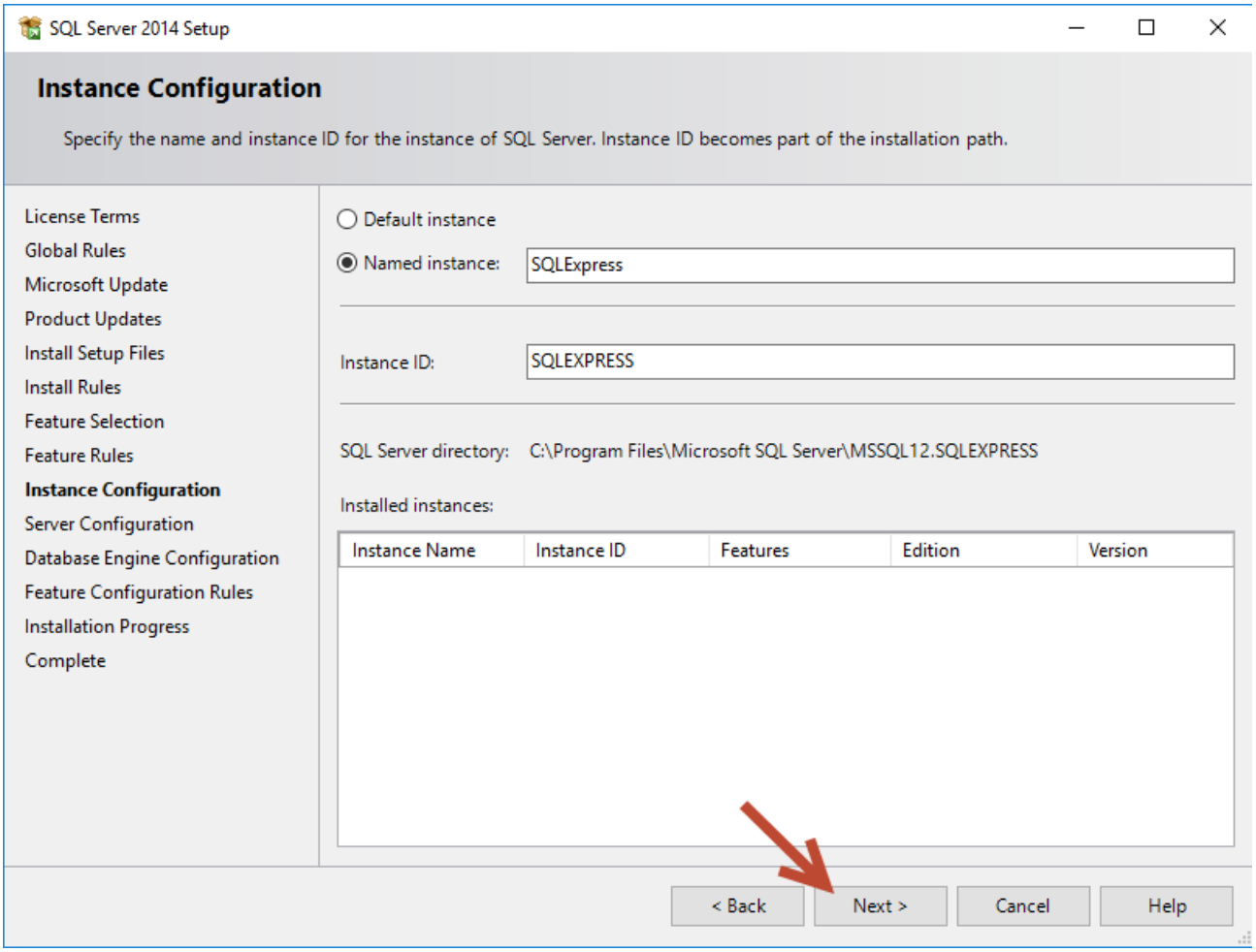



SQL Server 2014 Setup
Microsoft Update
Use Microsoft Update to check for important updates
License Terms
Global Rules
Microsoft Update
Product Updates
Install Setup Files
Install Rules
Feature Selection
Feature Rules
Feature Configuration Rules
Installation Progress
Complete
Microsoft Update offers security and other important updates for Windows and other Microsoft software, including SQL Server 2014. Updates are delivered using Automatic Updates, or you can visit the Microsoft Update website.
Use Microsoft Update to check for updates (recommended)
Microsoft Update FAQ
Microsoft Update Privacy Statement
< Back
Next >
Cancel

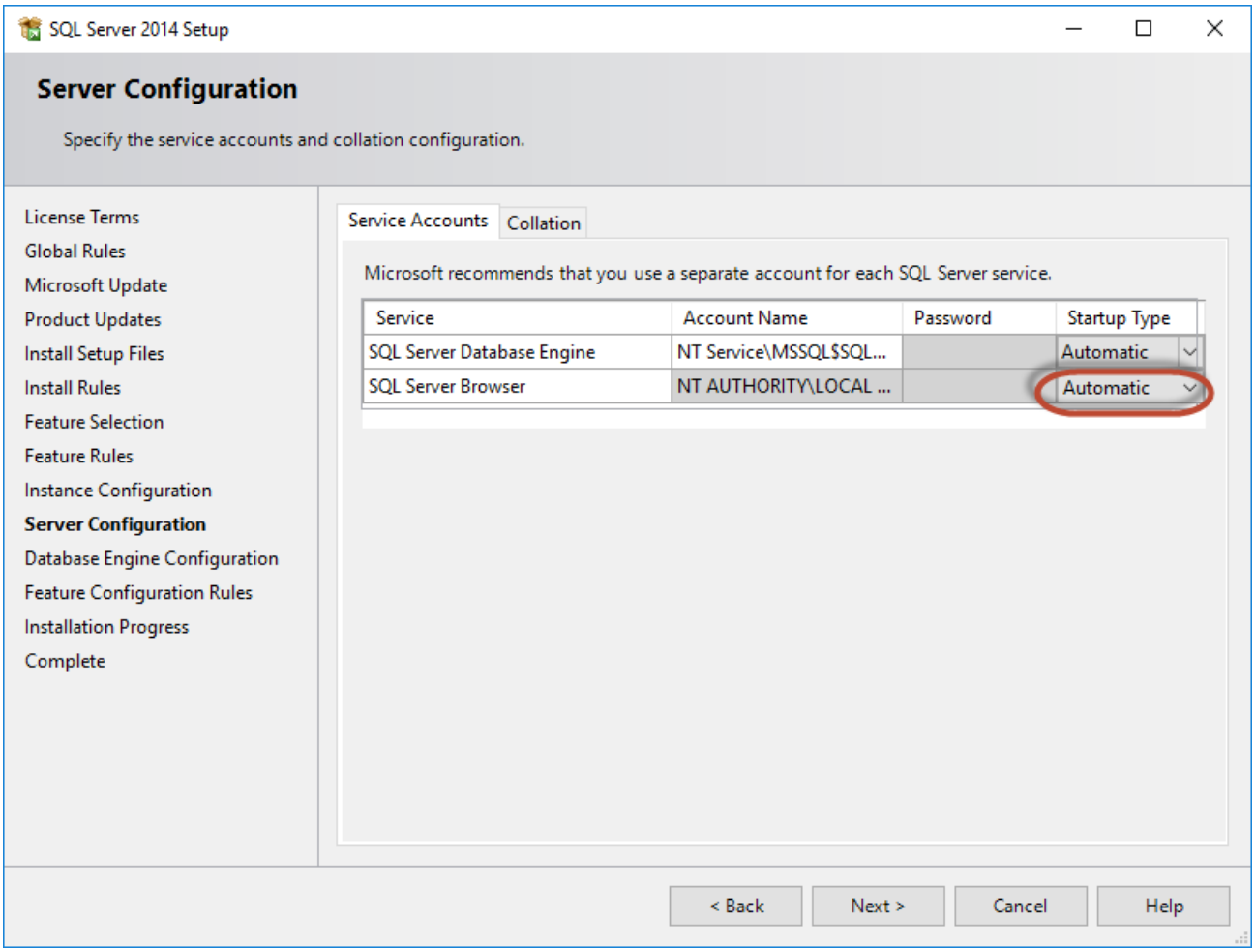
Select features and storage (directories):



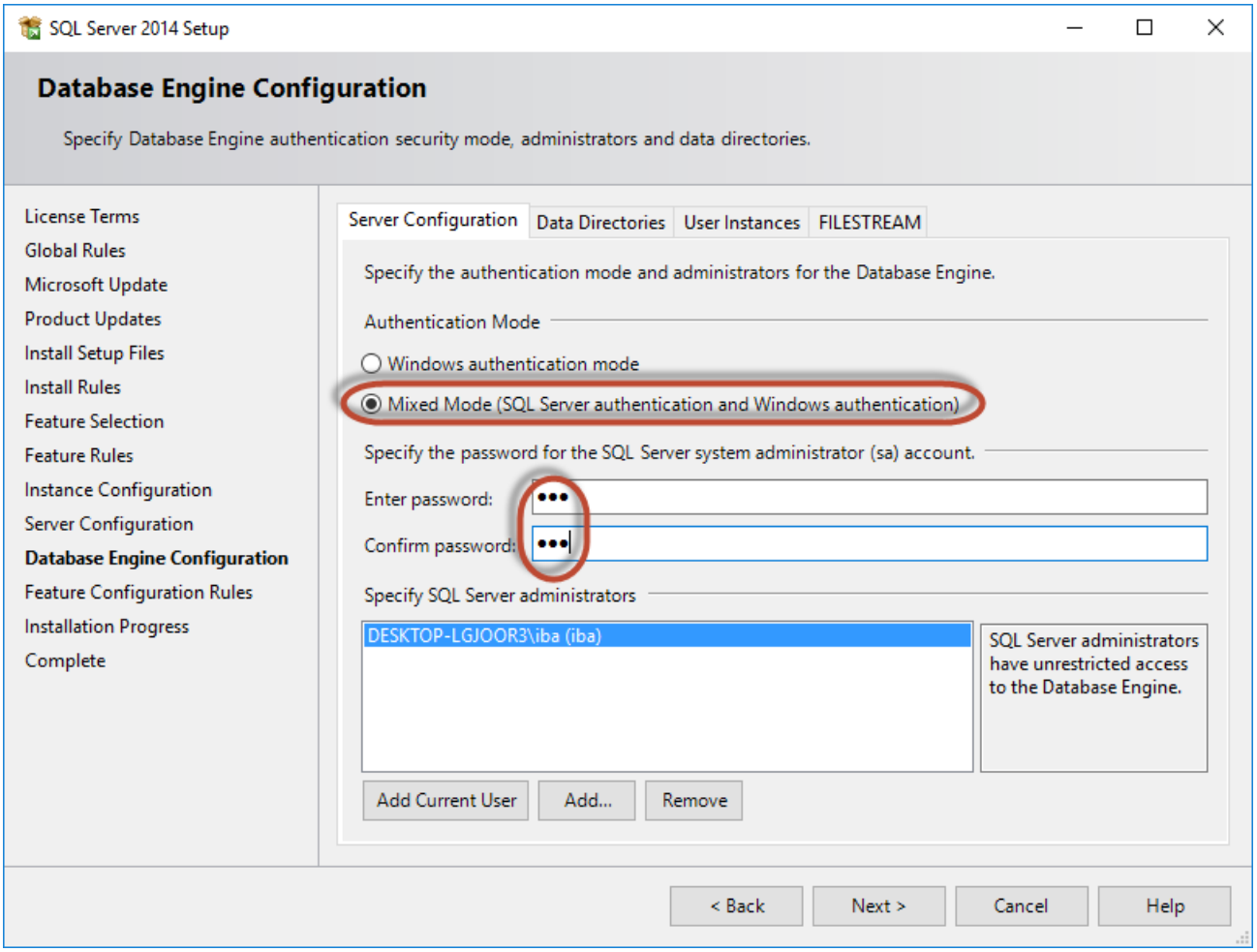
Select default instance name "SQLExpress" and ID "SQLEXPRESS":



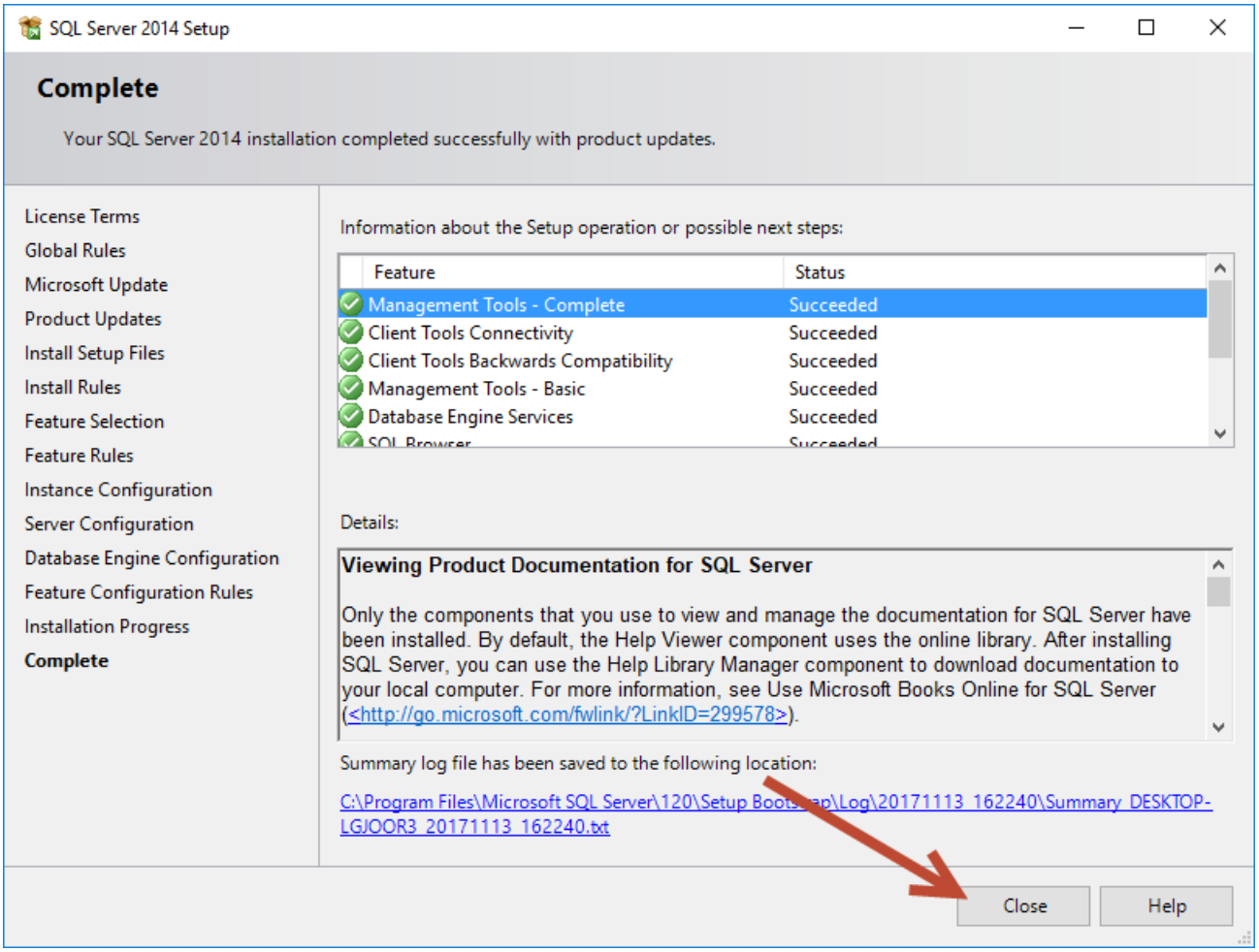
If the remote access to the database is required set the "SQL Server Browser" Startup Type to "Automatic". Further settings for remote access are described later.



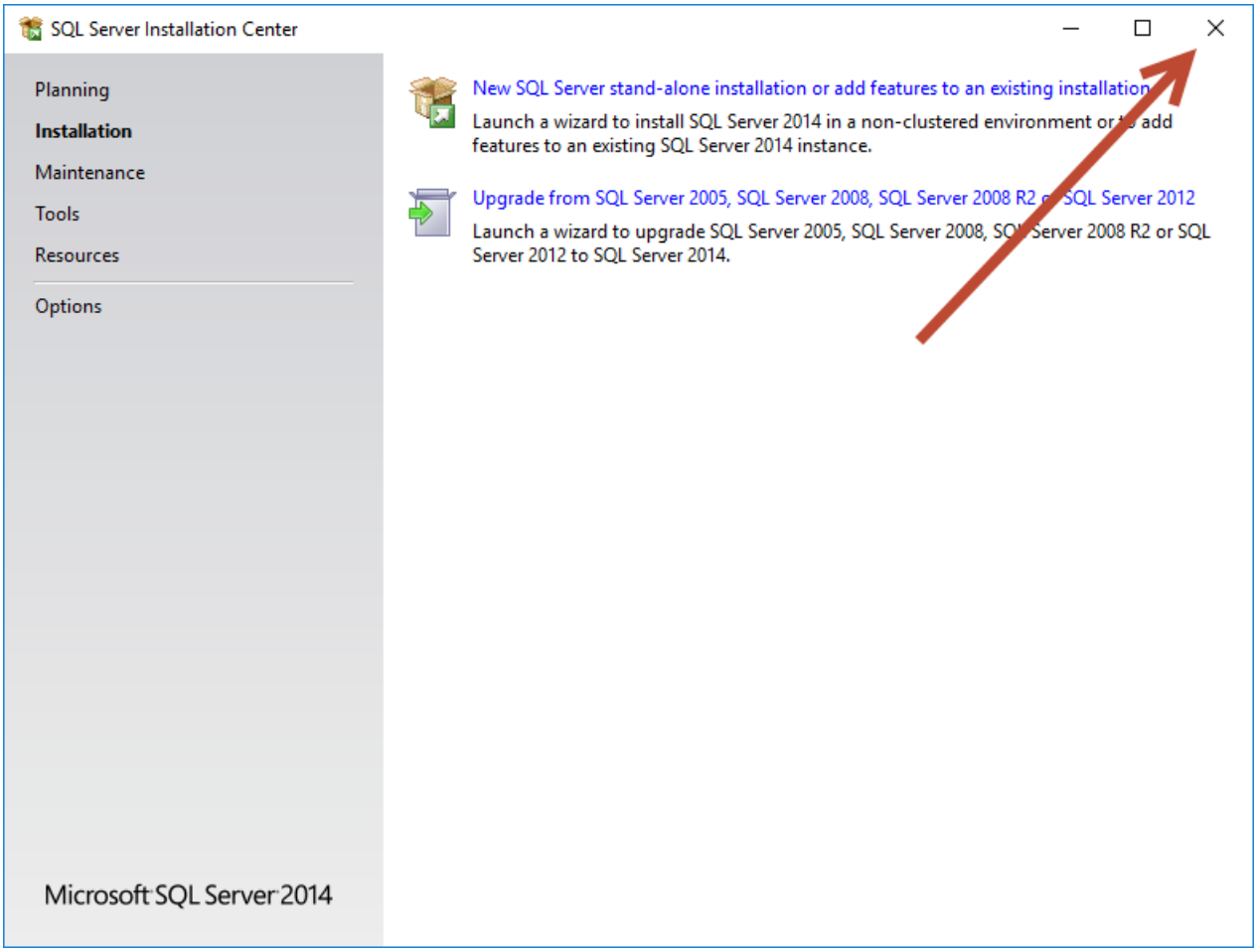
For external access choose "Mixed Mode…" and specify the password for the system administrator.



The installation now runs for a few minutes and at the end all features have "Succeeded".

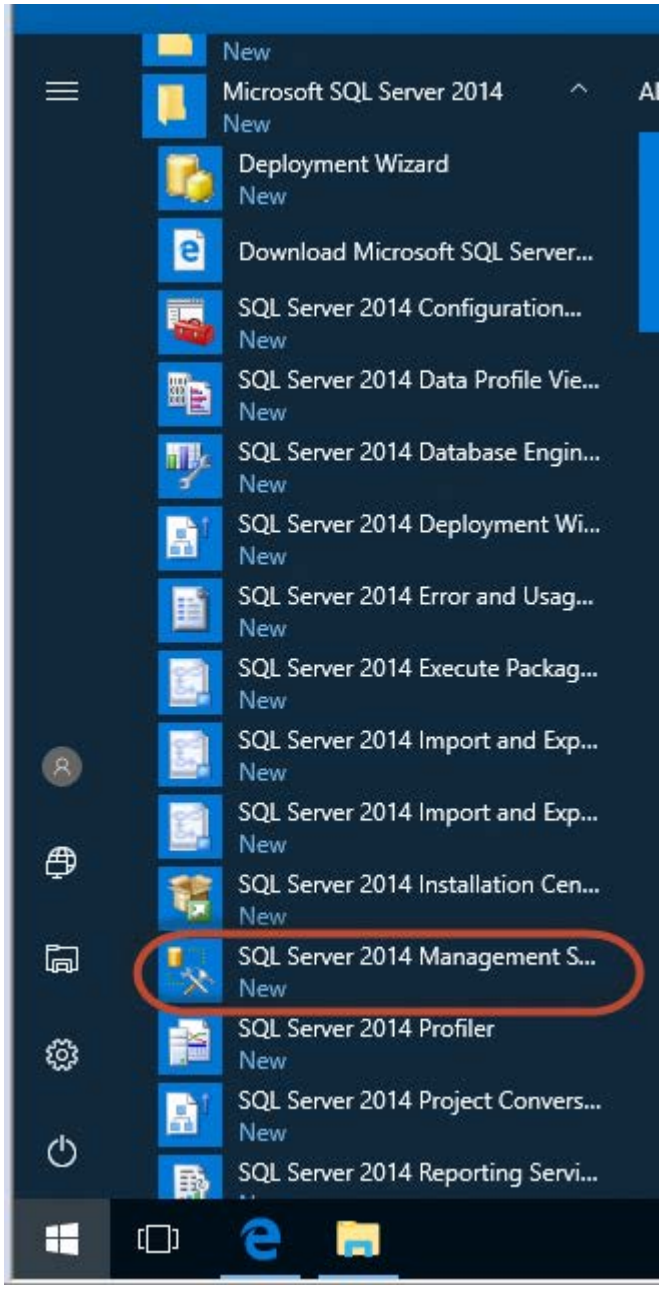


Close Install Program:

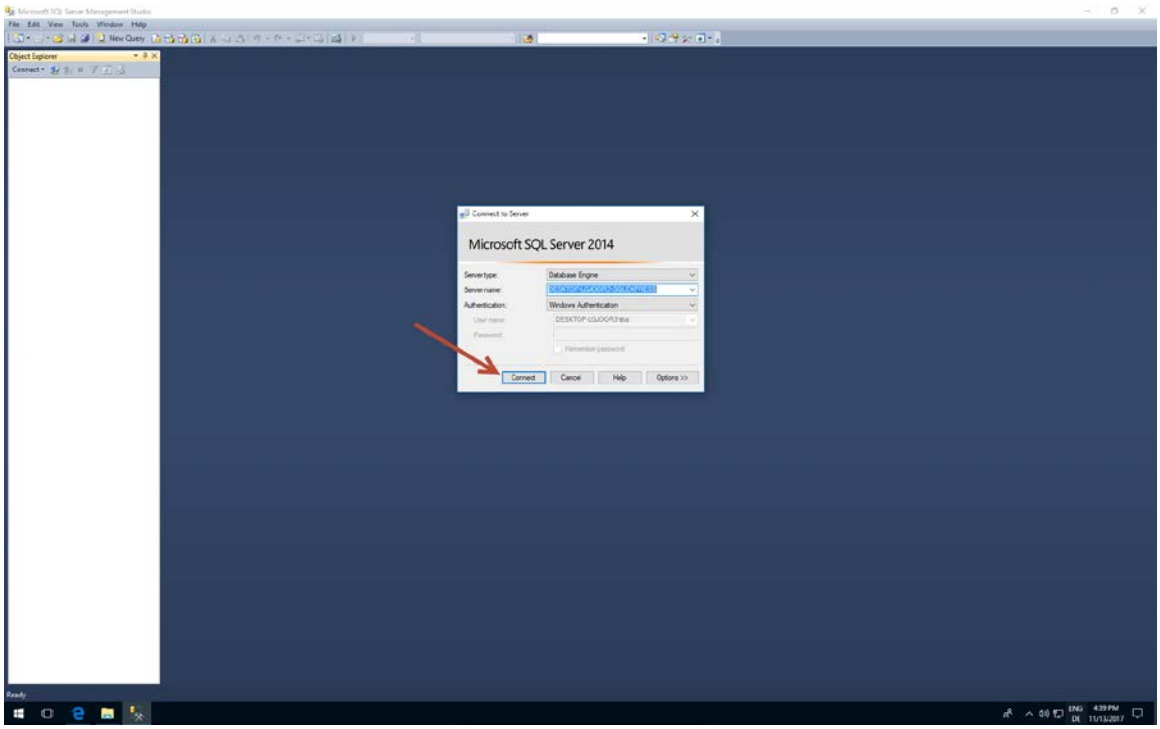


10.2 Create database and basic settings for the server:

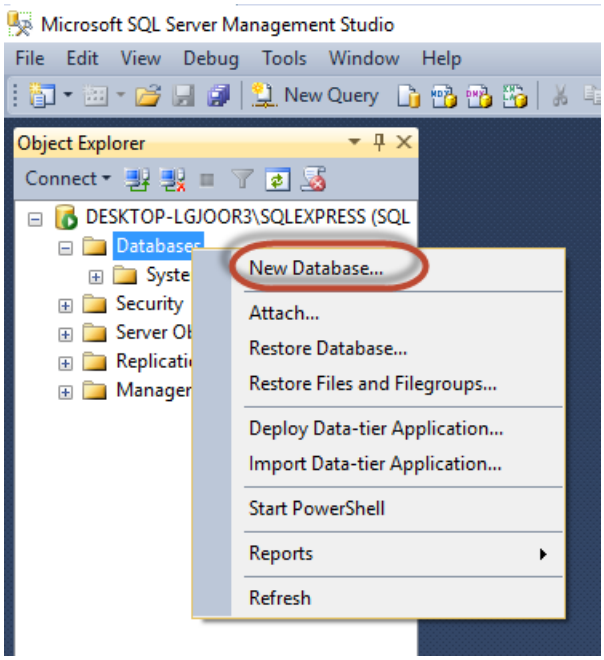
10.2.1 Start SQL Server 2014 Management Studio from start menu



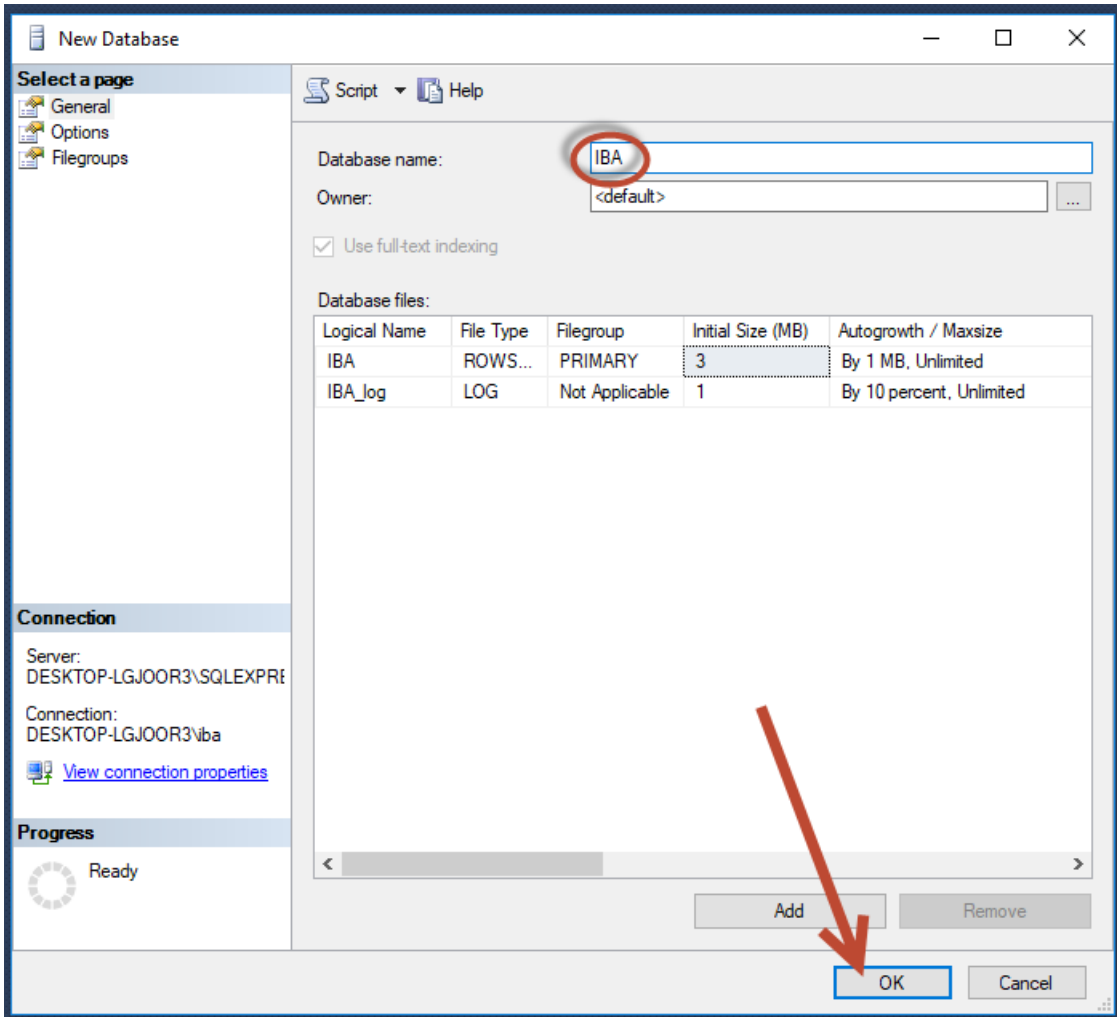
Connect to Server



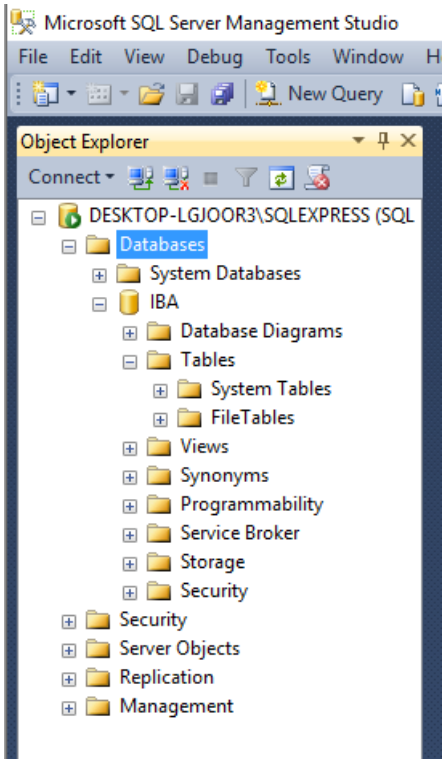
Create new database (right mouse click on "Database"):



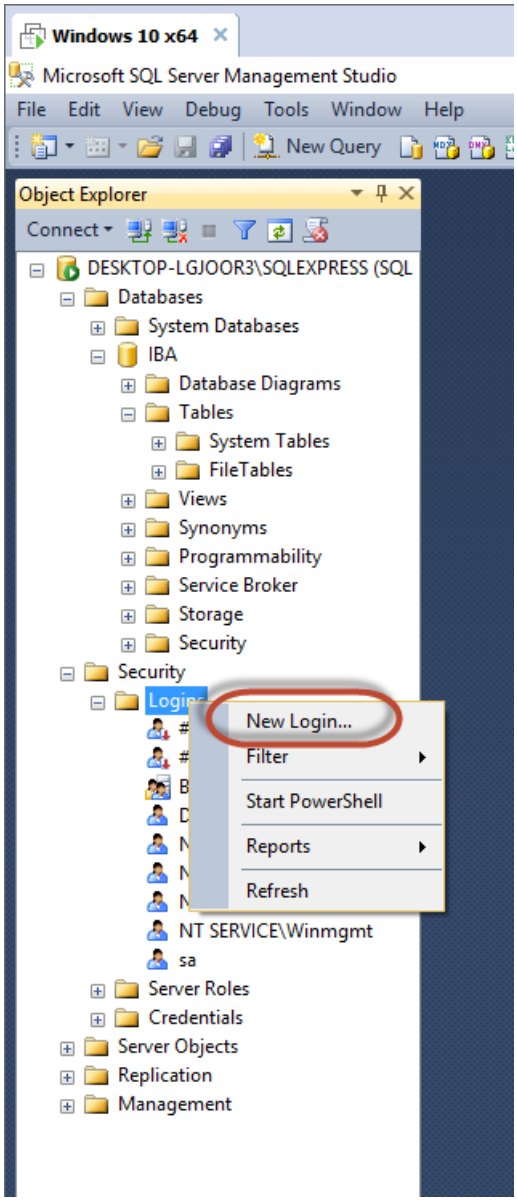
Specify the "Database name" and confirm default settings with <OK>:



Find the new database in "Object Explorer":

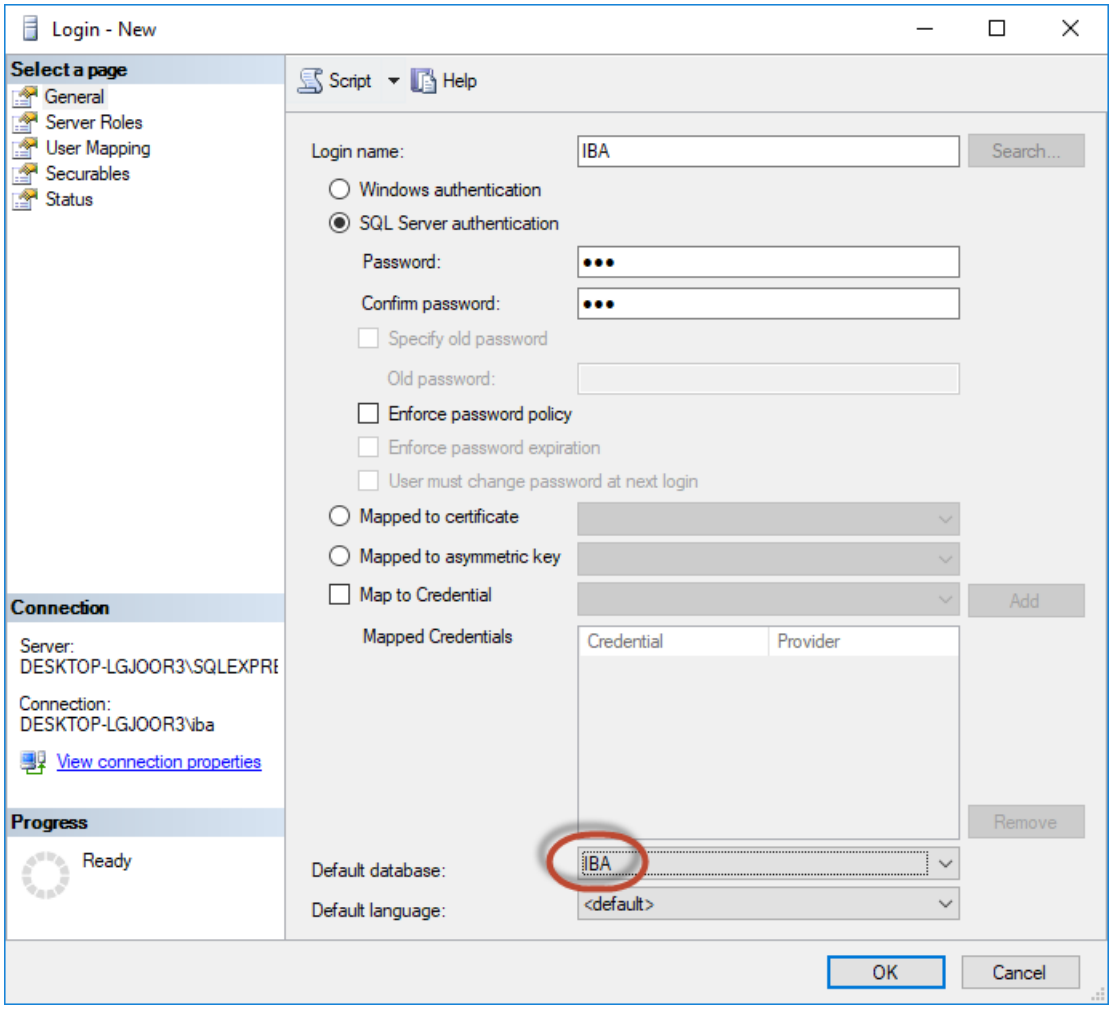


Create a new dedicated login (right mouse click on "Logins"):

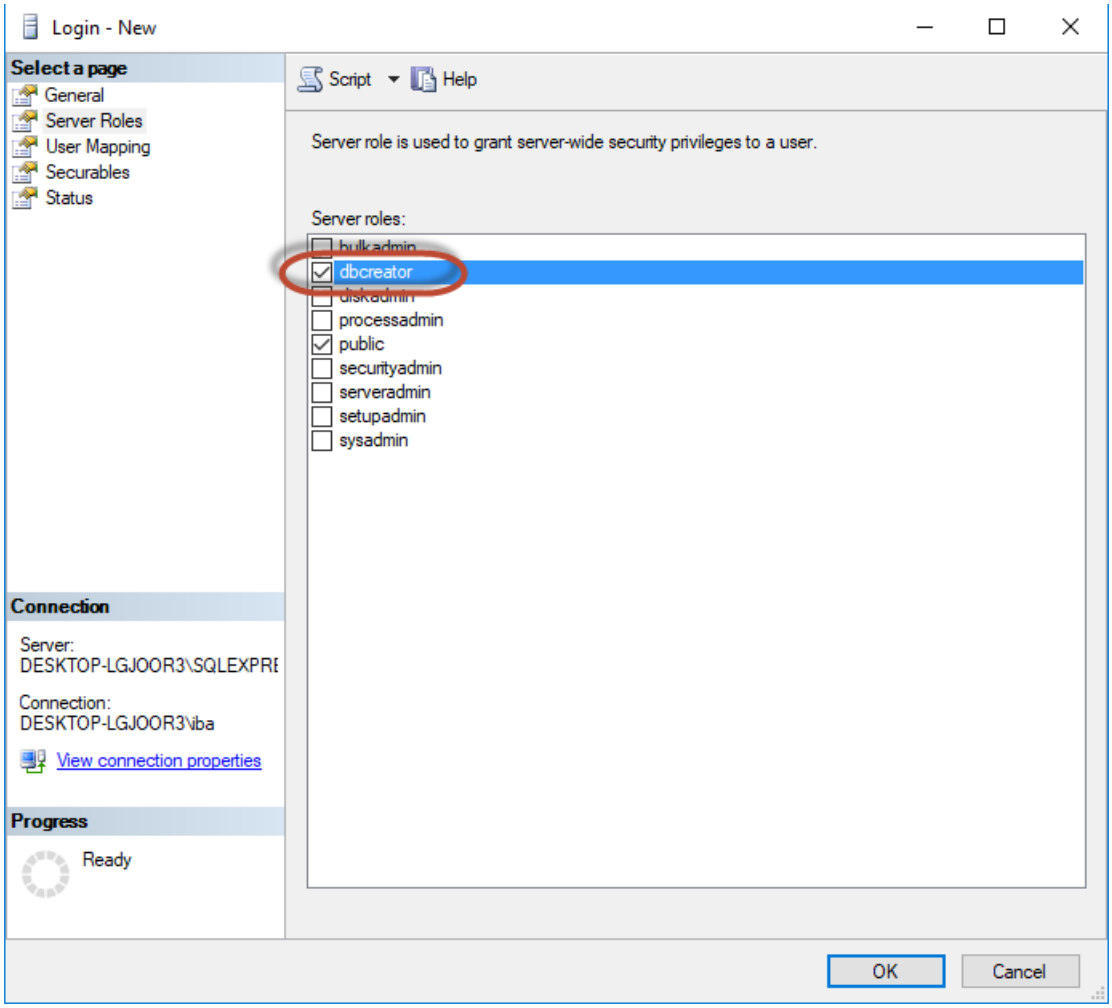


On the first page "General":

Specify the "login name", choose "SQL Server authentication" and define a password. Disable "Enforce password policy" and choose "Default database"

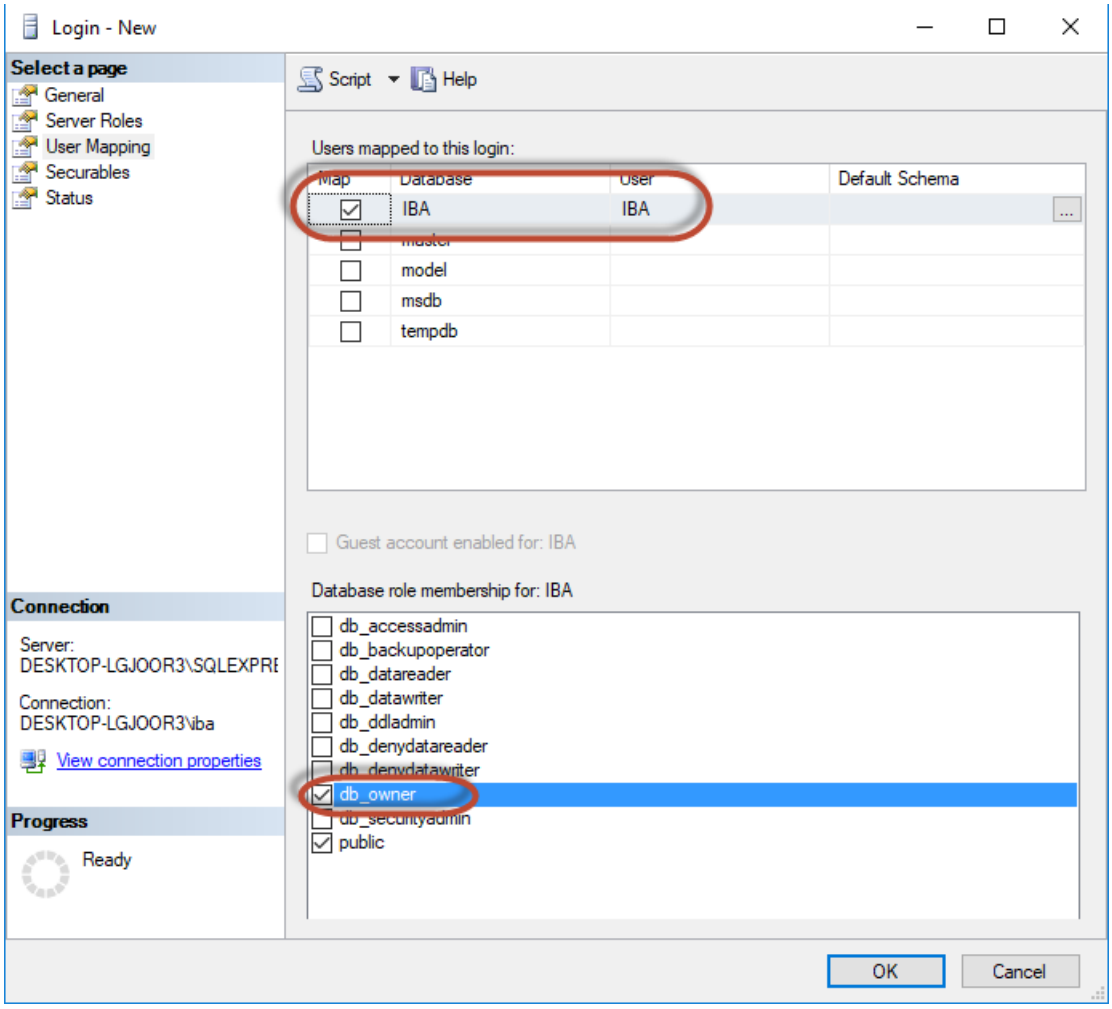


On the second page "Server roles" choose "dbcreator":



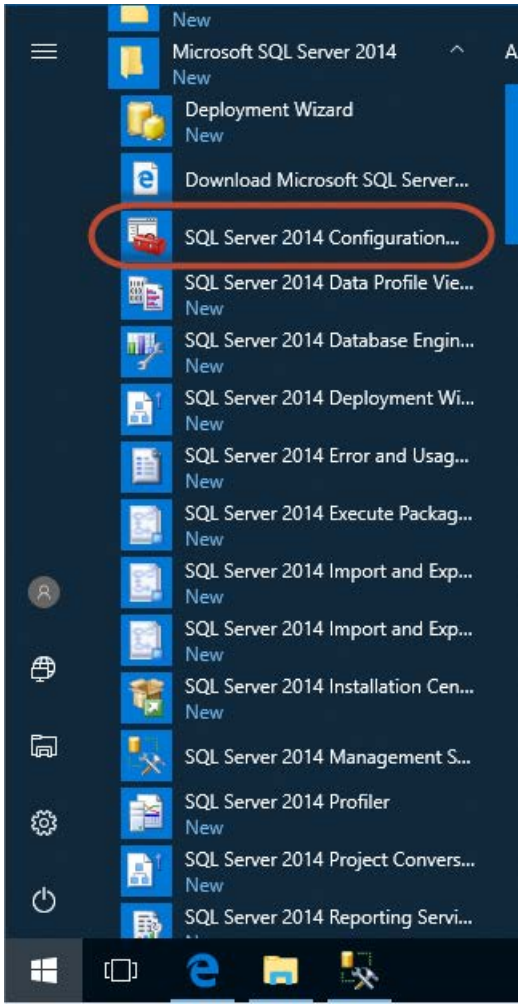
On the third page select "Database" in the window "User mapped to this login" and "db_owner" in the window "Database role membership for:".

Then confirm the Login configuration with <OK>:

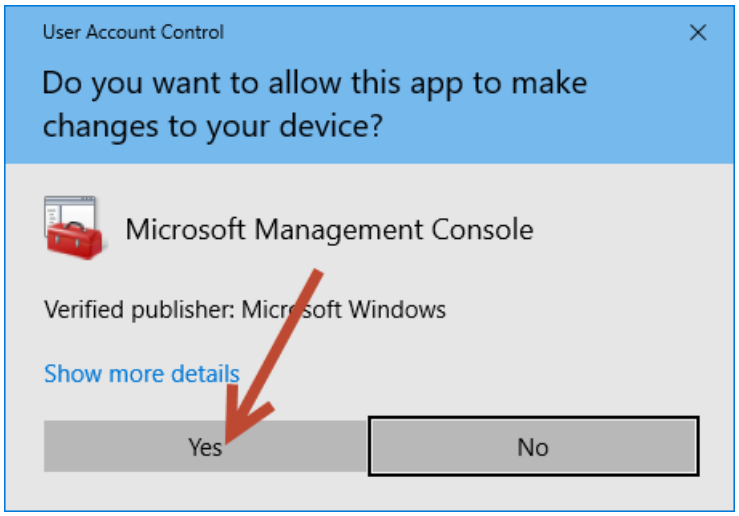


10.2.2 Customize Settings for remote access

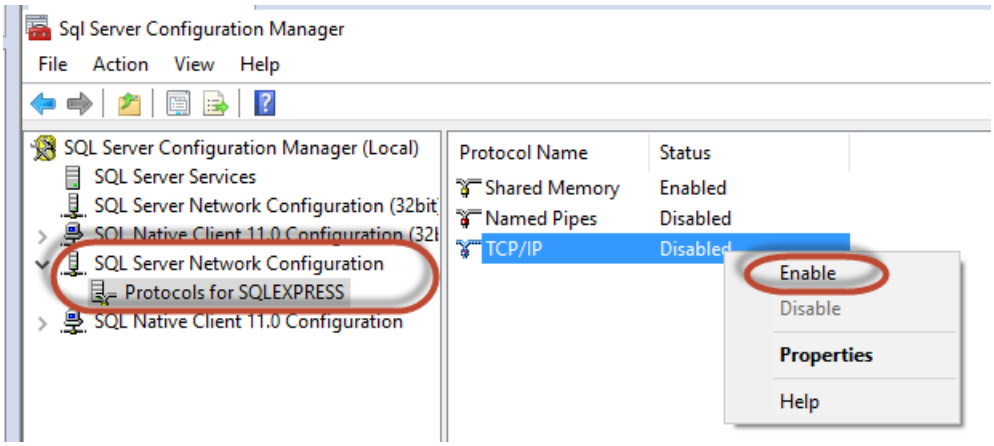
10.2.2.1 Start <SQL Server 2014 Configuration Manager>



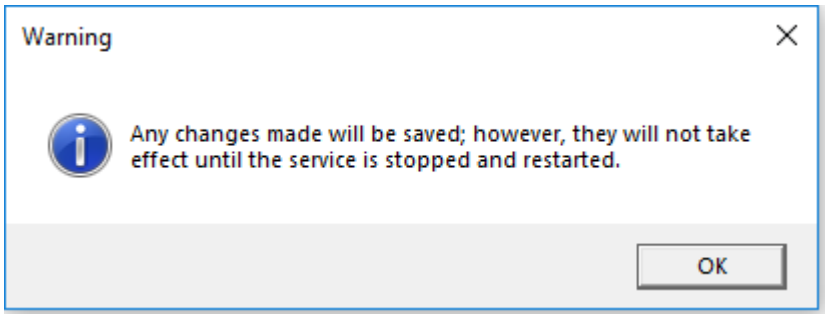
Allow system changes:



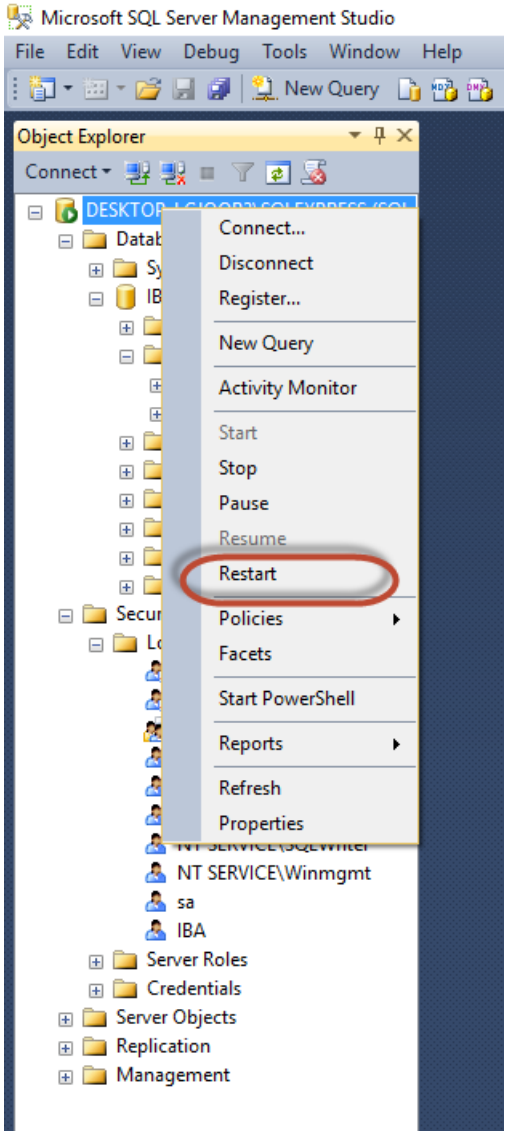
Activate TCP/IP protocol for "SQL Server Network Configuration" (right mouse click on "TCP/IP"):



Notice Warning => Restart Database Service



Restart Database Service from Management Console (right mouse click on DESKTOP-LGJOOR3\SQLE):

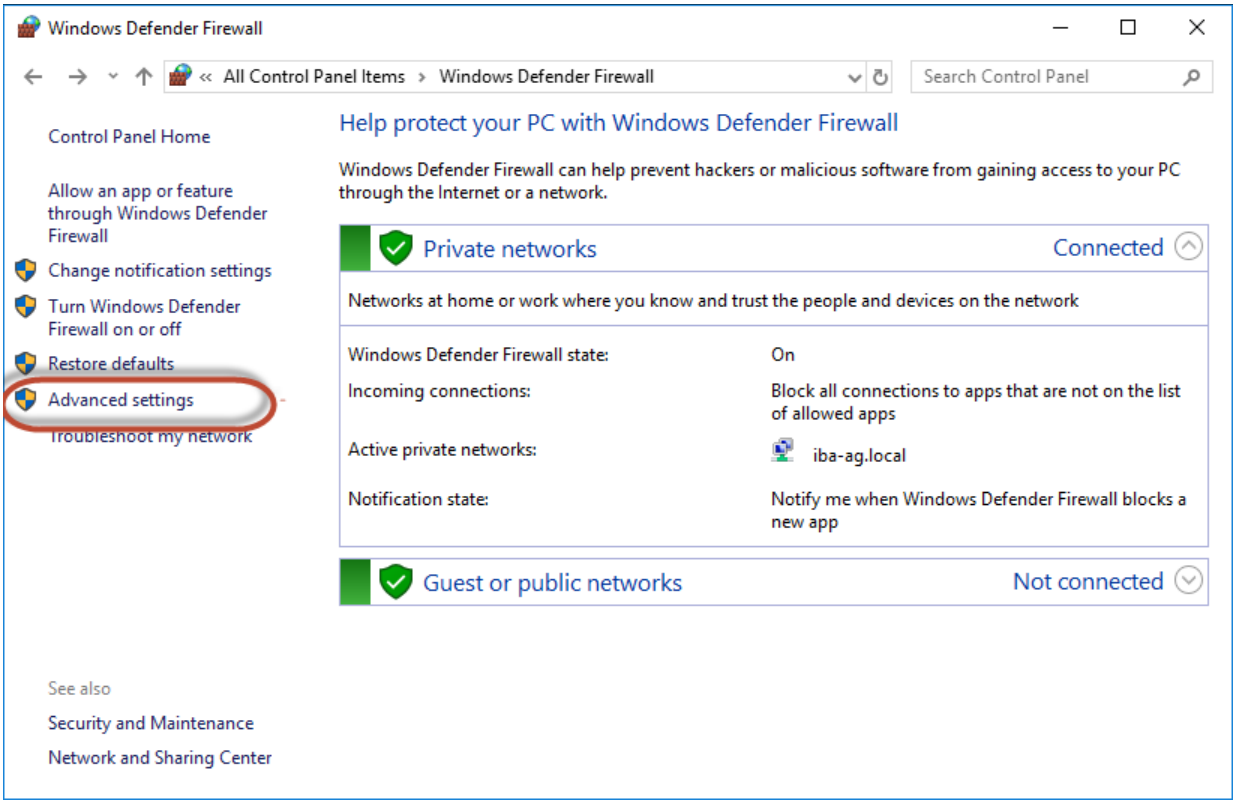


10.2.2.2 Firewall settings for remote access

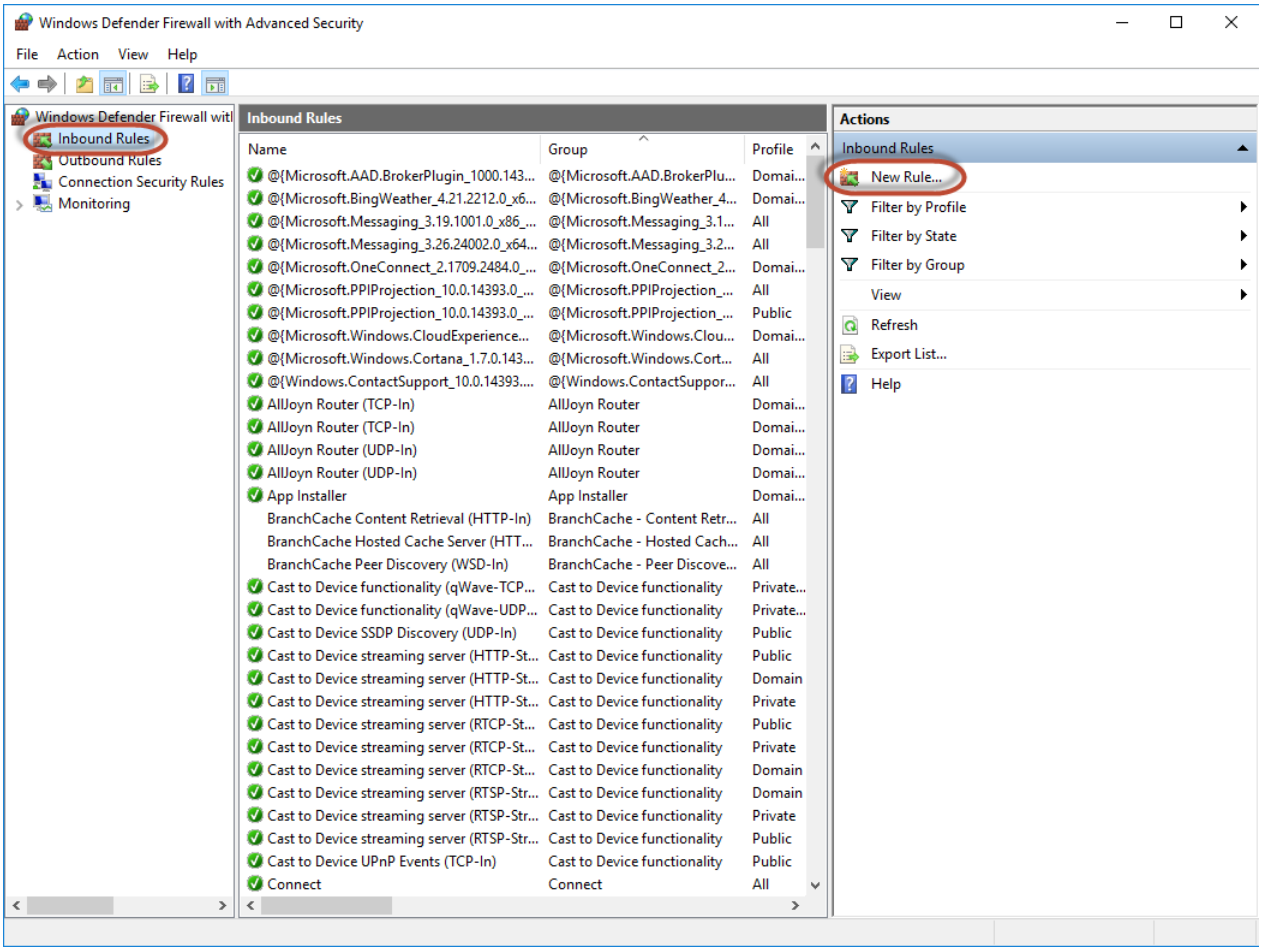
For remote access to the SQL Server three inbound rules must be configured in the windows firewall:

1. The "sqlservr.exe" executable
2. The TCP port 1433
3. The UDP port 1434

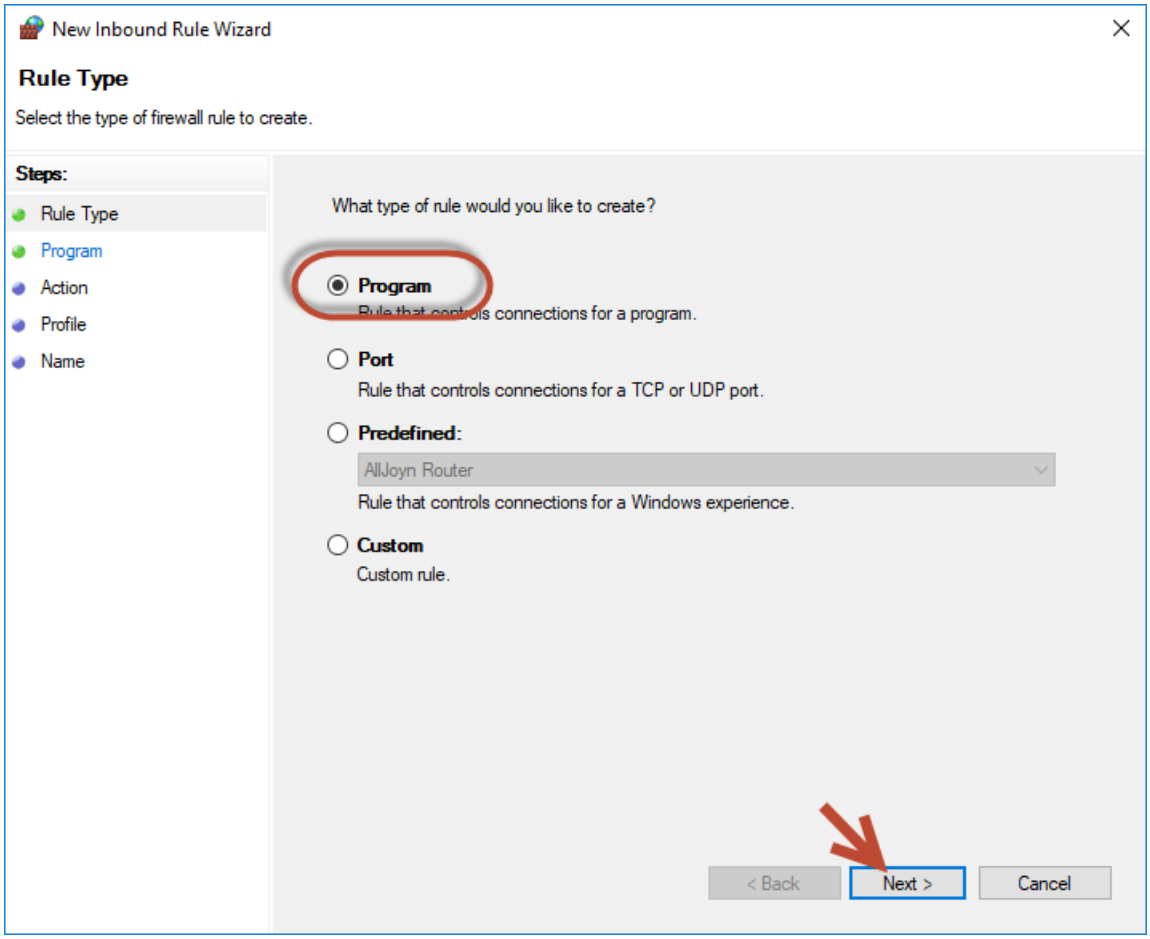
Open Firewall configuration from Control Panel and select "Advanced settings":



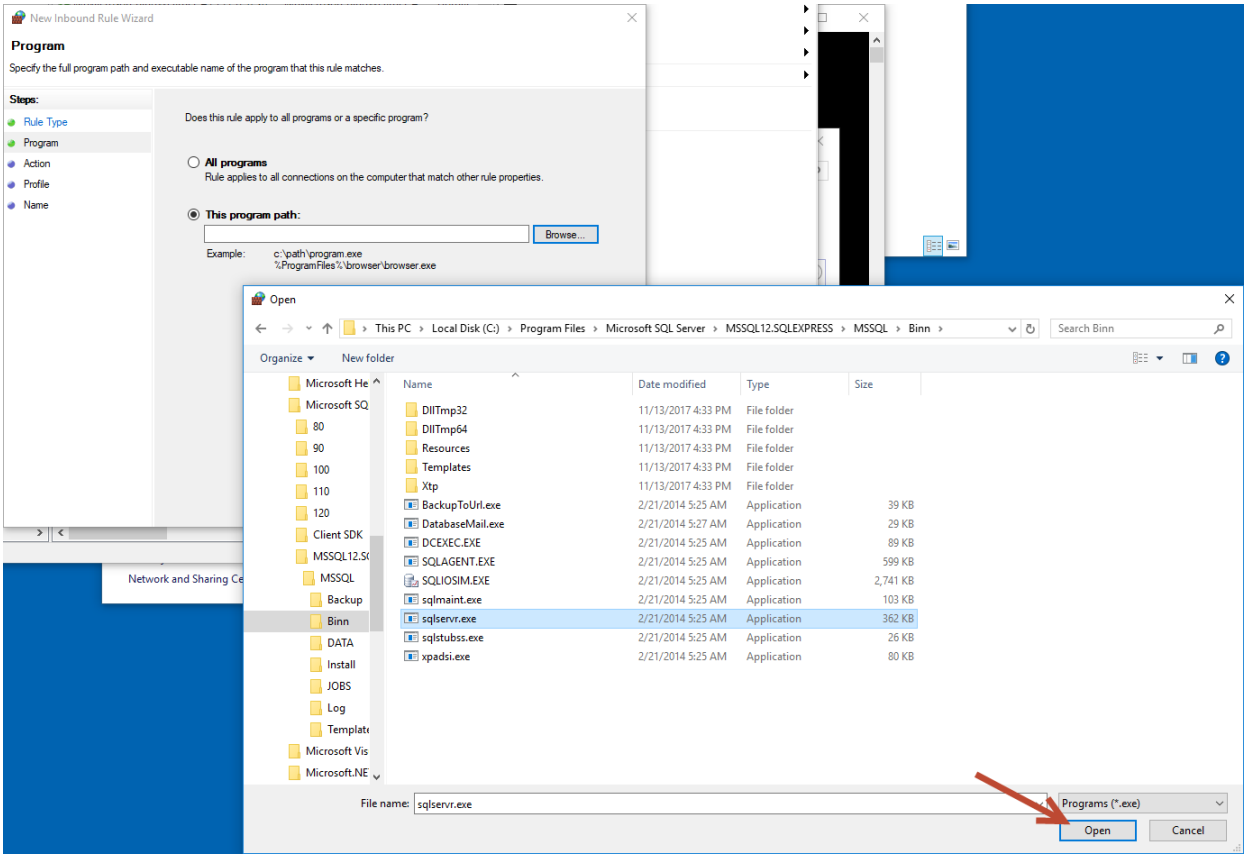
Define a new "Inbound Rule"



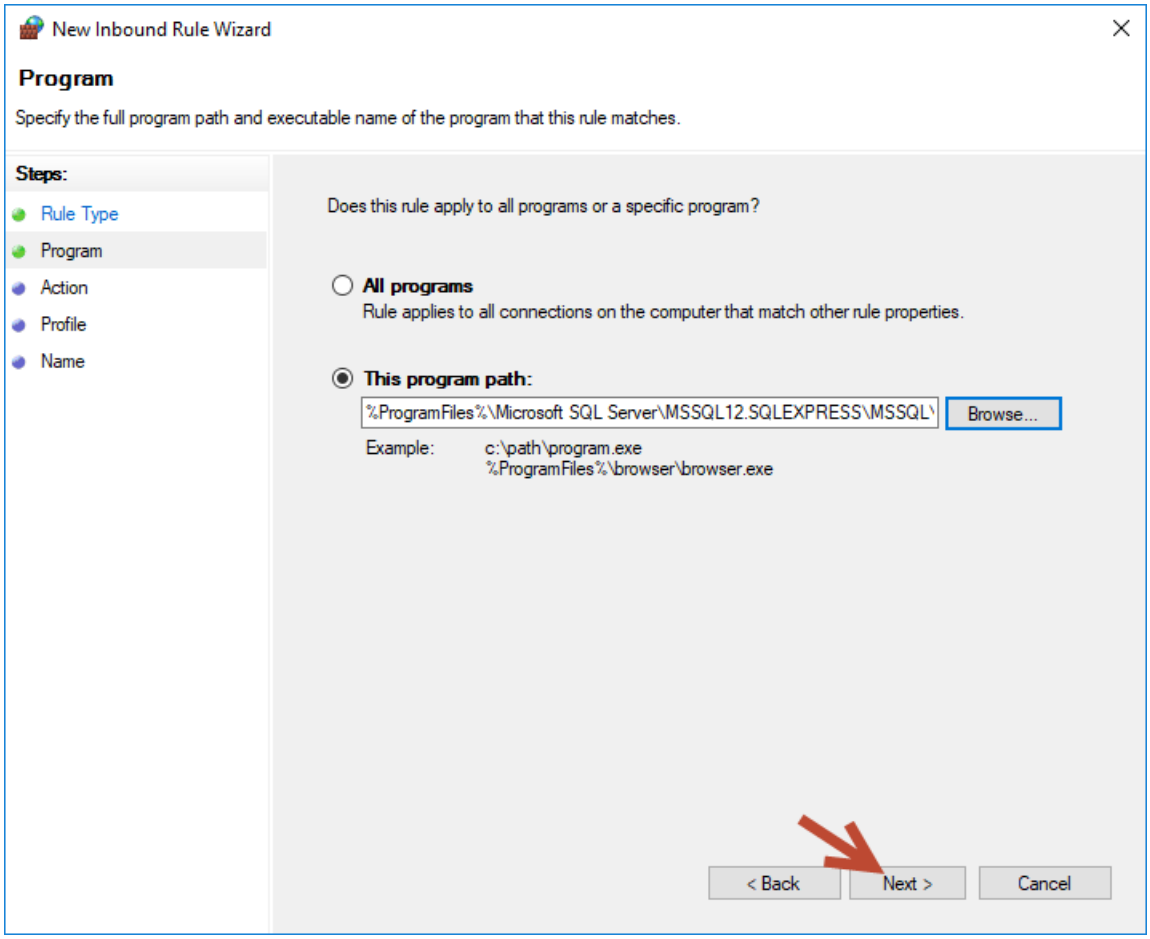
Select the *Program* rule type:



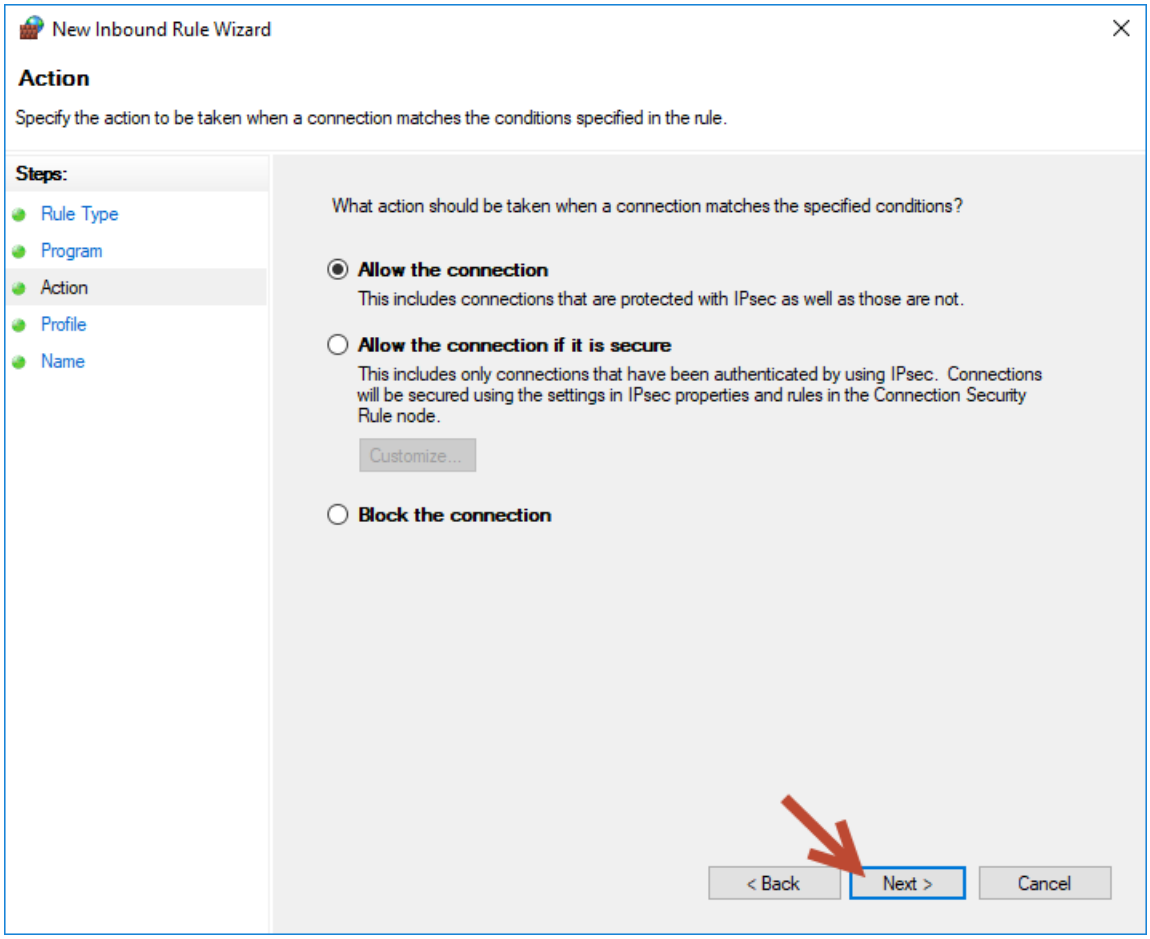
Select "The program path" and "Browse…" to the executable "sqlservr.exe"



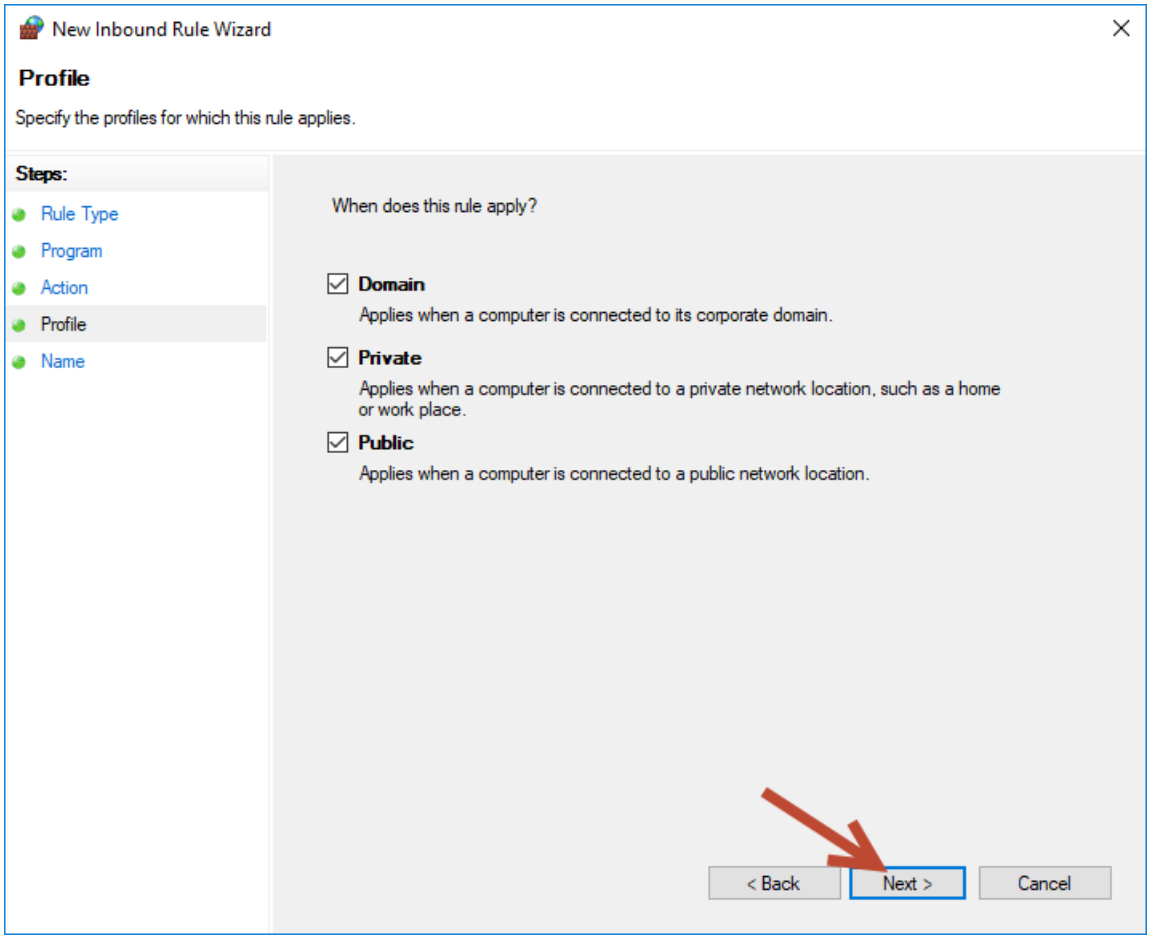
Select *This program path:* and confirm with <Next>.



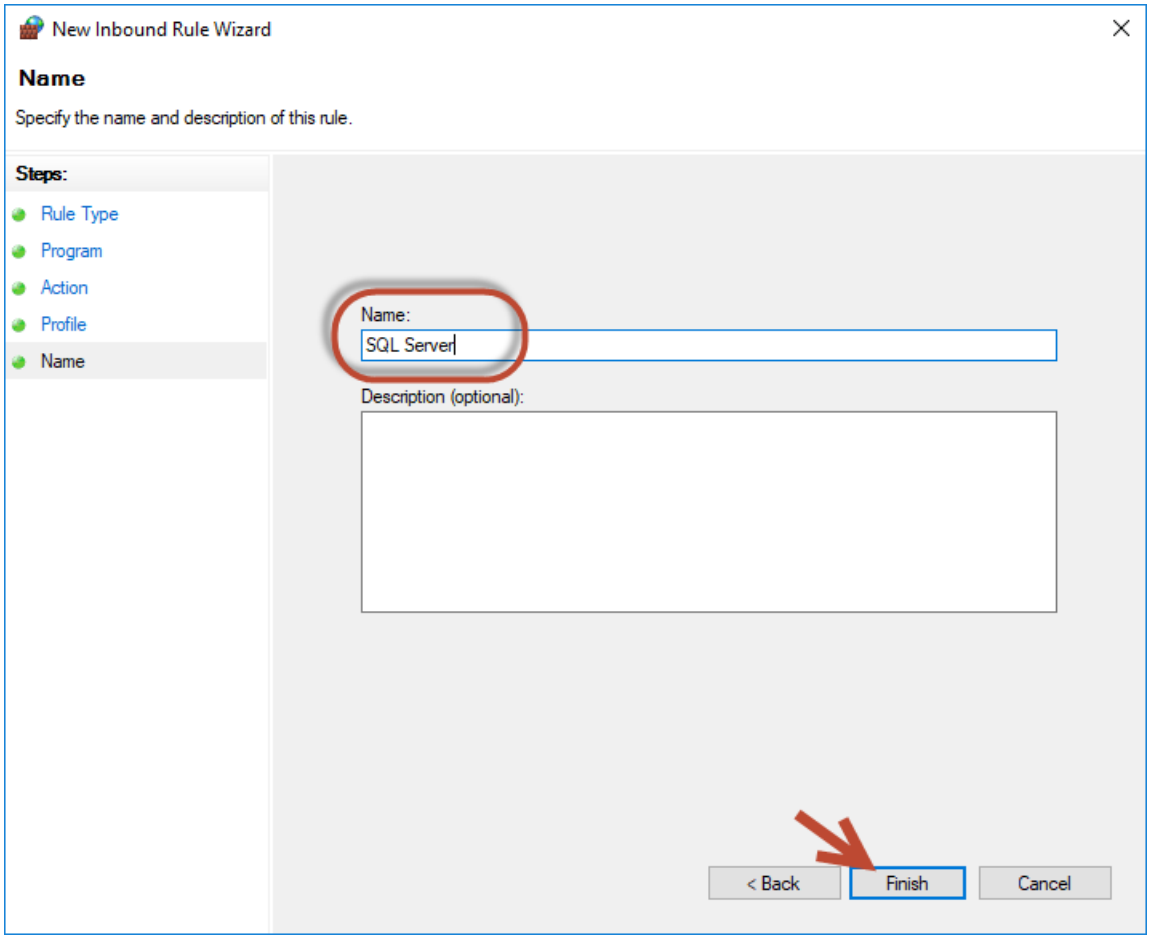
Allow the Connection – <Next>



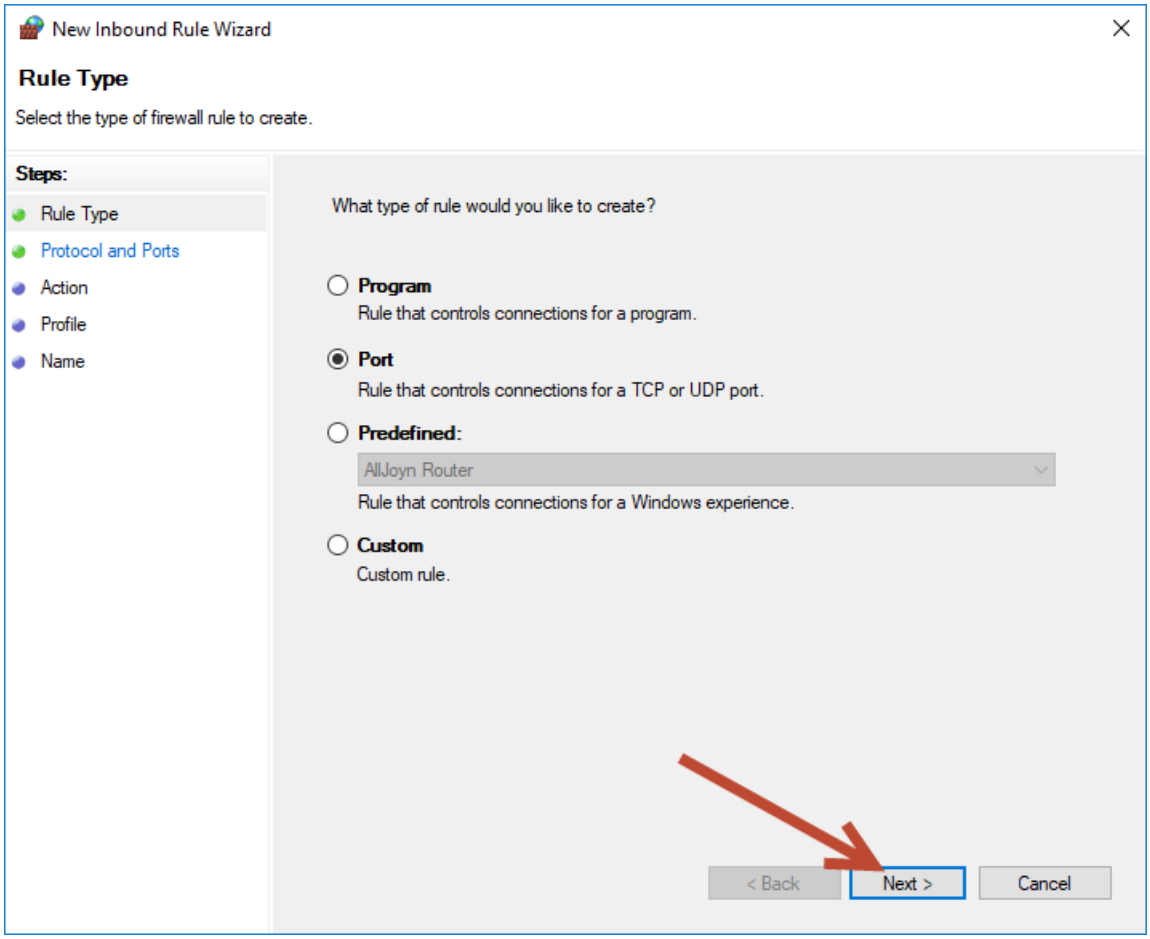
Apply "Profile" then <Next>



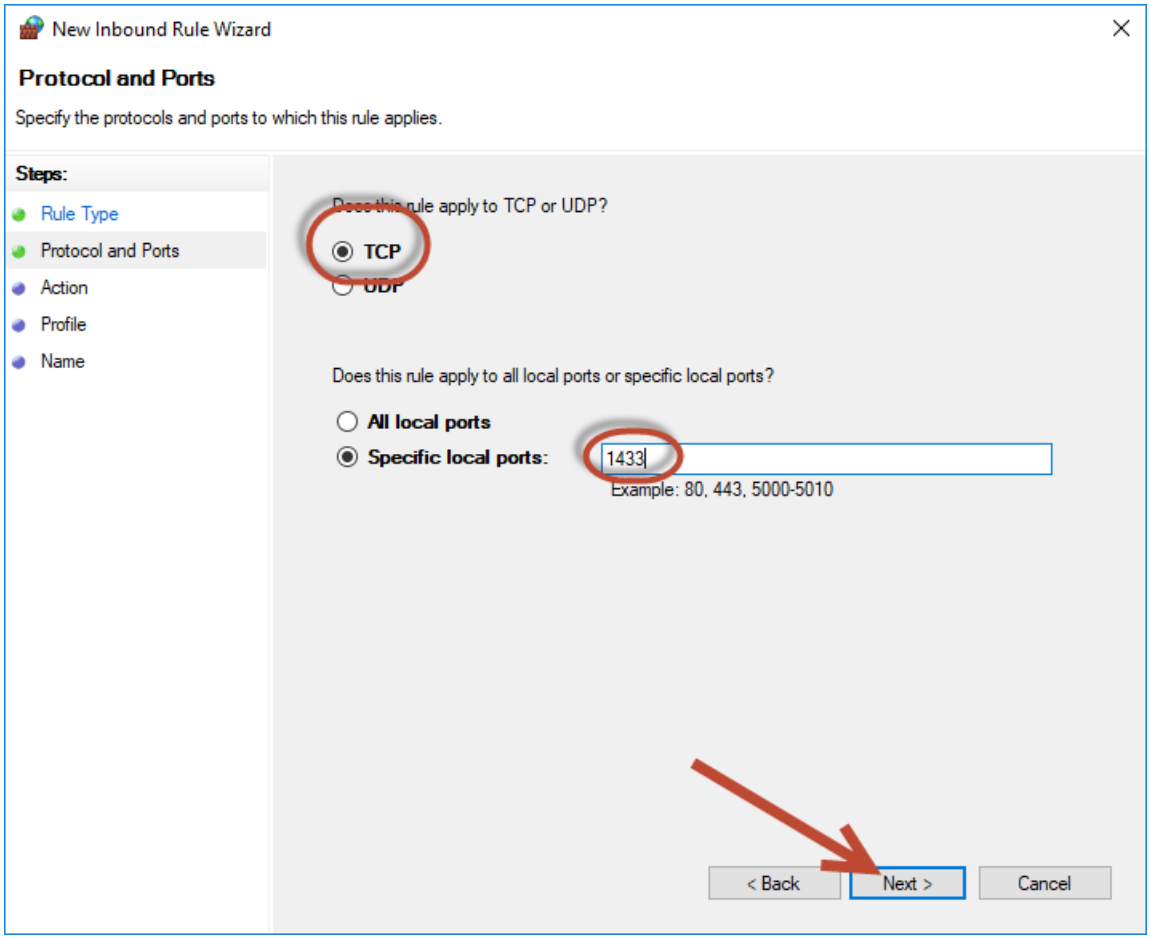
Specify *Name* and <Finish>



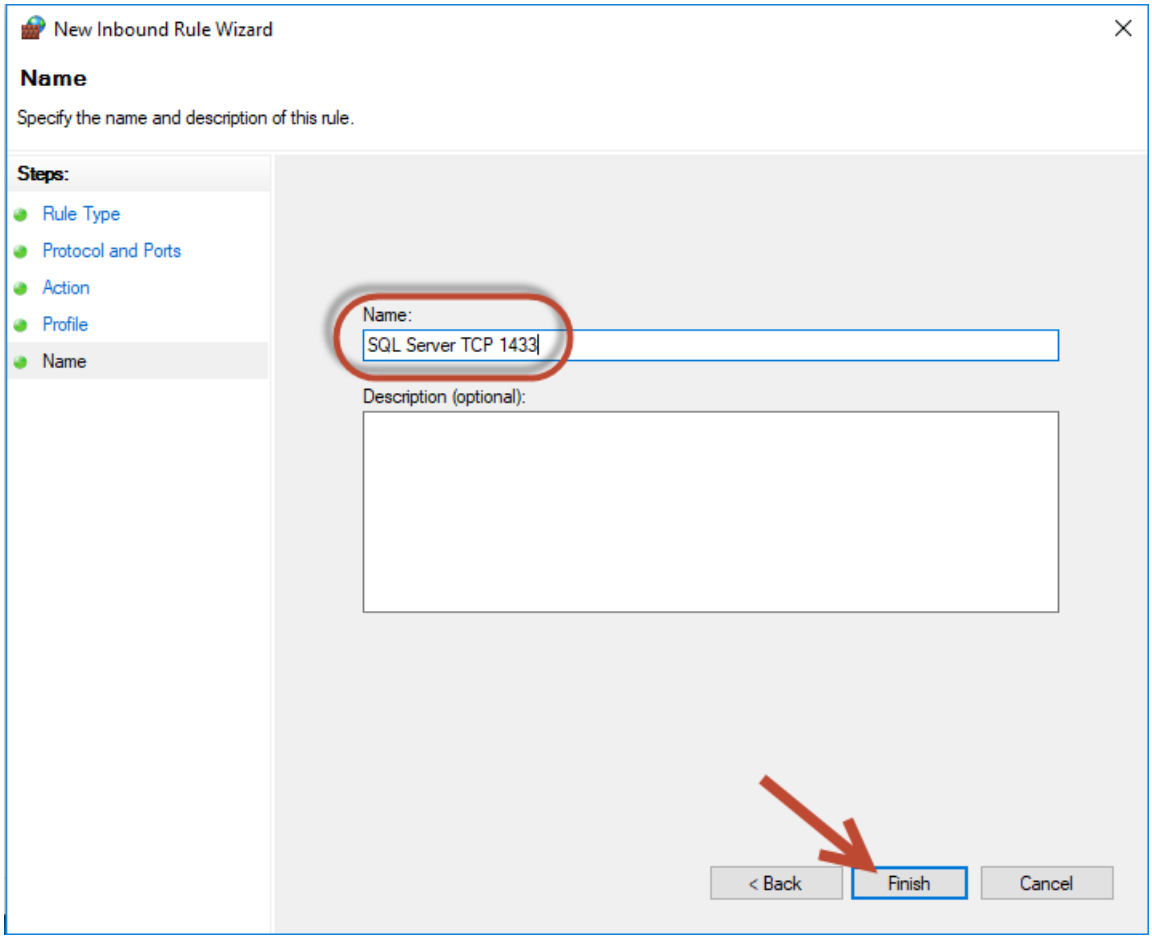
Select the *Port* Inbound rule type.



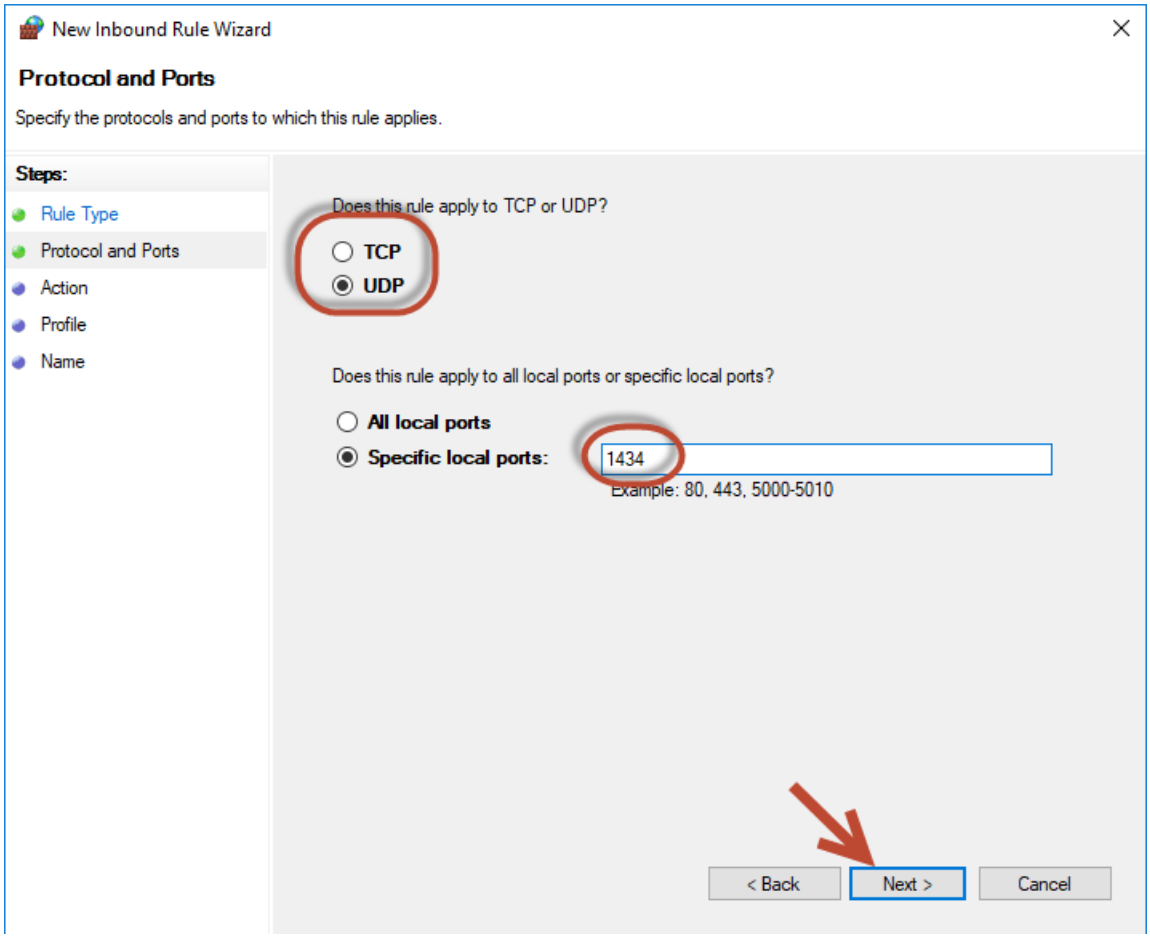
In window "Protocol and Ports" select *TCP*, *Specify local ports* and enter "1433":



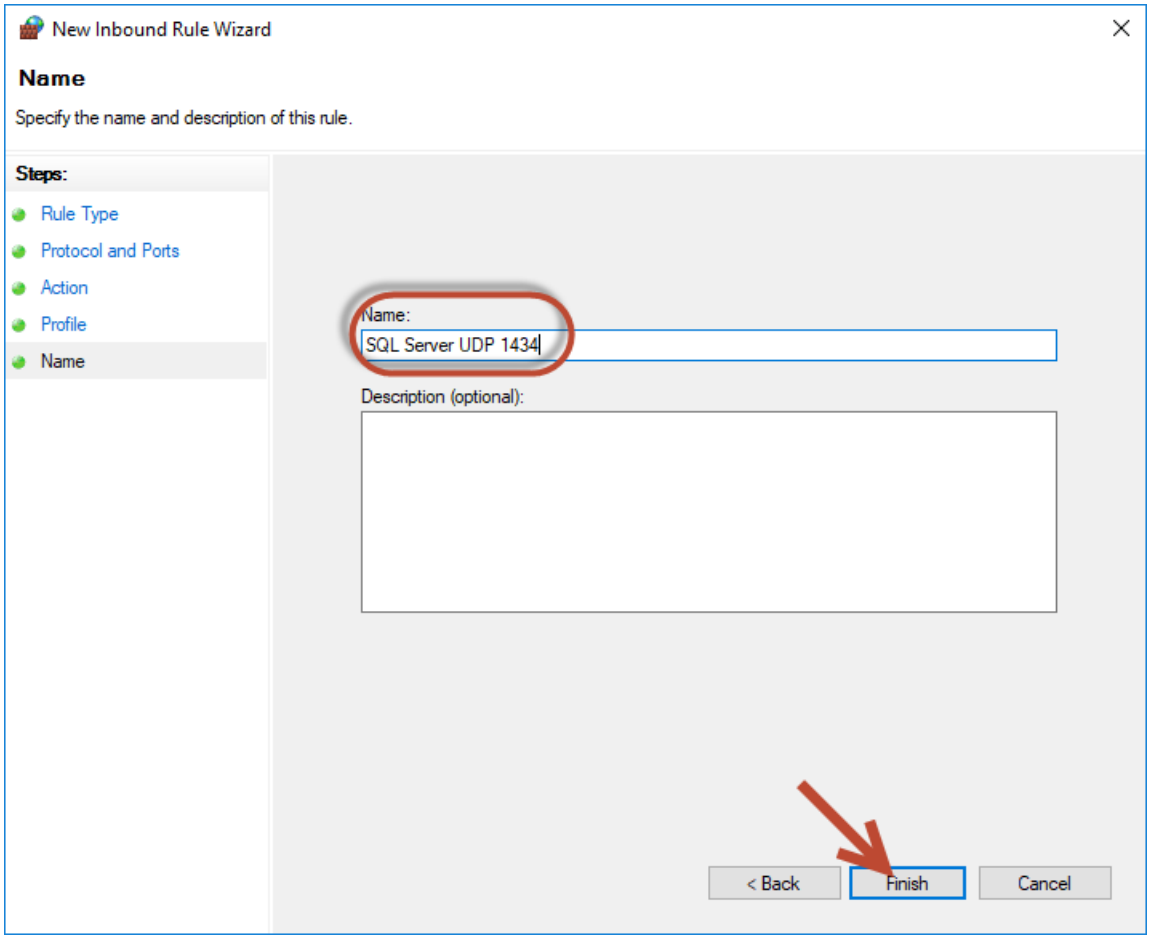
Specify *Name* and <Finish>



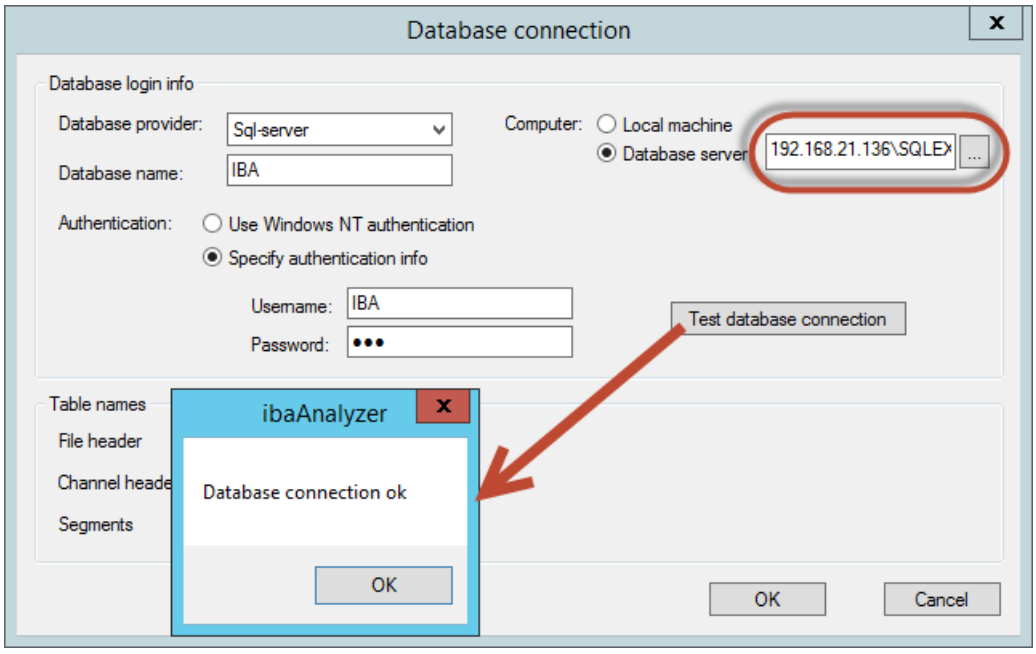
In window "Protocol and Ports" select *UDP*, *Specify local ports* and enter "1434":



Specify *Name* and <Finish>



Test connection from remote PC See chapter "Testing the Database Connection in ibaAnalyzer"



11 Support and contact

Support

Phone: +49 911 97282-14

Fax: +49 911 97282-33

Email: support@iba-ag.com

Note

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

Contact

Headquarters

iba AG
Koenigswarterstr. 44
90762 Fuerth
Germany

Phone: +49 911 97282-0

Fax: +49 911 97282-33

Email: iba@iba-ag.com

Contact: Mr Harald Opel

Regional and worldwide

For contact data of your regional iba office or representative please refer to our web site **www.iba-ag.com.**