SAP® MaxDB™ Expert Sessions

SAP MaxDB Tracing

Martin Koerner, SAP AG
Bettina Laidler, SAP AG

June 19, 2012

THE BEST-RUN BUSINESSES RUN SAP™

Objective of this session

Getting the answer to the following questions:

- What means tracing ?
- What is the purpose of tracing ?
- What kinds of traces are available ?
- When should I use which trace ?
- How can I create these traces ?

Agenda

1. Introduction
   - 1.1. What is tracing all about ?
   - 1.2. Kinds of traces used in MaxDB context

2. SAP MaxDB Database Trace
   - 2.1. At a glance
   - 2.2. Functional chain
   - 2.3. Additional functionality
   - 2.4. Using SAP MaxDB Tools for database tracing
     - 2.4.1. Database Manager CLI (DBMCLI)
     - 2.4.2. Database Manager GUI (DBMGUI)
     - 2.4.3. SQL Studio / SQLCLI
     - 2.4.4. Database Studio
     - 2.4.5. xkernprot
   - 2.5. Using SAP Transactions for database tracing
     - 2.5.1. DB50 / LC10
     - 2.5.2. DBACOCKPIT

Agenda

3. SAP MaxDB Interface Traces
   - 3.1. SQLDBC Trace
     - 3.1.1. Using SAP MaxDB Tools for SQLDBC tracing
     - 3.1.2. Using SAP Transactions for SQLDBC tracing
   - 3.2. JDBC Trace
   - 3.3. ODBC Trace

4. Additional Traces
   - 4.1. SQL Trace
   - 4.2. Developer Trace
   - 4.3. DBMRFC Trace

5. Useful Information Resources

Agenda

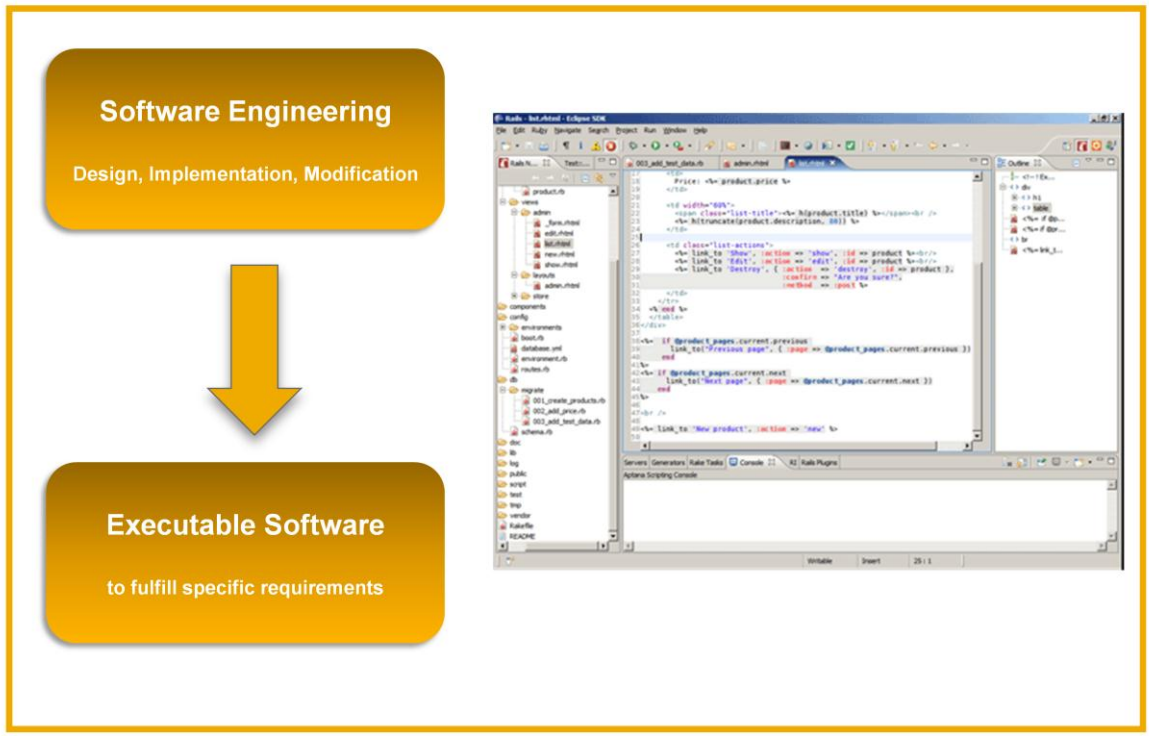
Please read the notes of the slides !

Agenda

1. Introduction
2. SAP MaxDB Database Trace
3. SAP MaxDB Interface Traces
4. Additional Traces
5. Useful Information Resources

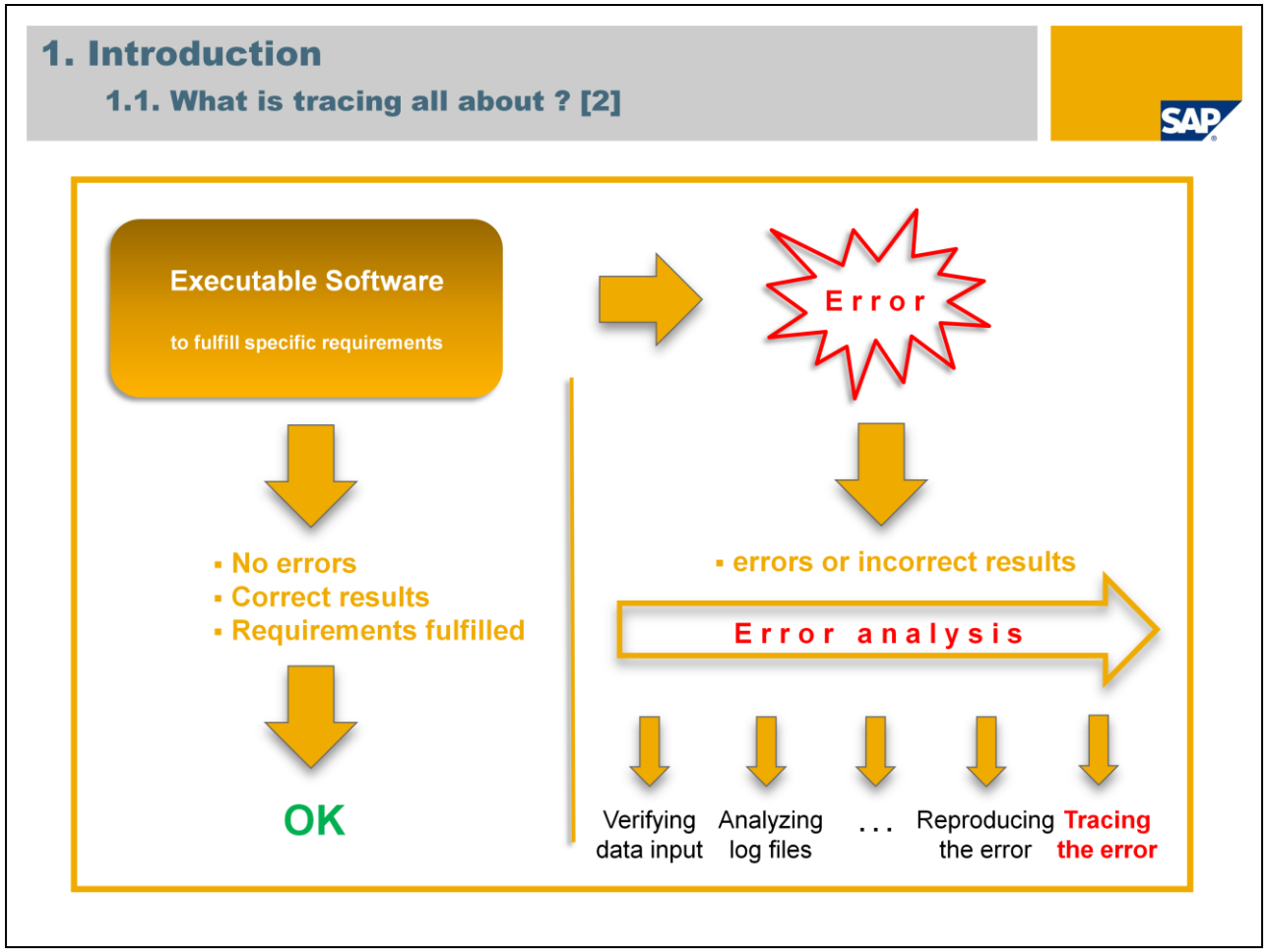
1. Introduction

1.1. What is tracing all about ? [1]



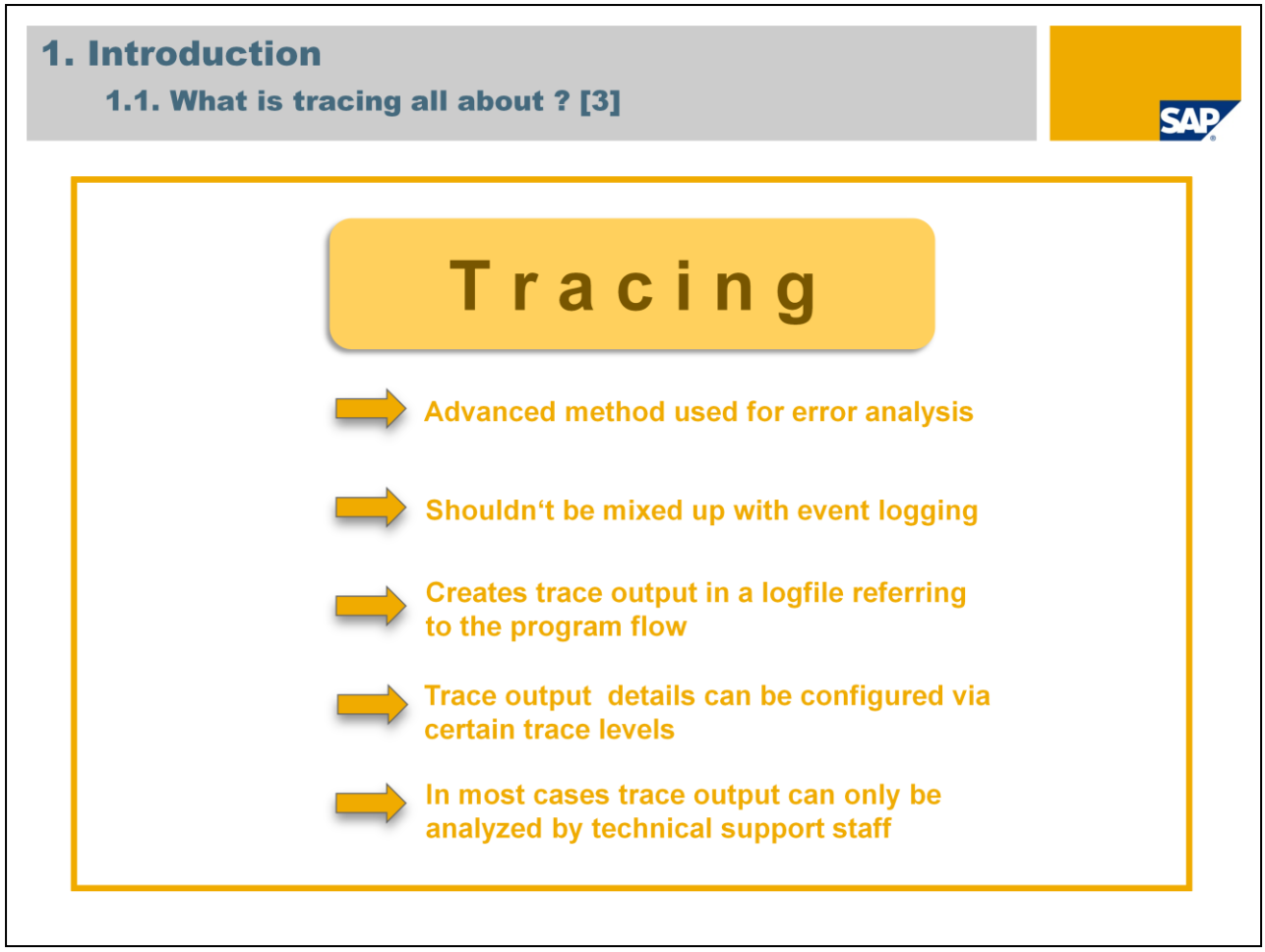
Creating business and database software is a complex process.

Such software products consist of different components which have to act in concert.



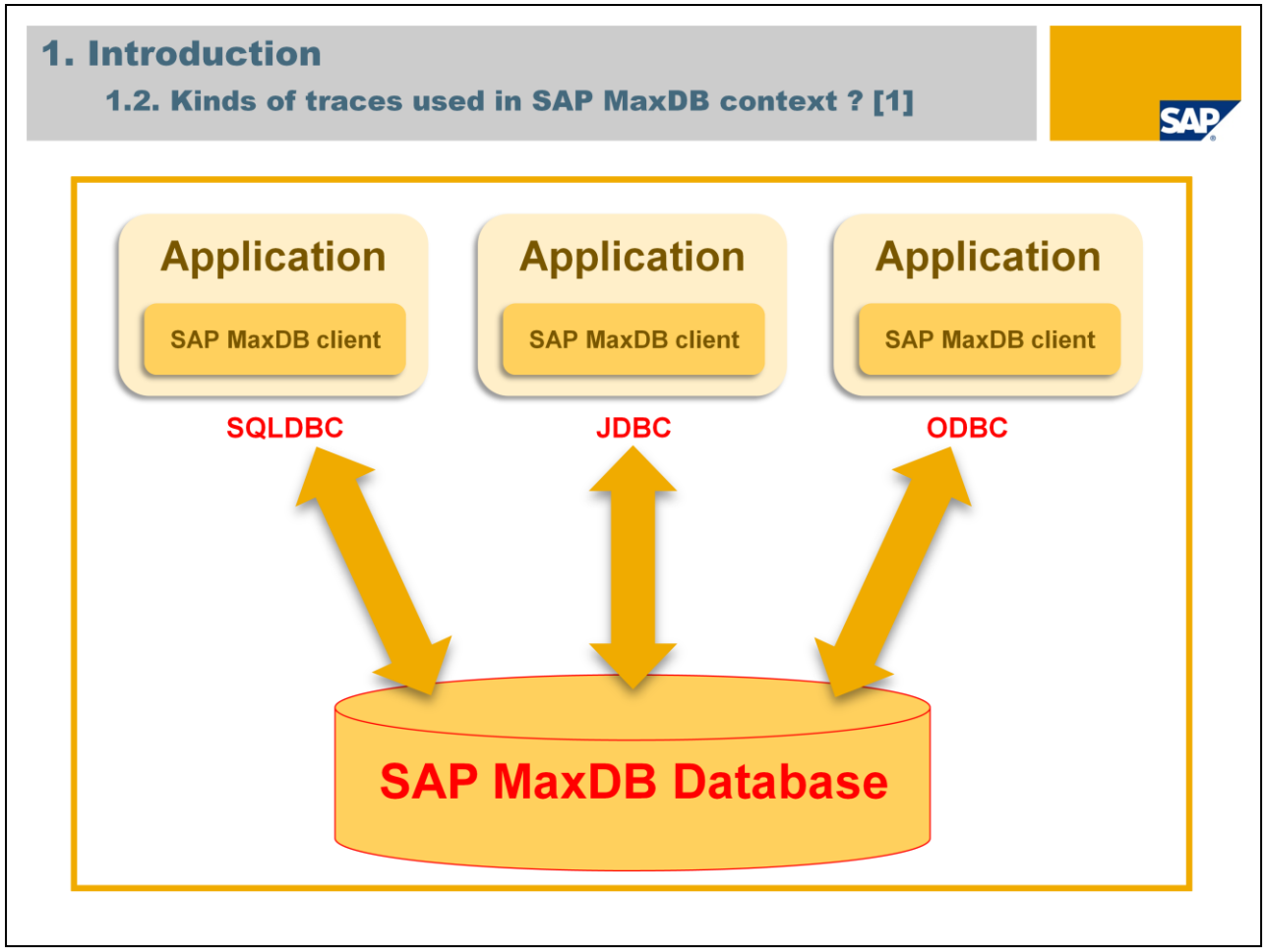
Complex software is error-prone. Sometimes the root causes of errors cannot be figured out without so-called “tracing“.

Tracing is a matter of advanced error analysis.



Tracing or creating a “trace“ is used for diagnostics of software. It creates entries in a trace logfile during the execution of a program. The level of details regarding these entries mostly can be configured by certain trace levels. A trace usually contains more extensive output than event logging. On the other hand, analyzing the trace output requires experienced technical staff mostly.

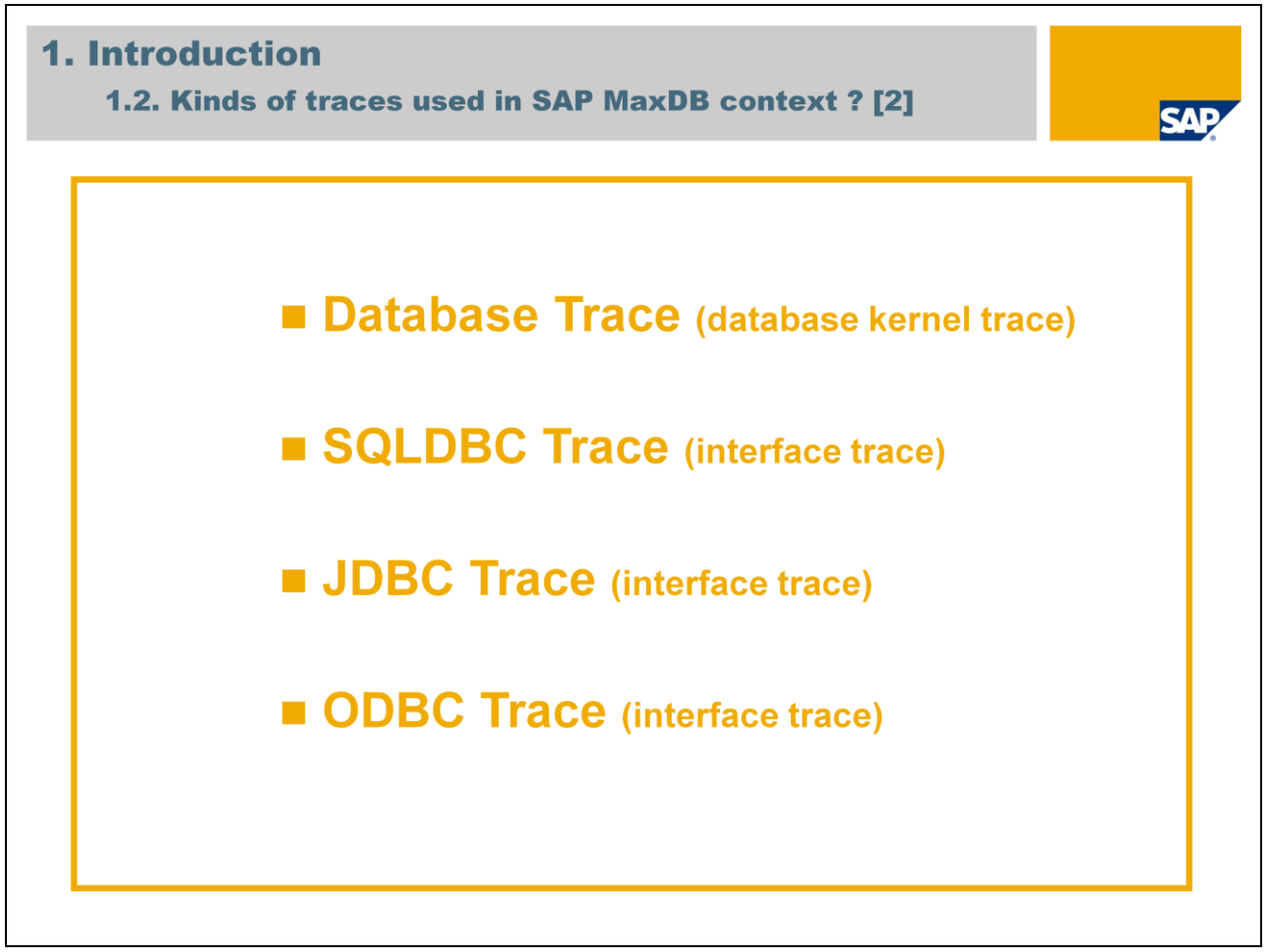
The trace output e.g. can contain information about entering specific parts of the program (e.g. functions or procedures) and which arguments have been transferred. This information can guide the specialist to an error within the pogram code.



Access to a MaxDB database from application side is carried out via certain interfaces. For these interfaces corresponding MaxDB software drivers are avaiable which are part of the MaxDB client software. Each drivers provides an own trace functionality.

The specific interface trace functionality is used if the error is supposed to occur on application side or on its way to the database kernel (via the interface). Knowing which interface is used is a precondition.

The MaxDB database kernel itself provides various trace options to create trace output for the analysis of errors which are supposed to happen there.



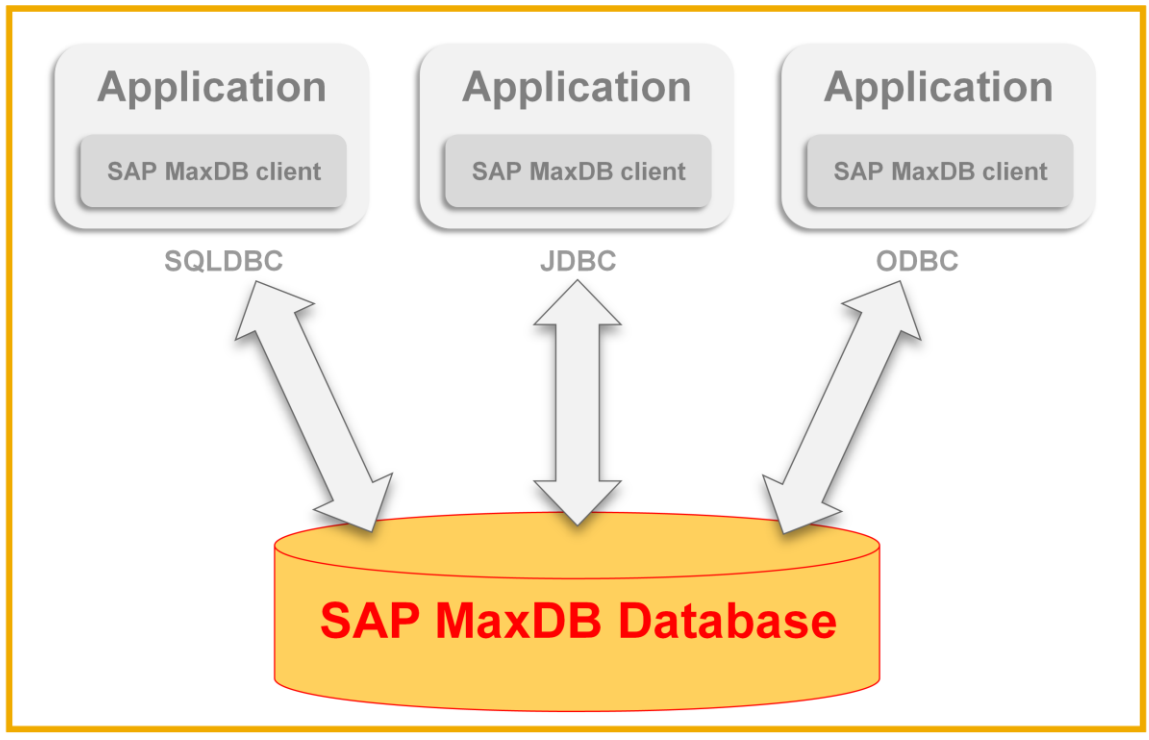
In addition to the SAP MaxDB Database Trace which logs the activity of the database kernel itself there are three interface traces. According to the specific interface they are called SAP MaxDB SQLDBC Trace, SAP MaxDB JDBC Trace and SAP MaxDB ODBC Trace.

Agenda

1. Introduction
2. SAP MaxDB Database Trace
3. SAP MaxDB Interface Traces
4. Additional Traces
5. Useful Information Resources

2. SAP MaxDB Database Trace

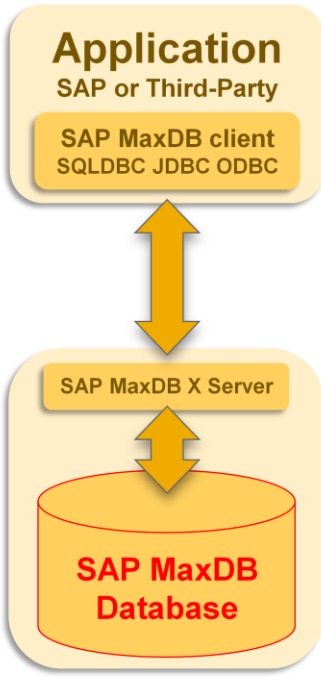
2.1. At a glance [1]



SAP MaxDB Database Trace logs the activity of the database kernel itself. It shows e.g. which kernel commands and which kernel modules were processed as well as performed steps in detail and the results / the success / errors of these activities.

2. SAP MaxDB Database Trace

2.1. At a glance [2]



When it is indicated to use this trace?

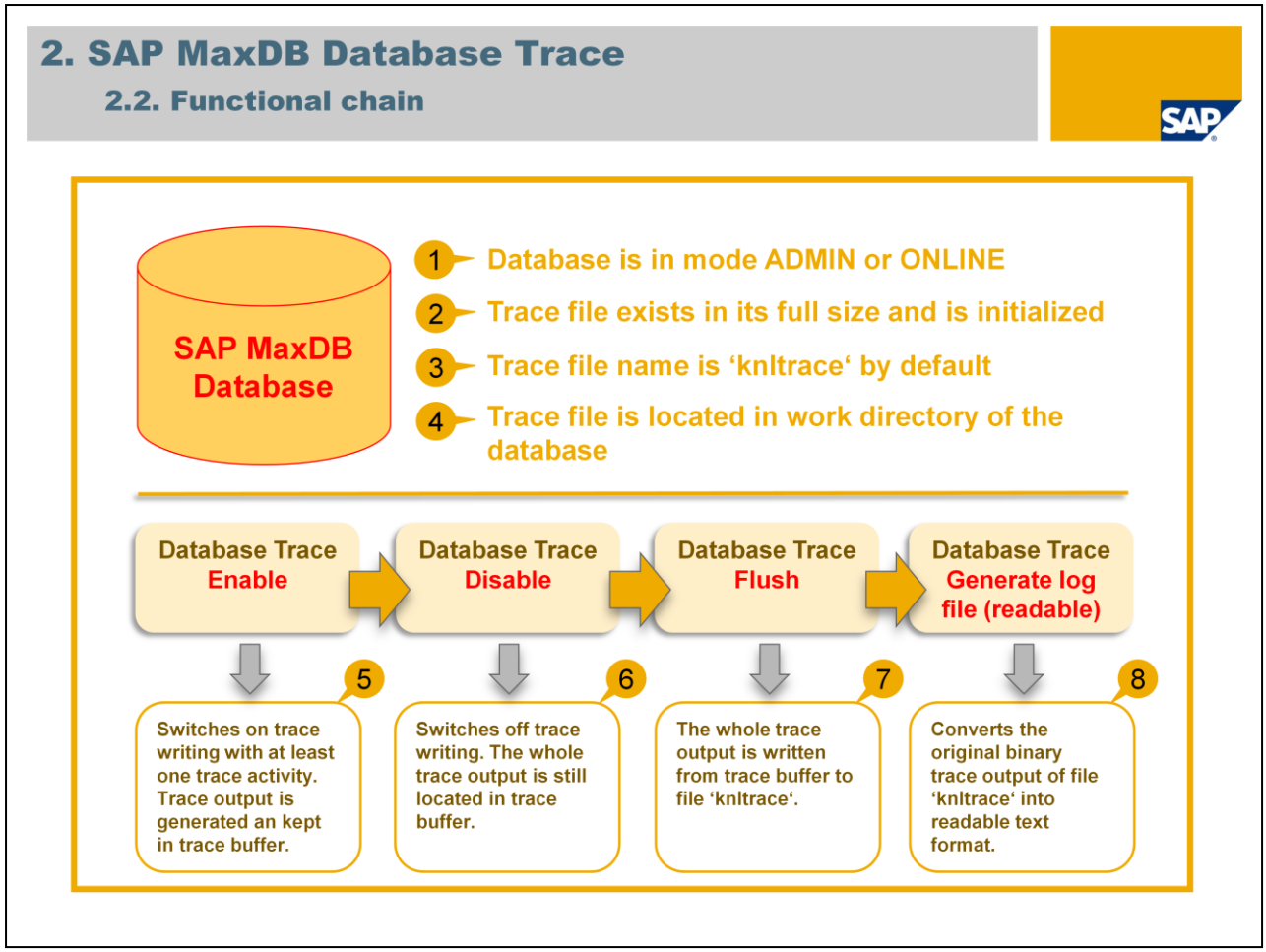
- An error occurs which most likely happens in database kernel
- Examples: SQL error, Table / Index error
  Catalog error

Which tools can be used to activate?

SAP MaxDB Tools	SAP Transactions
Database Studio	DB50 / LC10
Database Manager GUI	DBACOCKPIT
SQL Studio	
DBMCLI	
SQLCLI	

Where is the trace log file located?

- binary as well as readable trace log file are located in the work directory of the database instance (parameter RUNDIRECTORY)
- Example: /sapdb/data/wrk/<database name>



While starting the database to mode ADMIN or ONLINE a file named 'knltrace' is created in its full size. This size is calculated by the database kernel depending on other settings and cannot be changed. Size (in pages of 8KB) can be seen via parameter KernelTraceSize (1) (2).

The file name 'knltrace' is a default value and can be changed via parameter KernelTraceFile (3).

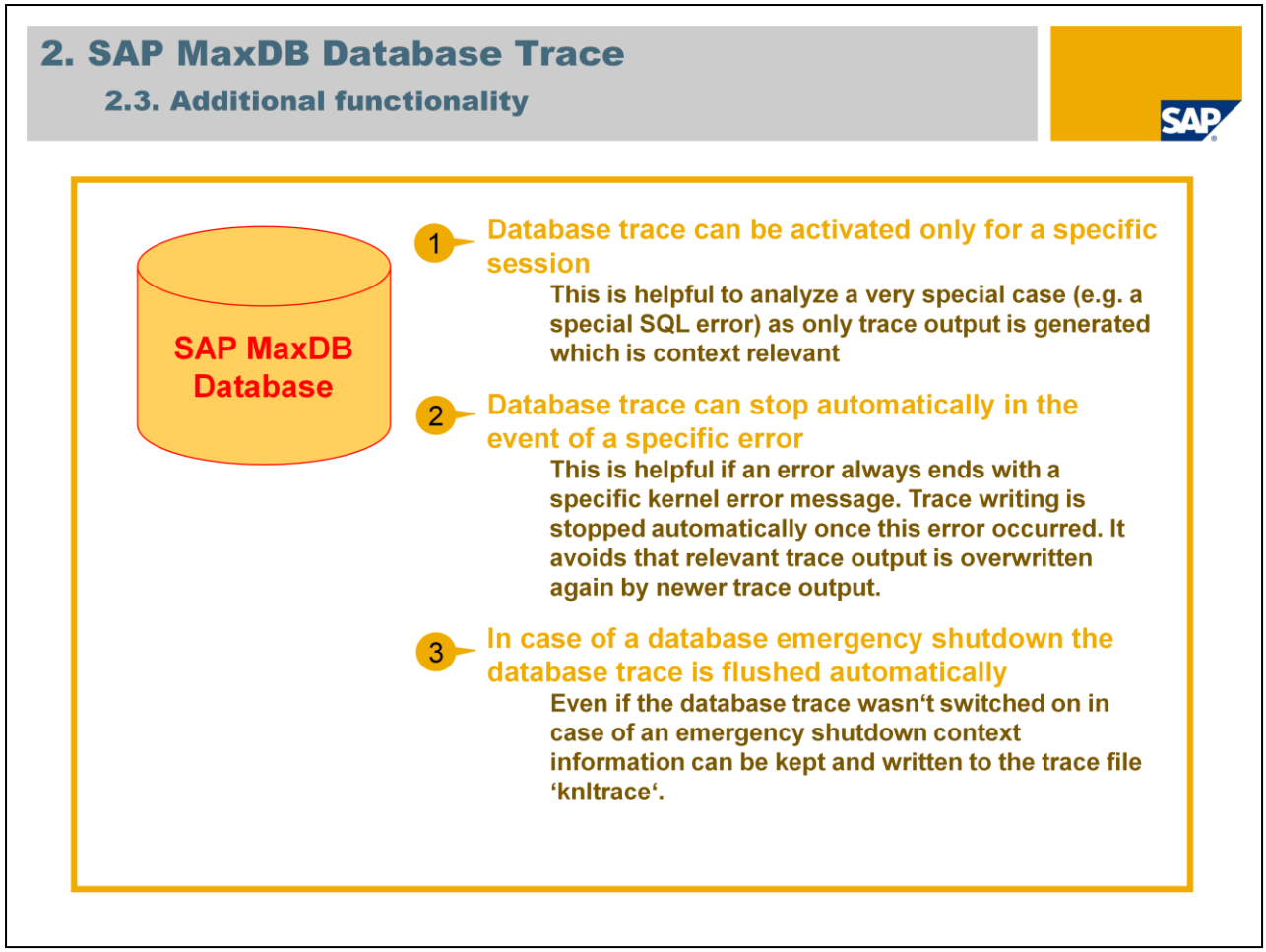
Location of the trace file is always the work directory of the database (set via parameter RUNDIRECTORY) (4).

When switching on the trace several trace activities can be set according to the context of the issue to be analyzed. Unless otherwise advised by SAP Support please always use activity DEFAULT. Once the trace is enabled trace output is written into a special area in data cache of the database, the trace buffer. The trace buffer is subdivided into sections which store the trace output of the different task types, e.g. user tasks, server tasks. The size of these sections can be configured via parameters. The trace buffer is written round robin (5).

Switching off the trace just stops writing of trace output into the trace buffer. The trace buffer is still filled with the trace output generated by then (6).

As memory content is volatile it is required to transfer the content of the trace buffer into file 'knltrace'. This is done by step 'flush' and this is very important. It is possible to flush while the trace is still enabled. In this case the latest content of trace buffer is transfered and trace writing is carried on (7).

The trace output kept in the trace buffer and transfered to file 'knltrace' exists in binary format. So it is hard to read and to analyze. That's why it has to be converted into readable text format. This step creates a new text file. The original file 'knltrace' remains unaffected (8).

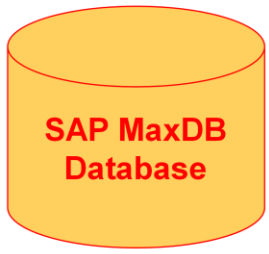


The capabilities to generate the trace output only for a certain session or to stop trace writing when a certain error occurred are quite helpful to analyze special cases (1) (2).

In case of an emergency shutdown of the database the trace file can contain useful context information of the case. So it should be saved by all means. After an emergency shutdown a directory named <Database name>_<date>_<time> is created automatically at next restart in directory <RUNDIRECTORY>/DIAGHISTORY (e.g. /sapdb/data/wrk/DB1/DIAGHISTORY/DB1_20120307_15-20-35). Here important log files, including file ‘knltrace‘, are saved (3).

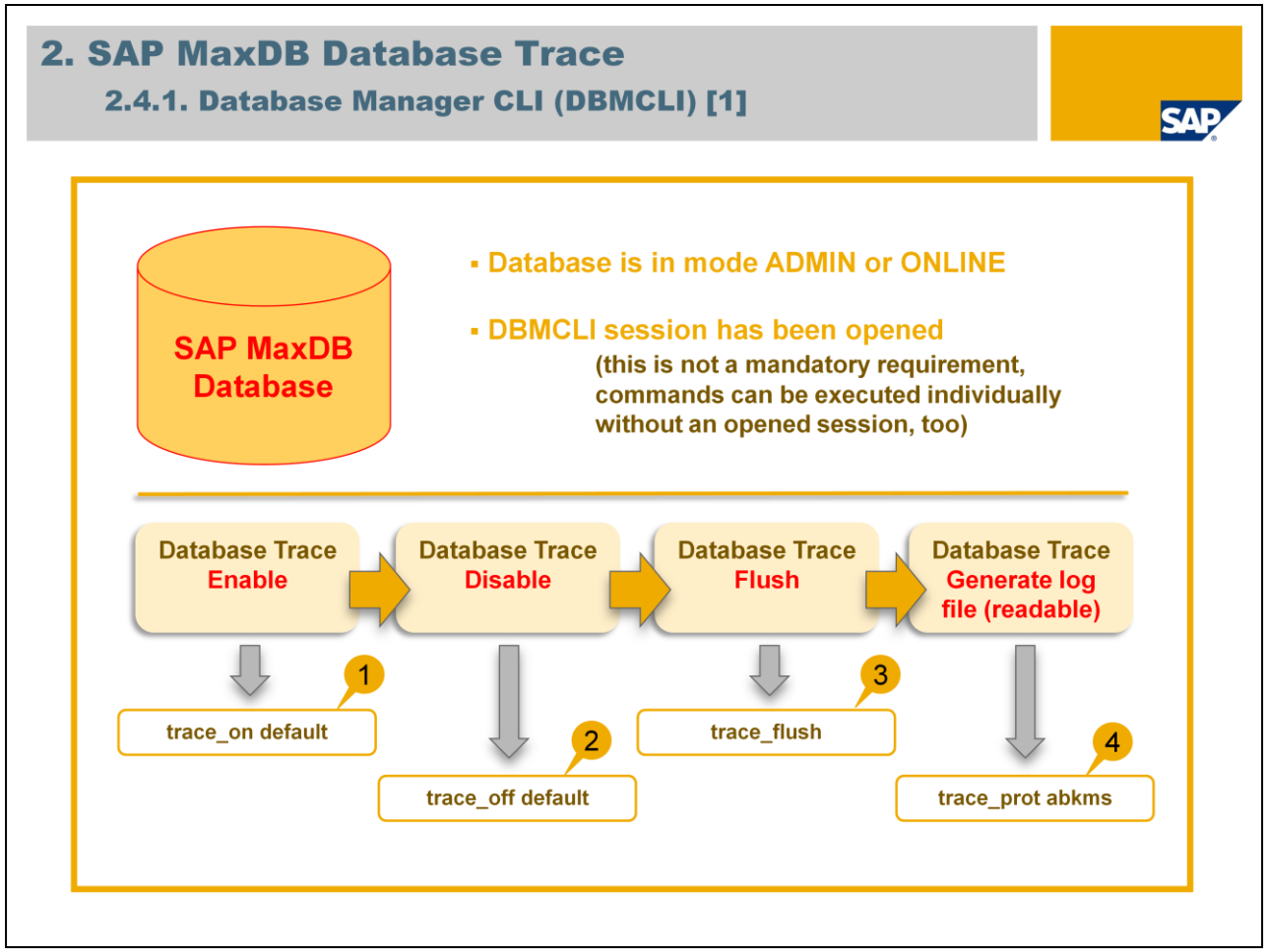
2. SAP MaxDB Database Trace

2.4. Using SAP MaxDB Tools for database tracing



Database Trace
is supported by MaxDB's own tools:

- Database Manager CLI (DBMCLI)
- Database Manager GUI (DBMGUI)
- SQL Studio / SQLCLI
- Database Studio



Using the Database Trace functionality is possible via DBM Server commands. The command to enable the trace is ‘trace_on default‘. Command element ‘default‘ is the trace activity and is sufficient in most cases. If other activities have to be set it will be advised by MaxDB Support (1).

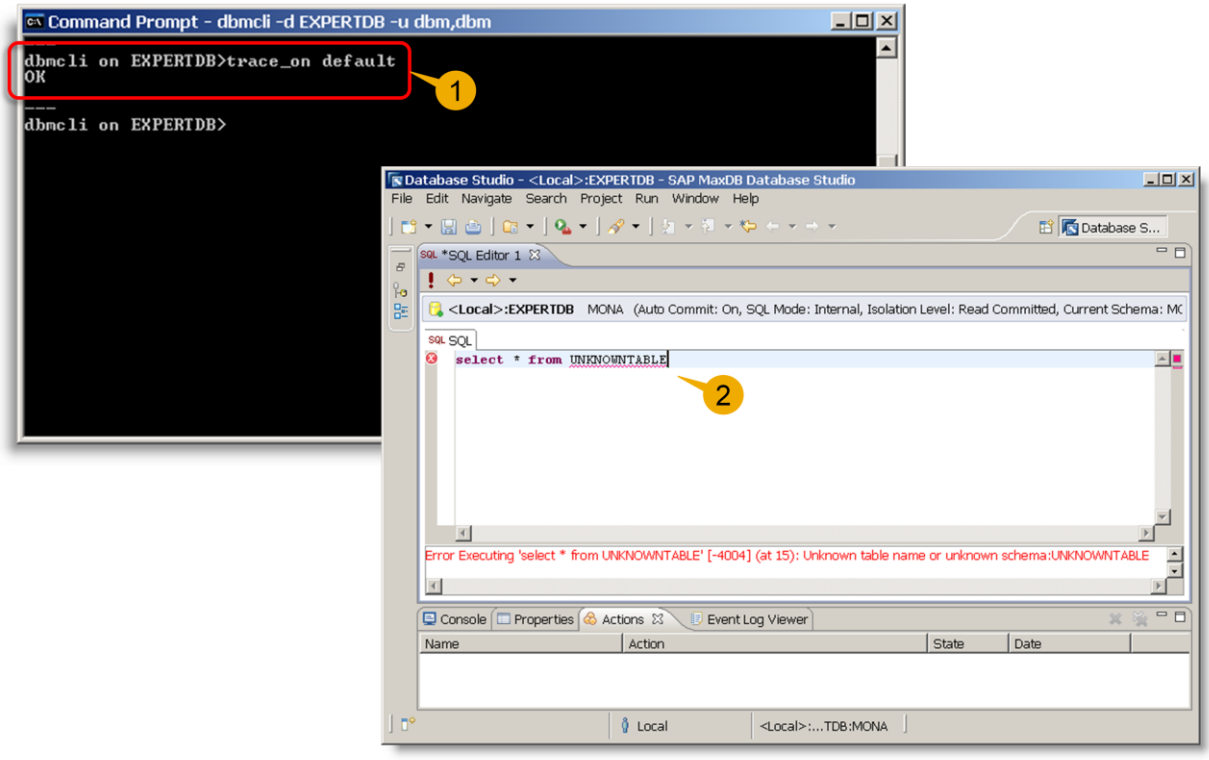
Once the trace is enabled the issue to be analyzed has to be started. When it is finished or an error occurred the trace should be switched off again to avoid generating unnecessary trace output. Command to disable the Database Trace is ‘trace off default‘ (2).

As mentioned before trace output is kept in a special area of the data cache. It has to be saved physically in file ‘knltrace‘. That‘s the purpose of command ‘trace_flush‘. This command can also be executed while the trace is still enabled (3).

As file ‘knltrace‘ exists in binary format and is not human-readable it has to be converted in text format. Command ‘trace_prot abkms‘ does this job. The options for the trace file generation at the end of the command (‘abkms‘) are sufficient for most cases. If other options have to be set a corresponding requirement will be given (4).

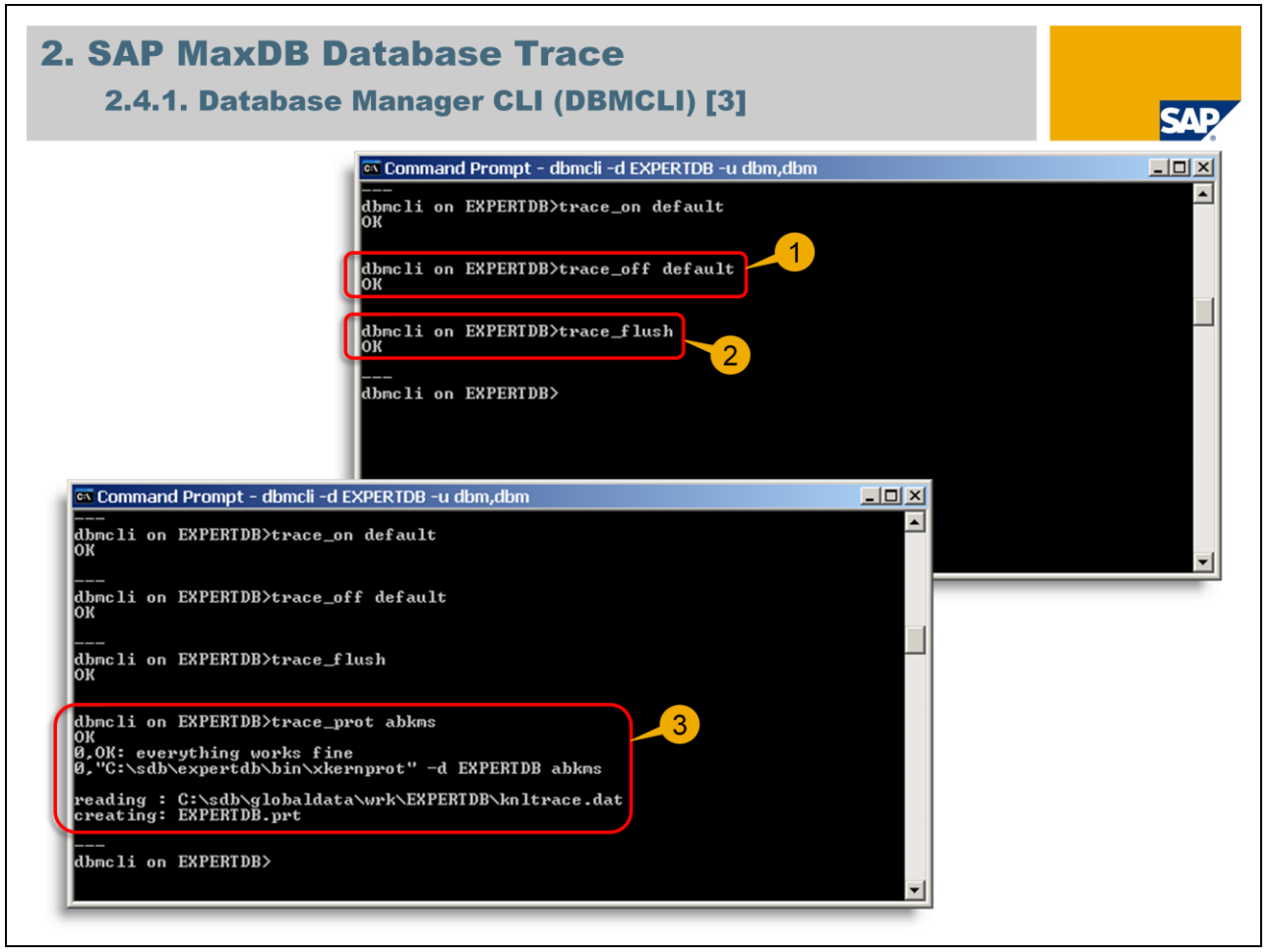
2. SAP MaxDB Database Trace

2.4.1. Database Manager CLI (DBMCLI) [2]



Database trace is enabled. Trace output of database kernel‘s work is generated. In fact this output is generated for every connected user from any connected client. It comprises the entire kernel work (1).

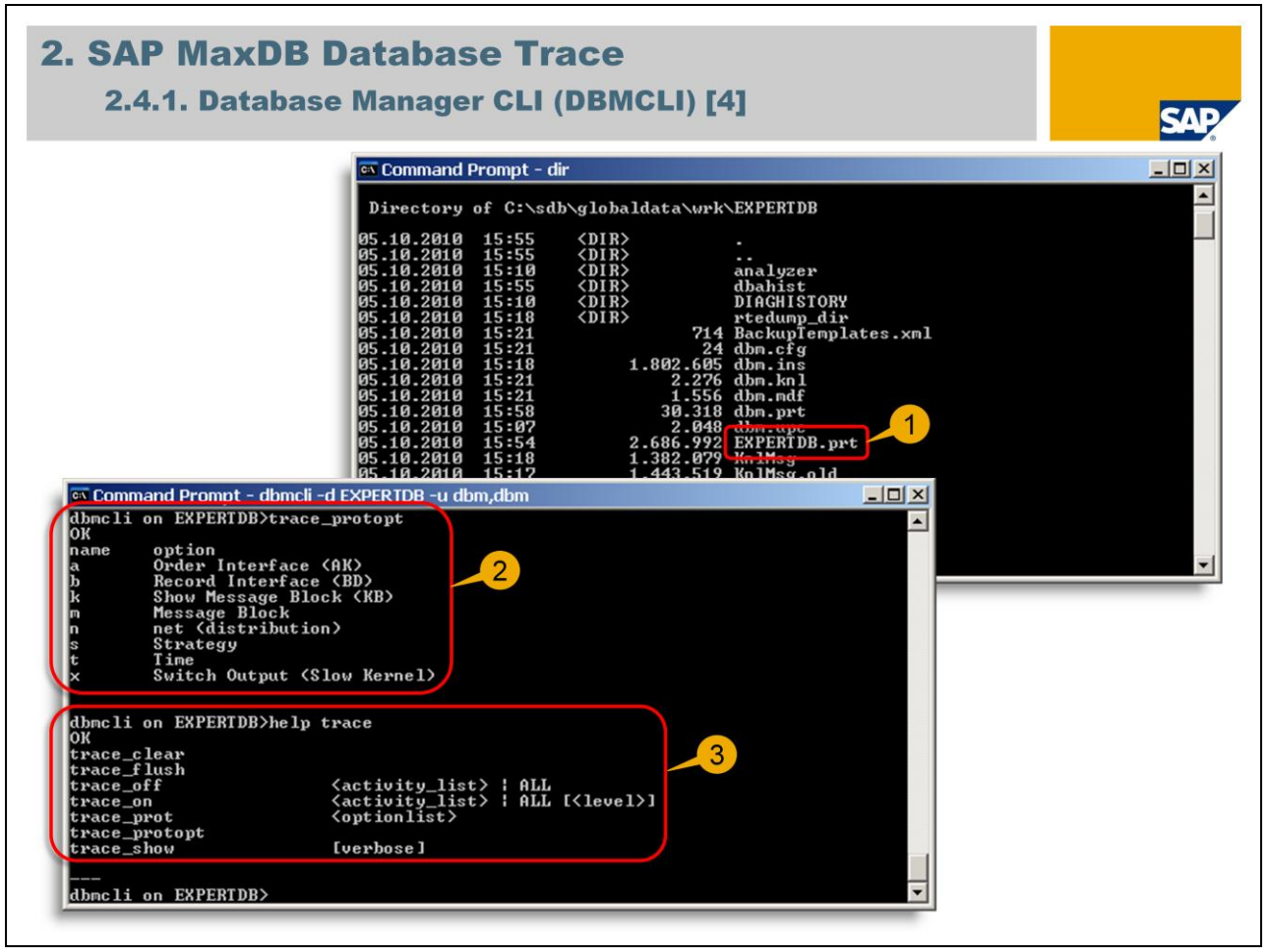
Afterwards the error producing issue is repeated, e.g. as seen in the very basic example of selecting from an unknown table (2).



After error or procedure to be analyzed has been executed database trace is switched off again to avoid generating unnecessary trace output (1).

Now the trace output is transferred as it is to file ‘knltrace‘ (2).

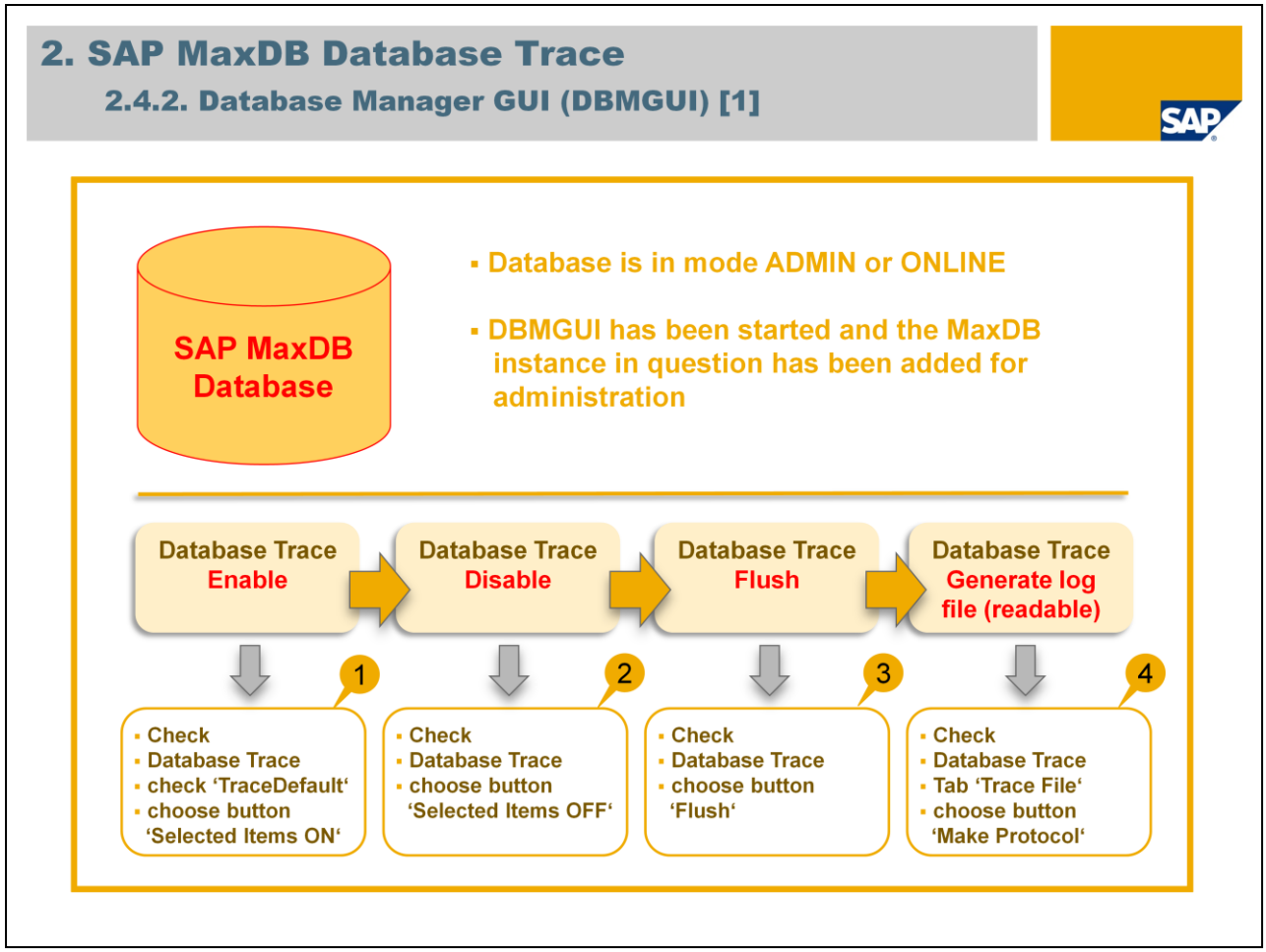
To get a human-readable file of the trace output command ‘trace_prot abkms‘ is executed finally. In a first step trace output is sorted and file ‘knltrace.dat‘ is created. This file is the input for creating the final trace logfile in text format. This file is named ‘<SID>.prt‘ (in our example ‘EXPERTDB.prt‘. This file can be opened with every text viewer (3).



The final trace logfile ‘EXPERTDB.prt‘ is located in the work directory of the database (set via parameter RUNDIRECTORY). In our example it is /sdb/globaldata/wrk/EXPERTDB. This file can be analyzed now or (which is recommended) can be passed on to a technical support expert (1).

As said before in most cases trace options ‘abkms‘ are sufficient. To see all available options command ‘trace_protopt‘ can be used. Anyway, this information is useful for an expert only. Unless otherwise indicated options should be set as mentioned before (2).

To see all available DBM Server commands regarding the database trace command ‘help trace‘ is useful (3).

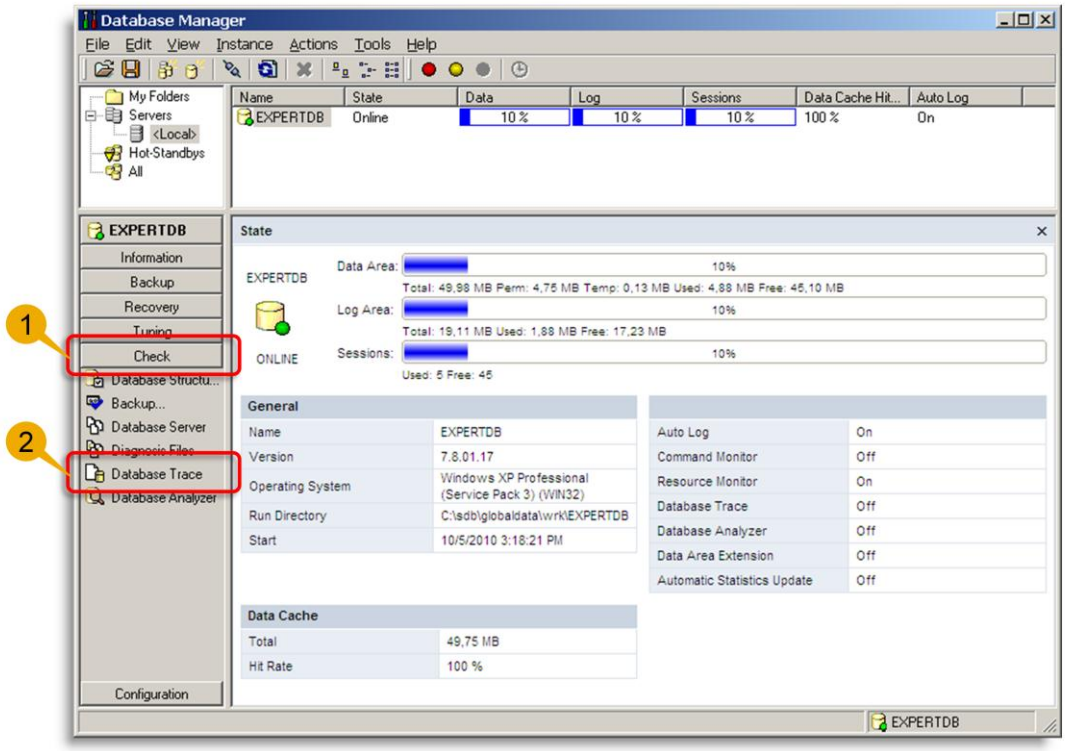


Database Manager GUI is the former graphical administration tool. It can be considered as the graphical user interface for dbmcli commands.

To enable, disable and flush the Database Trace as well as generate the human-readable trace log file certain buttons, check marks, list entries and tabs have to be used (1) (2) (3) (4).

2. SAP MaxDB Database Trace

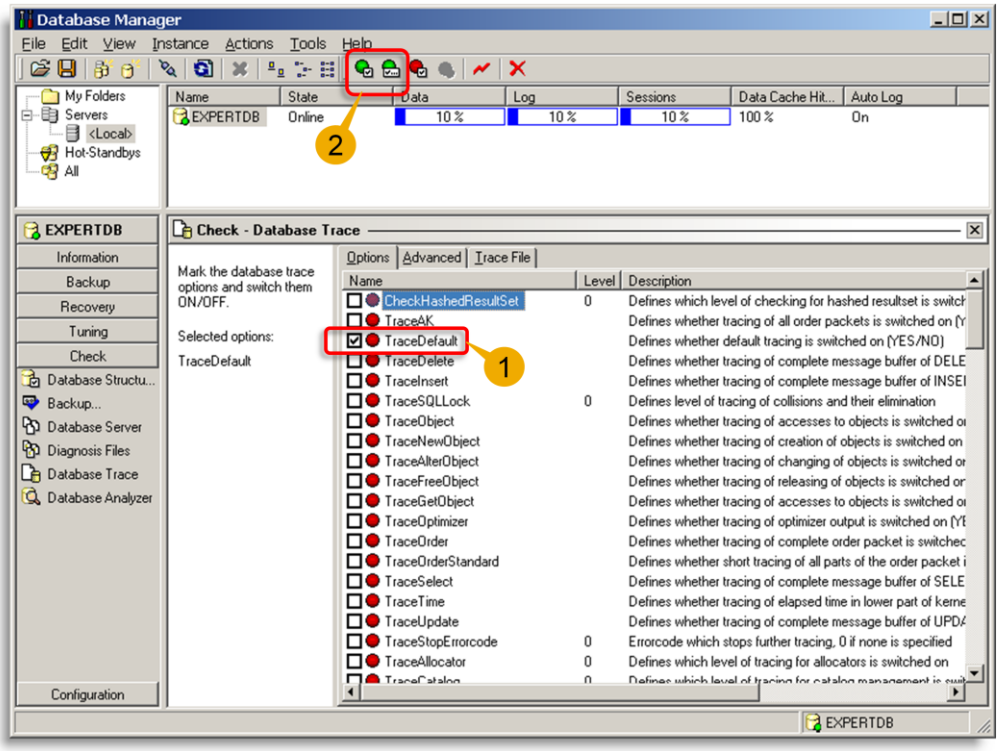
2.4.2. Database Manager GUI (DBMGUI) [2]



To reach the area of Database Trace within Database Manager GUI at first ‘Check‘ and ‘Database Trace‘ have to be chosen. Both on the left-hand side (1) (2).

2. SAP MaxDB Database Trace

2.4.2. Database Manager GUI (DBMGUI) [3]

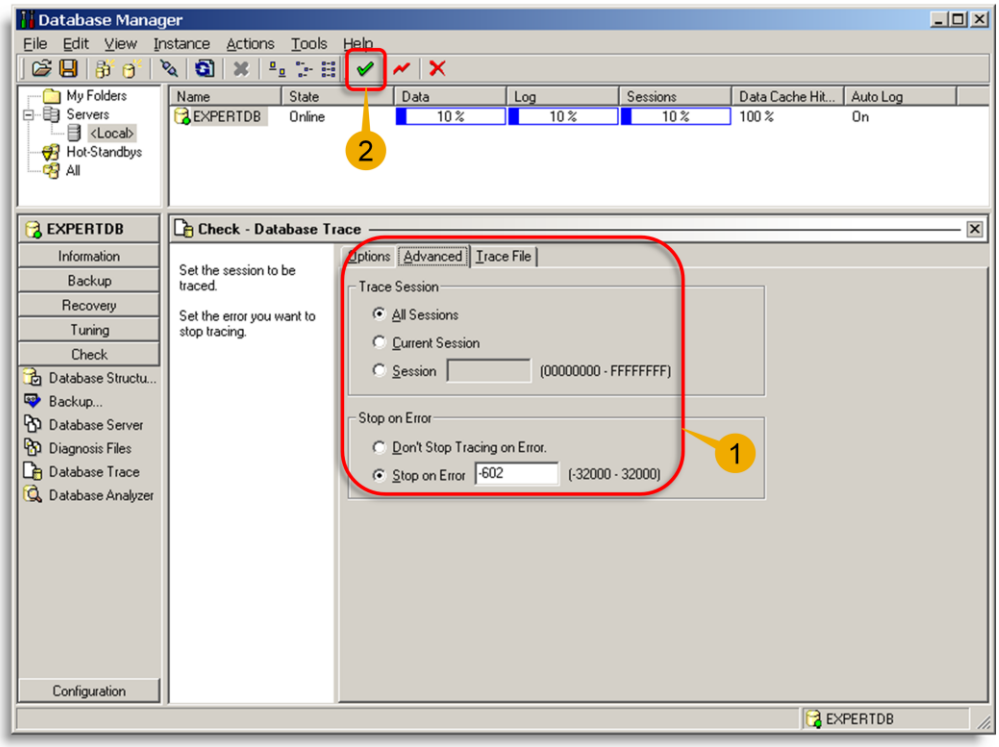


Via tab ‘Options‘ (which is selected automatically at first) a list of all available trace activities is shown. Those which are enabled show a green colored icon, disabled ones a red colored icon. As mentioned before in most cases ‘default‘ is sufficient. In the activity list here it is called ‘TraceDefault‘ and has to be checked. Some trace activities are interdependent. This also applies for ‘TraceDefault‘. When enabling this activity others are enabled implicitly. (1).

Next is to enable the trace with this setting. This is done via the icon ‘Selected Items ON‘ within the toolbar on top. There is a similar icon next to it which is called ‘Selected Items ON With Level …‘. For some trace activities it is possible to activate different trace levels. This is only relevant if you are advised accordingly by a Support Expert. (2).

2. SAP MaxDB Database Trace

2.4.2. Database Manager GUI (DBMGUI) [4]

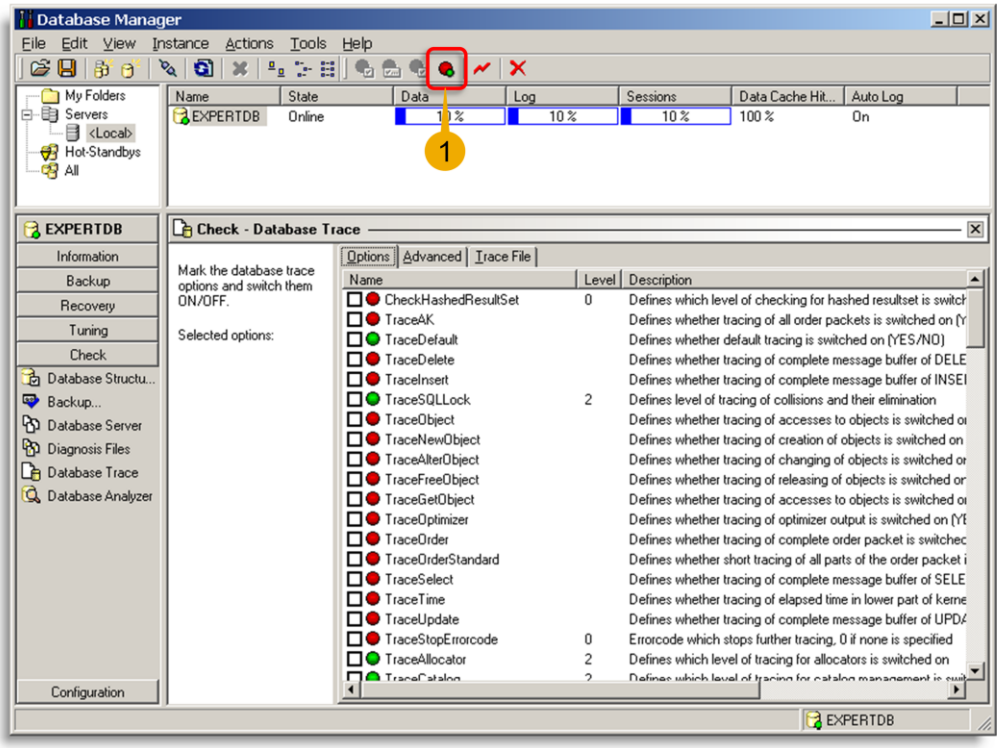


Tab ‘Advanced‘ allows to make special settings. So it is possible to create trace output for a particular session only. On the other hand the trace can be stopped automatically in the event of a certain error. Both settings can be combined (1).

These settings take effect once button ‘Set Advanced Options‘ is used (2).

2. SAP MaxDB Database Trace

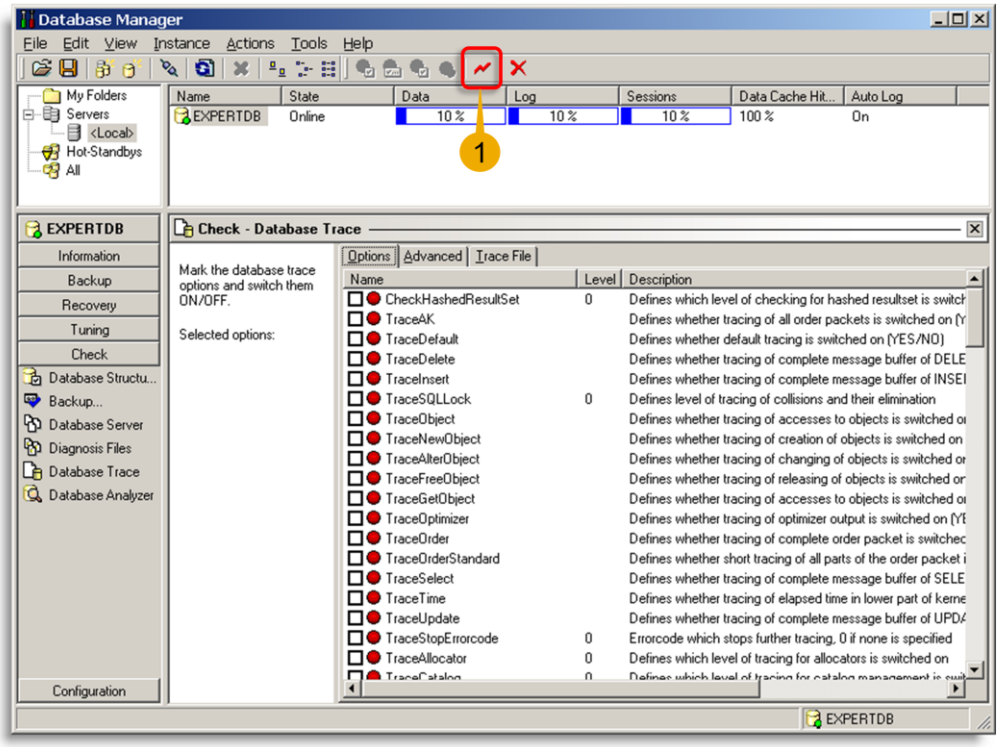
2.4.2. Database Manager GUI (DBMGUI) [5]



After error or issue to be analyzed has been executed Database Trace is switched off again by choosing button ‘Selected Items OFF‘. The icons of all trace activities are red colored again afterwards (1).

2. SAP MaxDB Database Trace

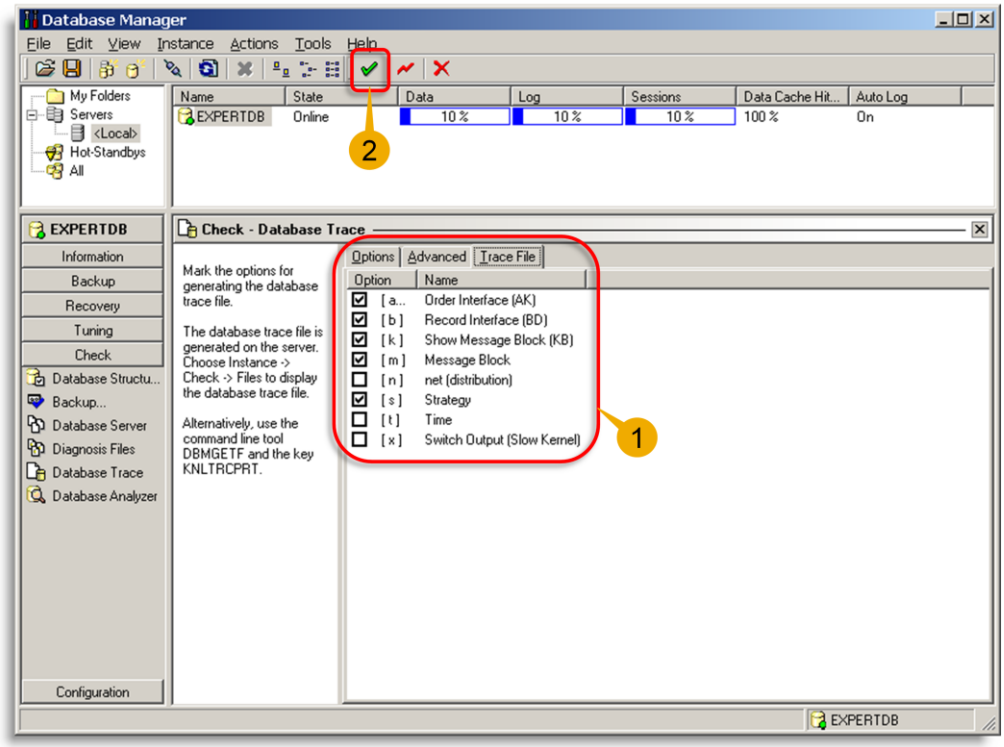
2.4.2. Database Manager GUI (DBMGUI) [6]



Now the trace output has to be transferred from the trace buffer in memory as it is to file ‘knltrace‘. This is done by choosing the icon which shows the red colored broken line within the toolbar. This icon is named ‘Flush‘. Please be careful and do not choose the icon next to it which shows the red colored cross (named ‘Clear‘) because this one is used to empty the trace buffer (1).

2. SAP MaxDB Database Trace

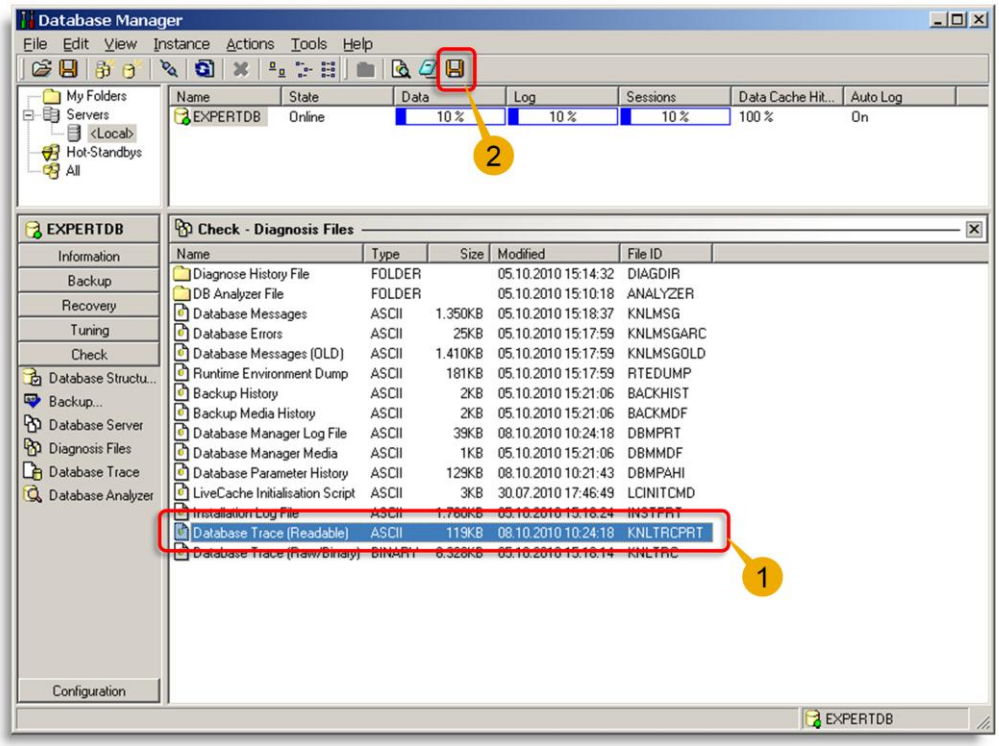
2.4.2. Database Manager GUI (DBMGUI) [7]



Finally the human-readable file of the trace output has to be generated. Options ‘abkms‘ are chosen (unless other advice by MaxDB support has been given) and the generation is started via button ‘Make Protocol‘ (1) (2).

2. SAP MaxDB Database Trace

2.4.2. Database Manager GUI (DBMGUI) [8]

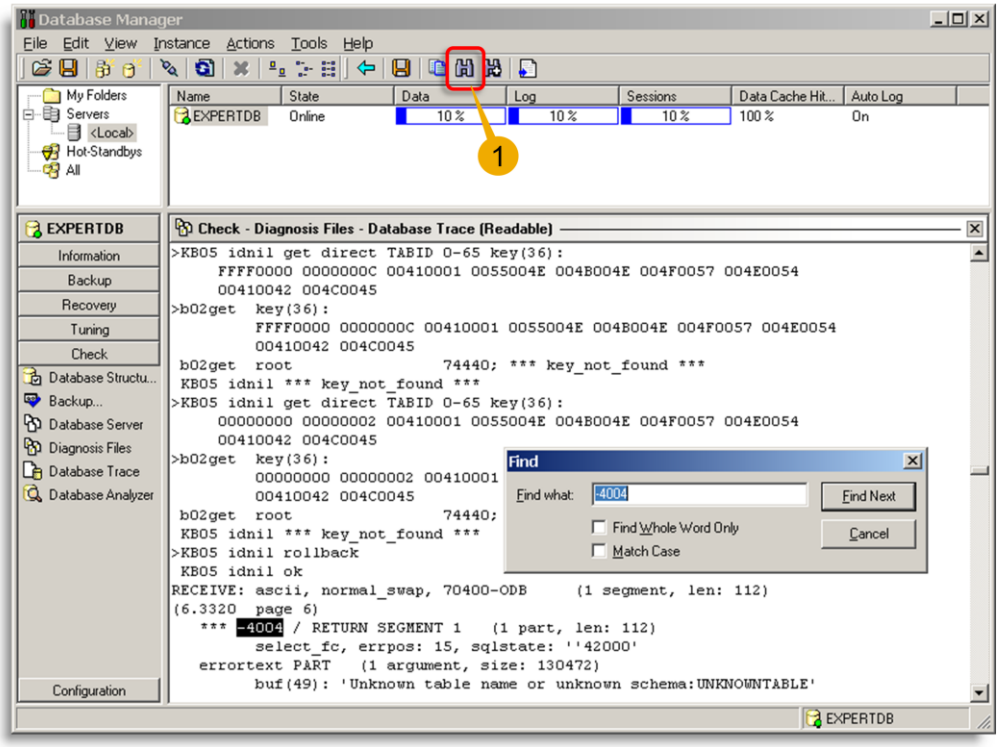


The file is named ‘<SID>.prt‘ (in our example ‘EXPERTDB.prt‘) and is located in the work directory of the database (set via parameter RUNDIRECTORY). To display files Database Manager GUI uses a DBM Server command which identifies the corresponding file via a file key (column File ID). For trace file ‘<SID>.prt‘ this key is named ‘KNLTRCPRT‘. So a double-click on this line opens the file (1).

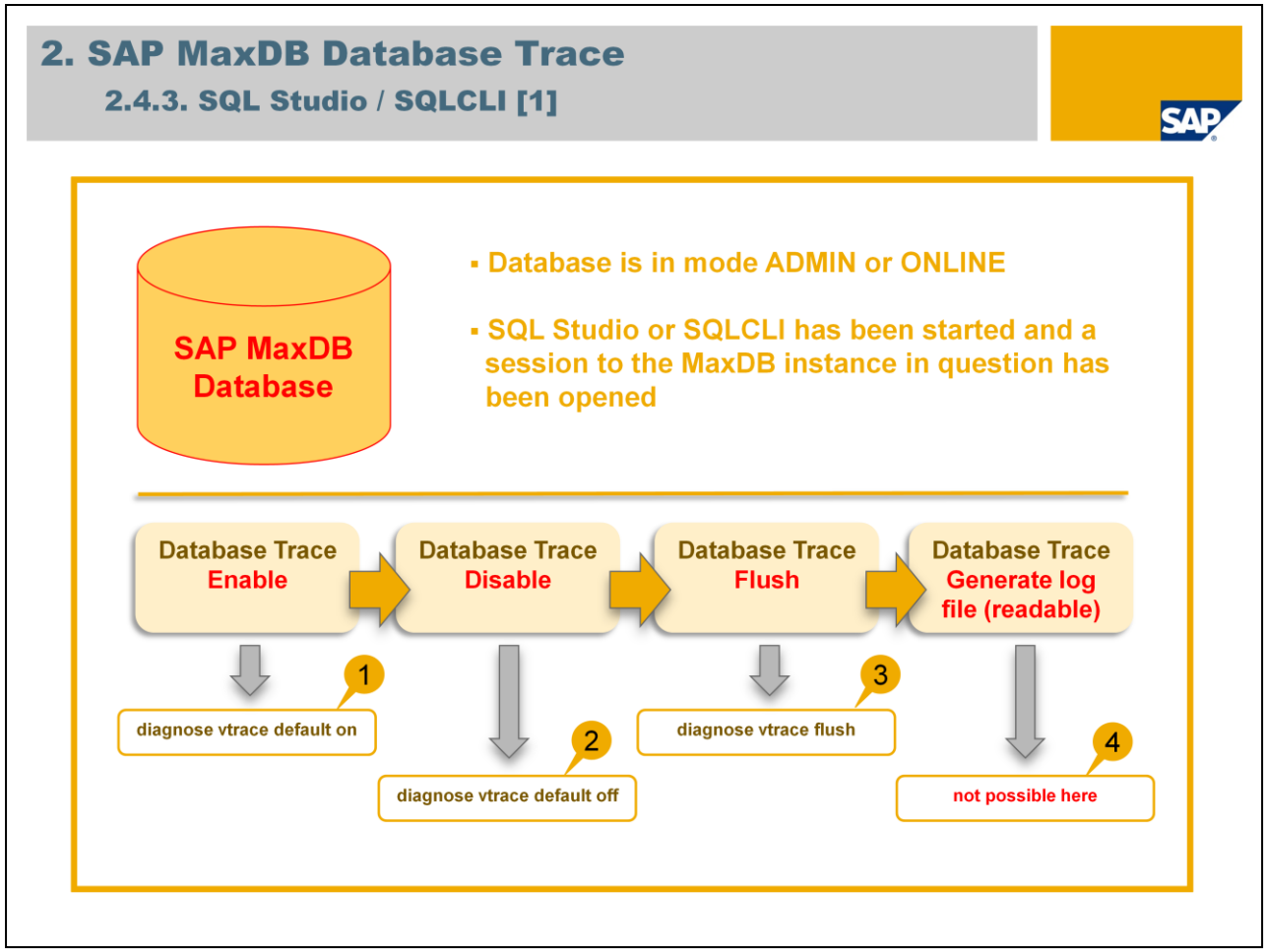
It is also possible to save the file directly by using the icon which displays the disk (2).

2. SAP MaxDB Database Trace

2.4.2. Database Manager GUI (DBMGUI) [9]



As mentioned before it is a tough job to read and understand the trace file for people who aren‘t experts. So this is definitely the job of the MaxDB experts. However, in case a certain error is analyzed you could check if the corresponding error has been caught within the trace which is a good preparation for the later analysis (1).

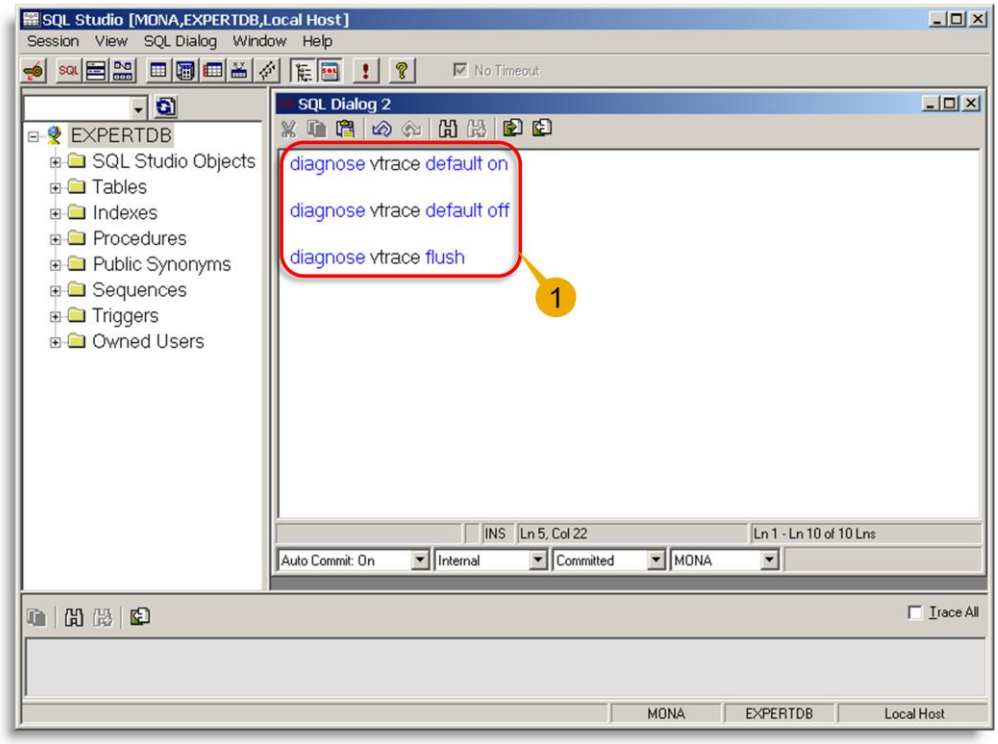


It is possible to enable, disable and flush the Database Trace via special SQL commands. That‘s why also MaxDB SQL clients can be used for this purpose. The term ‘vtrace‘ within the SQL commands is based on a historical notation and has survived here up to today (1) (2) (3).

Only generating the human-readable trace file is not possible via this way so this task is reserved for the administration tools (4).

2. SAP MaxDB Database Trace

2.4.3. SQL Studio / SQLCLI [2]

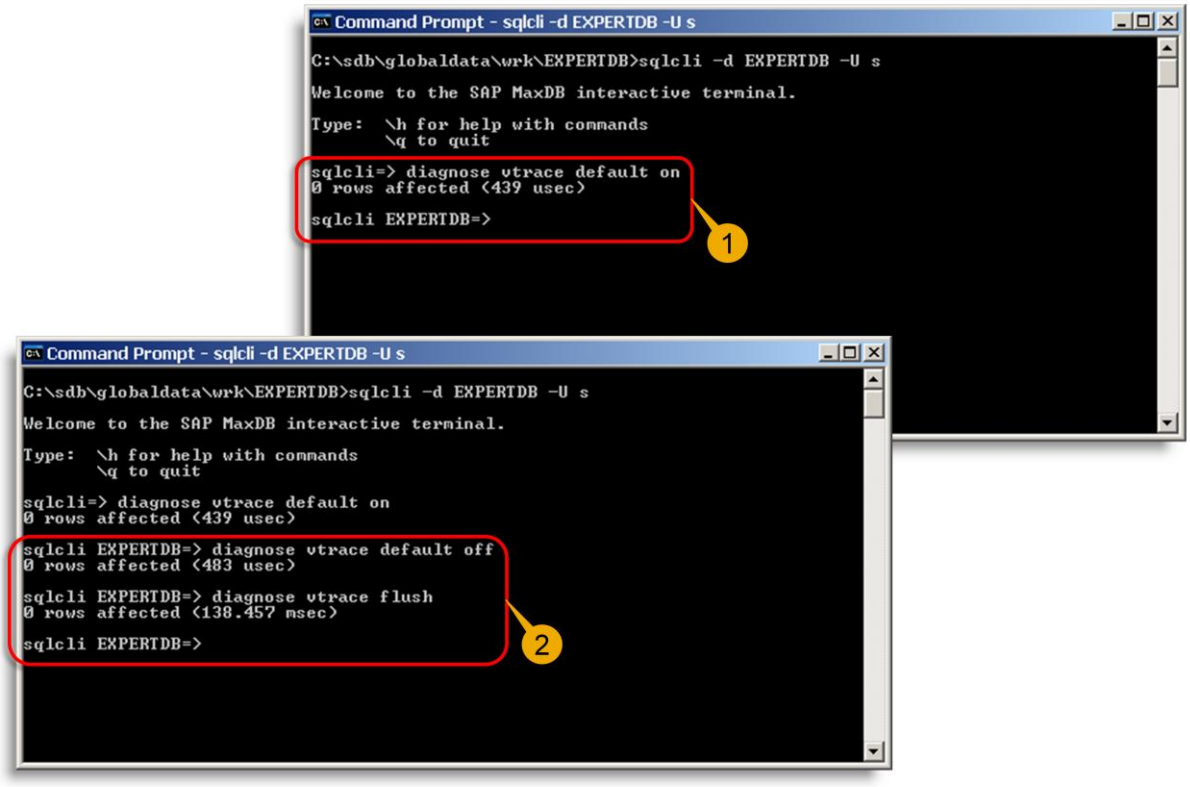


Once an SQL connection has been established using an SQL user of type DBA (to have the required permissions) it is possible to execute the SQL commands directly (1).

If the trace output has been flushed you can use DBMCLI (see slide ‘2.4.1. Database Manager CLI (DBMCLI) [3]’) or Database Manager GUI (see slide ‘2.4.2. Database Manager GUI (DBMGUI) [7]’) or Database Studio for generating the human-readable trace file.

2. SAP MaxDB Database Trace

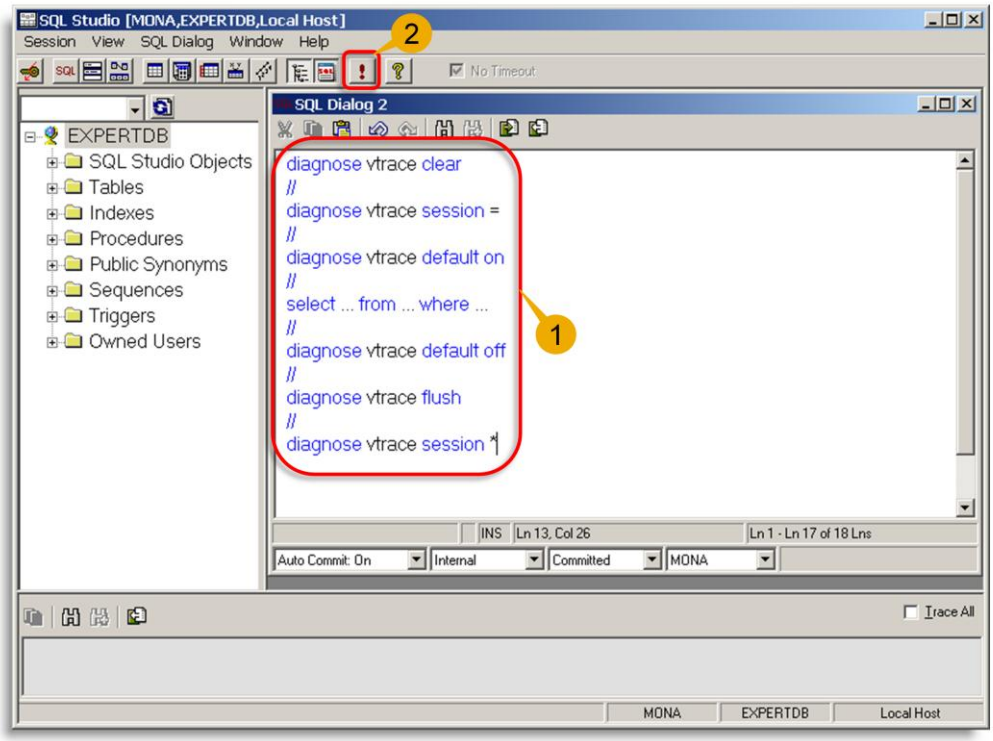
2.4.3. SQL Studio / SQLCLI [3]



The same SQL commands can be executed using command line tool SQLCLI for enabling (1), disabling and flushing the trace (2).

2. SAP MaxDB Database Trace

2.4.3. SQL Studio / SQLCLI [4]



SQL Studio (as well as the successor Database Studio) offers a quite convenient way to create a Database Trace of the current session (see slide '2.4.2. Database Manager GUI (DBMGUI) [4]'). It is not required to figure out the ID of the current session (opened via SQL Studio). It is well-suited to trace a certain SQL command.

In SQL Studio it is possible to execute a sequence of SQL commands. The SQL commands are separated by a line containing two slashes. The first command ('diagnose vtrace clear') will empty the trace buffer. The second one ('diagnose vtrace session =') is responsible for the advanced setting that the trace will be active for the current session only (the session where this command was carried out). The third SQL command switches the trace on with option 'DEFAULT' ('diagnose vtrace default on') .

Once this has been done the SQL command to be traced is executed (in the example above shown as 'select … from … where …').

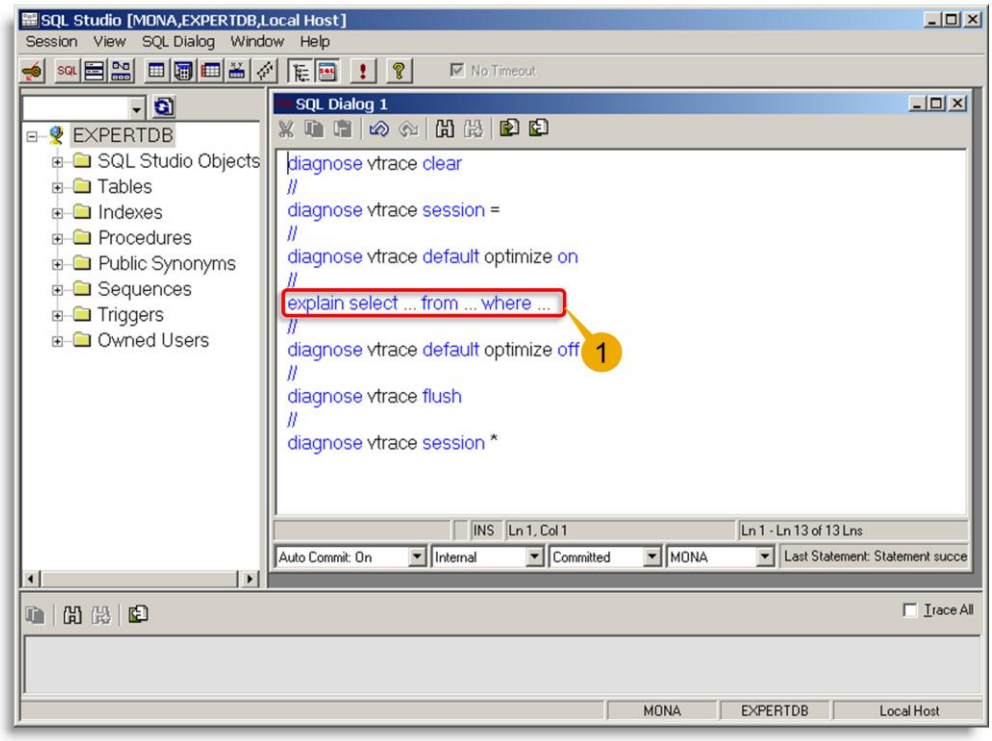
Then the trace is switched off again ('diagnose vtrace default off') and flushed ('diagnose vtrace flush'). Finally the trace is set for all sessions again (('diagnose vtrace session *') (1).

This sequence of SQL commands can be executed in one step by highlighting them as a block and afterwards choosing the icon showing the exclamation mark (2).

After all these steps file 'knltrace' is filled with the trace output of the execution of the certain SQL command which is going to be analyzed. The only task to be done is to create the human-readable trace file as mentioned before.

2. SAP MaxDB Database Trace

2.4.3. SQL Studio / SQLCLI [5]

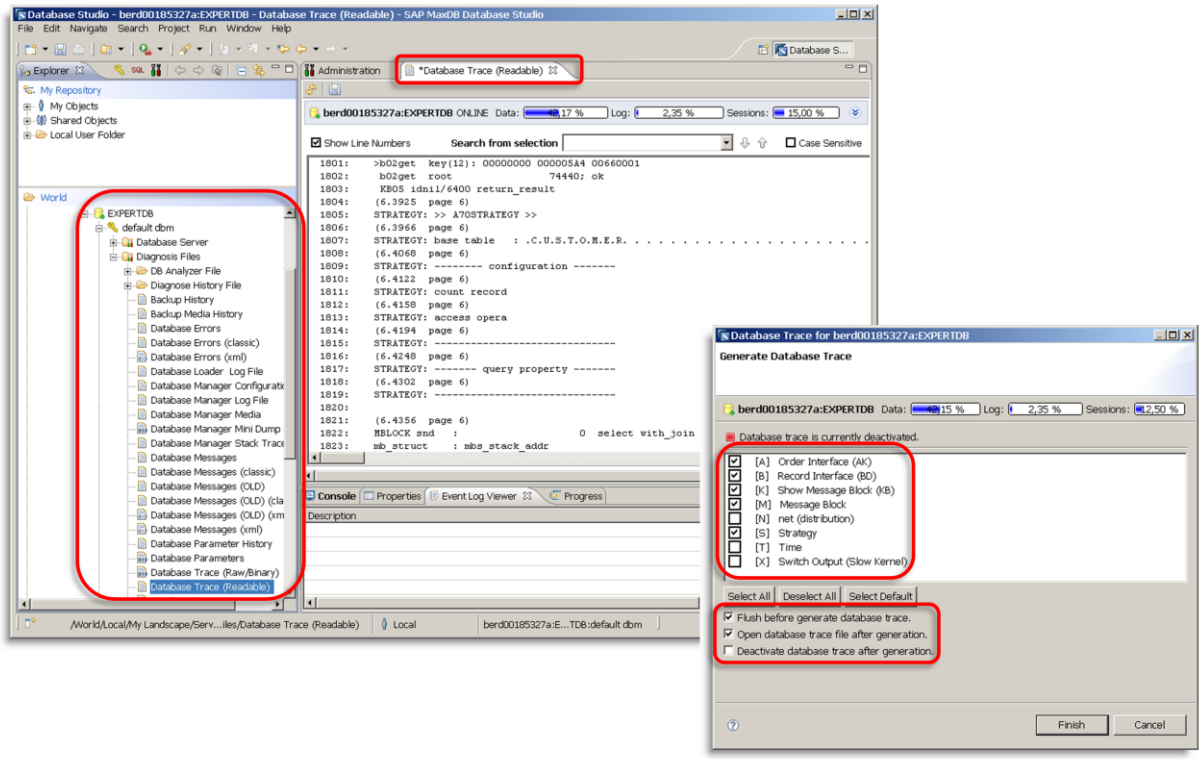


The same method can be used to trace the approach of the MaxDB SQL Optimizer for finding the optimal execution plan for a certain SQL command. To create this so-called ‘strategy trace’ it is necessary to add trace option OPTIMIZE and to set the key word ‘explain’ directly before the SQL command (1).

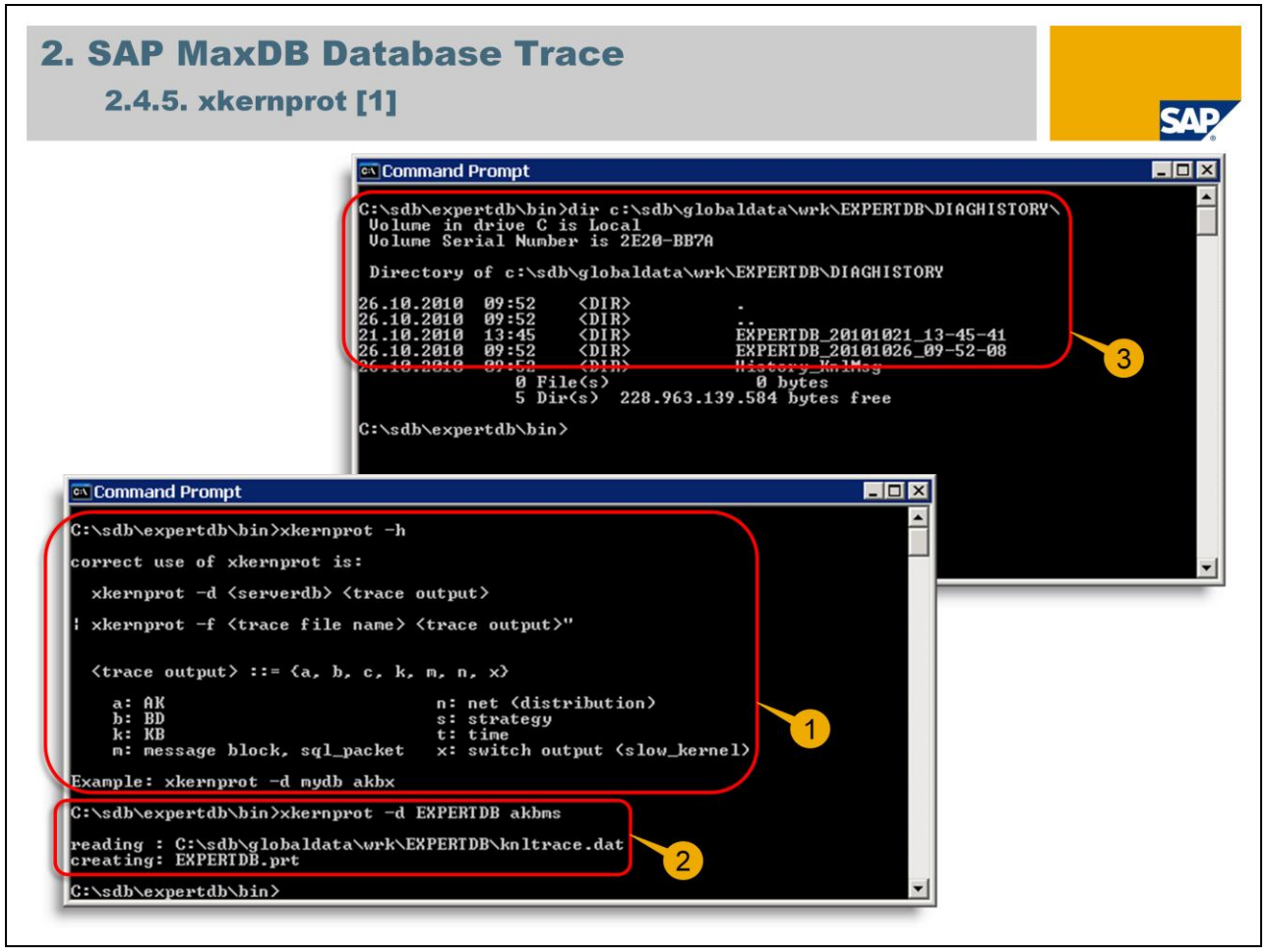
The trace output will contain only relevant entries regarding this action.

2. SAP MaxDB Database Trace

2.4.4. Database Studio



Database Studio is the current MaxDB administration tool which also includes a complete SQL client. It replaces the tools ‘Database Manager GUI‘ and ‘SQL Studio‘ as of MaxDB version 7.7. Database Studio offers the whole set of trace functionality from enabling up to creating, displaying and saving the human-readable trace file. Please refer to SAP MaxDB Expert Session 2: ‘Basic Administration with Database Studio’ which among others demonstrate the Database Trace functionality while using this tool.



SAP MaxDB command line tool ‘xkernprot‘ is the one which is finally called by the other tools like dbmcli, Database Studio or Database Manager GUI to generate the human-readable trace file.

It can also be used directly and is located in directory ‘<InstallationPath>/bin‘. The installation path can be determined via command ‘xinstinfo <SID>‘. To get an overview about the available options of ‘xkernprot‘ it can be executed with option ‘-h‘ (1).

To create the human-readable trace file based on file ‘knltrace‘ which is located in the work directory of the database (see output of command ‘xinstinfo <SID>‘ or parameter RUNDIRECTORY)

xkernprot is called with specifying the database name and the trace options. In a first step file ‘knltrace‘ is sorted and a second file named ‘knltrace.dat‘ is created (still in binary format). Using this file the final human-readable trace file named ‘<SID>.prt‘ is generated. It is also located in database‘s work directory (2).

In case of an emergency shutdown of the database important log files are saved during next restart in directory ‘<RUNDIRECTORY>/DIAGHISTORY/<SID>_<date>_<time>‘ (3).

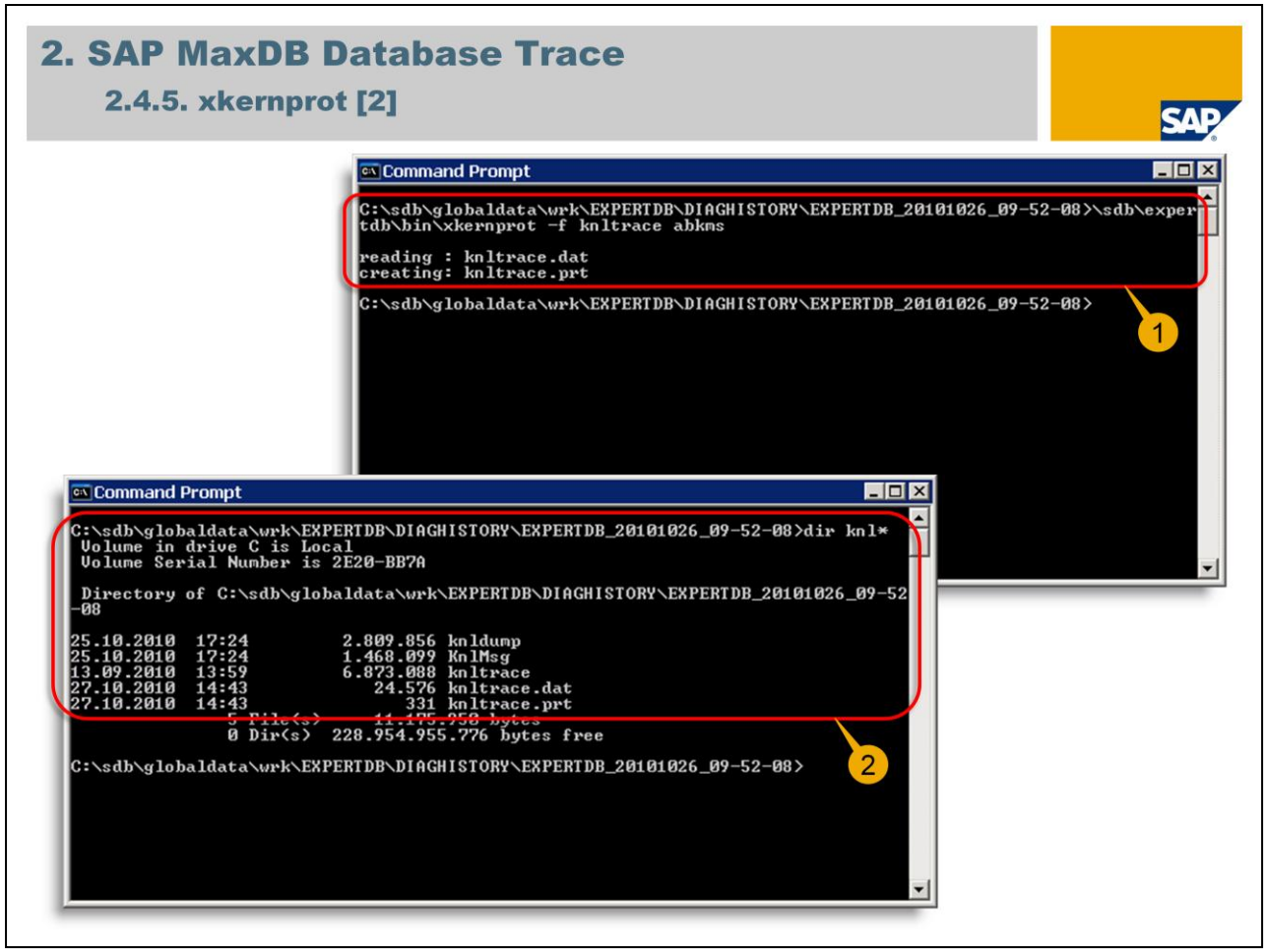
The placeholders are filled with:

<RUNDIRECTORY> : see output of command ‘xinstinfo <SID>‘ or parameter RUNDIRECTORY

<SID> : name of the database

<date> : date of restart

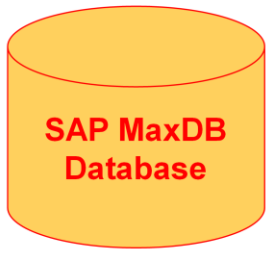
<time> : time of restart



File knltrace is also copied to these subdirectories in its original format. To generate the human-readable trace file here xkernprot can be executed with option ‘-f‘ and specifying the trace file name (including path if required) and the trace options. In this case the generated file isn‘t named ‘<SID>.prt‘ but ‘knltrace.prt‘ and is located in the current directory (1) (2).

2. SAP MaxDB Database Trace

2.5. Using SAP Transactions for database tracing

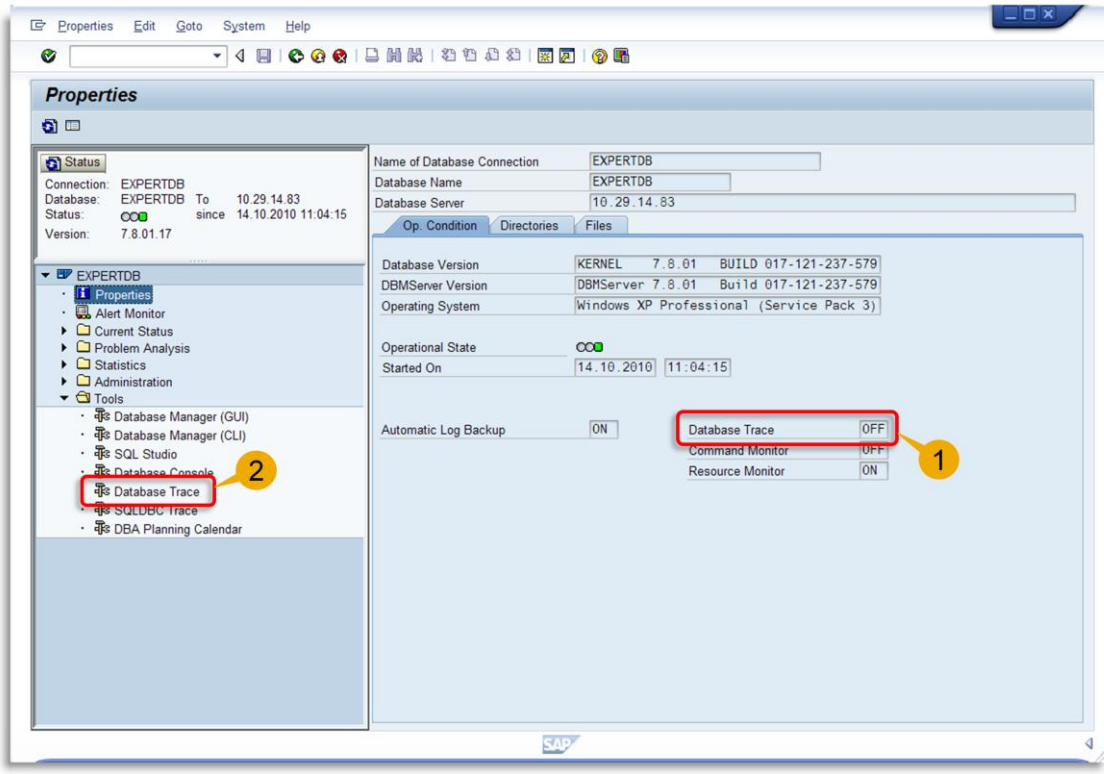


Database Trace
is supported by SAP Transactions:

- DB50
- LC10
- DBACOCKPIT

2. SAP MaxDB Database Trace

2.5.1. DB50 / LC10 [1]

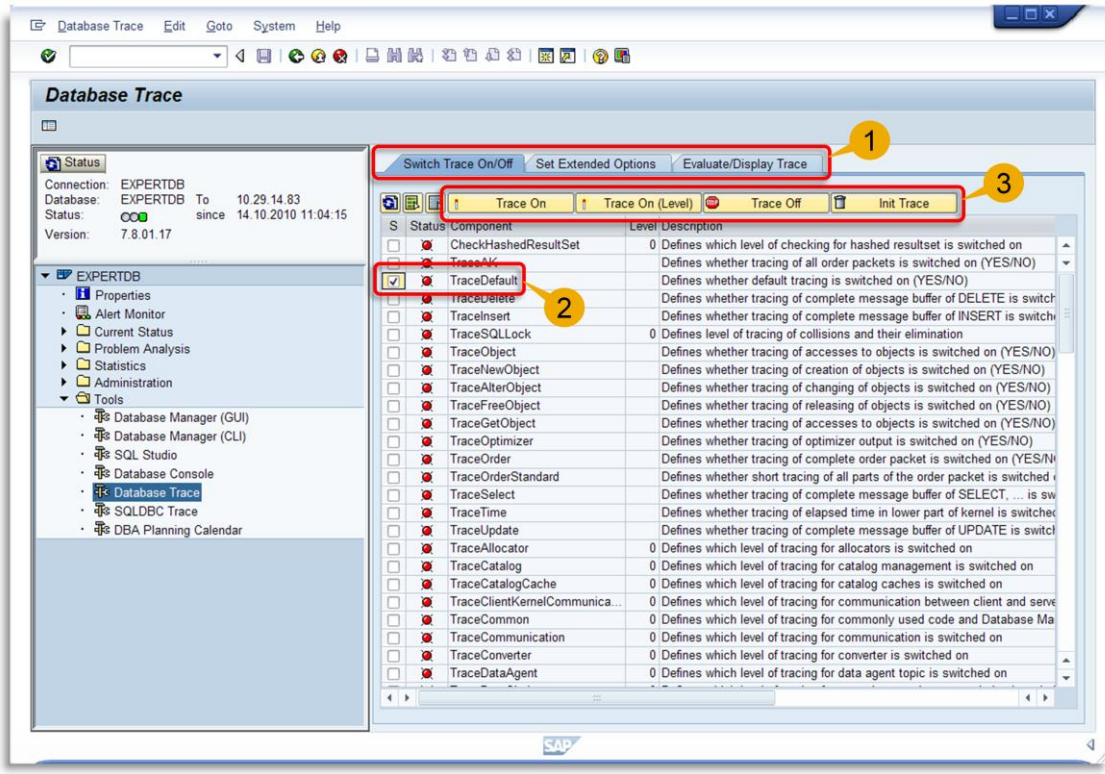


Having called transaction DB50 the overview shown via explorer tree item ‘Properties‘ provides the information whether the Database Trace is enabled or not (1).

The Database Trace functionality can be reached via subtree ‘Tools‘ (2).

2. SAP MaxDB Database Trace

2.5.1. DB50 / LC10 [2]



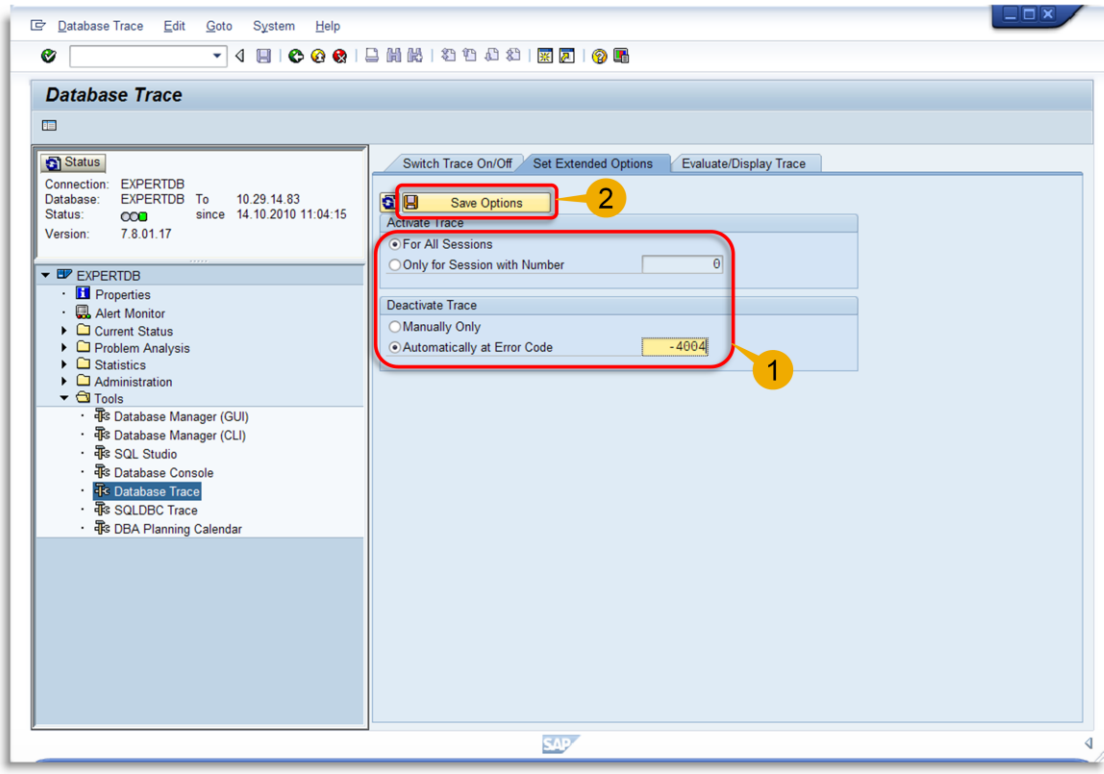
The implementation of the Database Trace functionality is similar to the Database Manager GUI. There are three tabs offering access to the trace activities, the advanced settings (called ‘Extended Options‘ here) and to the area of generating and displaying the human-readable trace file (called ‘Evaluate/Display Trace‘ here) (1).

To set trace activity ‘Default‘ (also called ‘TraceDefault‘ here) it is required to check the corresponding box within the list (2).

The icon bar on top of the list contains the buttons for enabling, enabling with a certain trace level and disabling the trace as well as to empty the trace buffer (Init Trace) (3).

2. SAP MaxDB Database Trace

2.5.1. DB50 / LC10 [3]

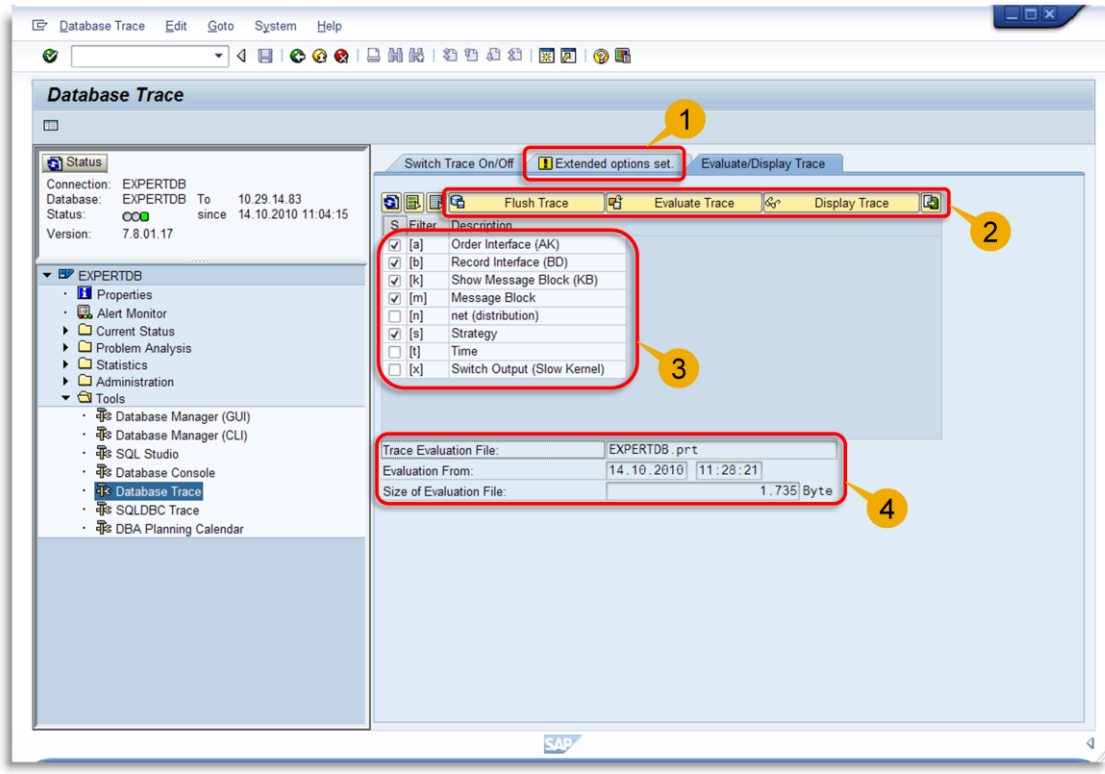


Via tab ‘Set Extended Options‘ it is possible to restrict the trace output regarding a specific session or to request the trace to stop automatically in the event of a certain error (1).

It is important to save these settings to get them active (2).

2. SAP MaxDB Database Trace

2.5.1. DB50 / LC10 [4]



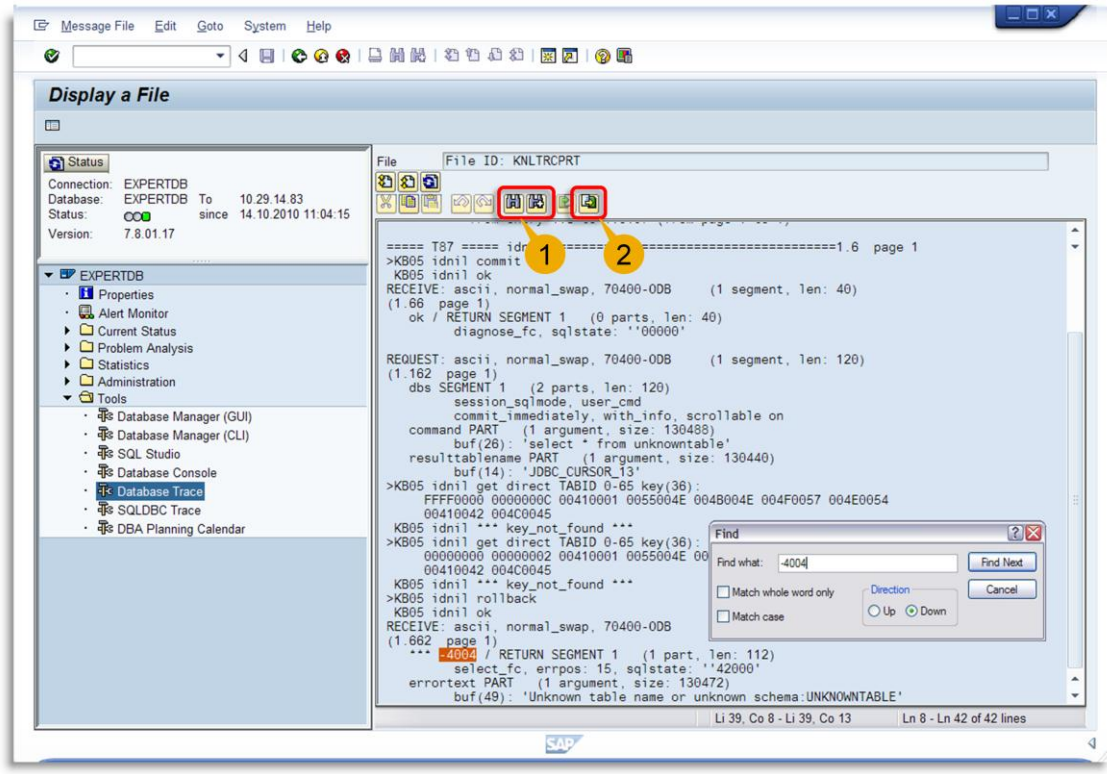
If extended trace options are set an icon containing an exclamation mark is visible and tab label changed to ‘Extended options set‘ to indicate this fact (1).

Tab ‘Evaluate/Display Trace‘ opens the area to flush, evaluate and display the trace. A corresponding icon bar is visible. Button ‘Flush Trace‘ triggers the transfer of the content of the trace buffer to file ‘knltrace‘. Button ‘Evaluate Trace‘ is responsible for generating the human-readable trace file. Before using this button the options for this generating have to be checked in the list below (‘abkms‘). Finally it is possible to display the trace file in place by choosing button ‘Display Trace‘. The small button rightmost in the icon bar allows to save the trace file directly (without displaying before). If the trace file has a size of several megabytes it is better to save the file locally and to open it afterwards in a text viewer. This is faster than opening a big trace file directly within the transaction (2) (3).

Below the trace options list some additional information is listed such as trace file name, evaluation date and size of the trace file (4).

2. SAP MaxDB Database Trace

2.5.1. DB50 / LC10 [5]

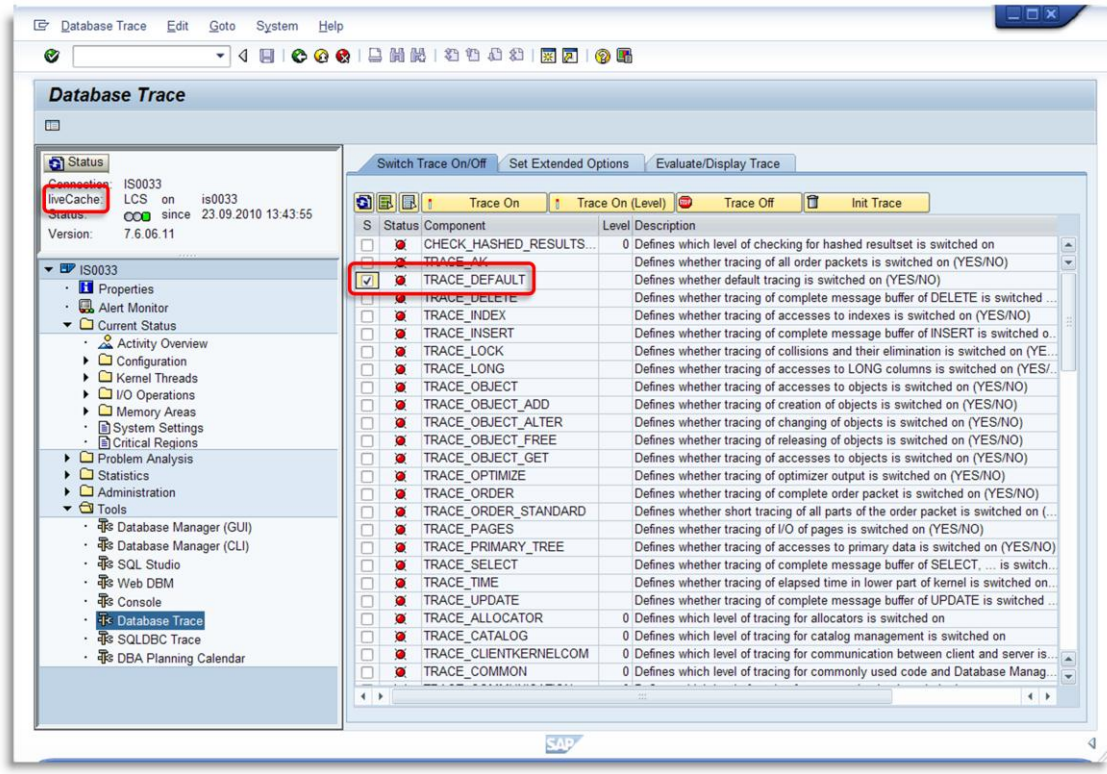


When displaying the trace file in place it possible to search for special strings like e.g. error messages. The icons showing the spyglass trigger this feature. It is also possible to use hotkey ‘Ctrl-f‘ (1).

Saving the trace file can also be done after displaying it. This is possible via button ‘Save as local file‘ at the far right (2).

2. SAP MaxDB Database Trace

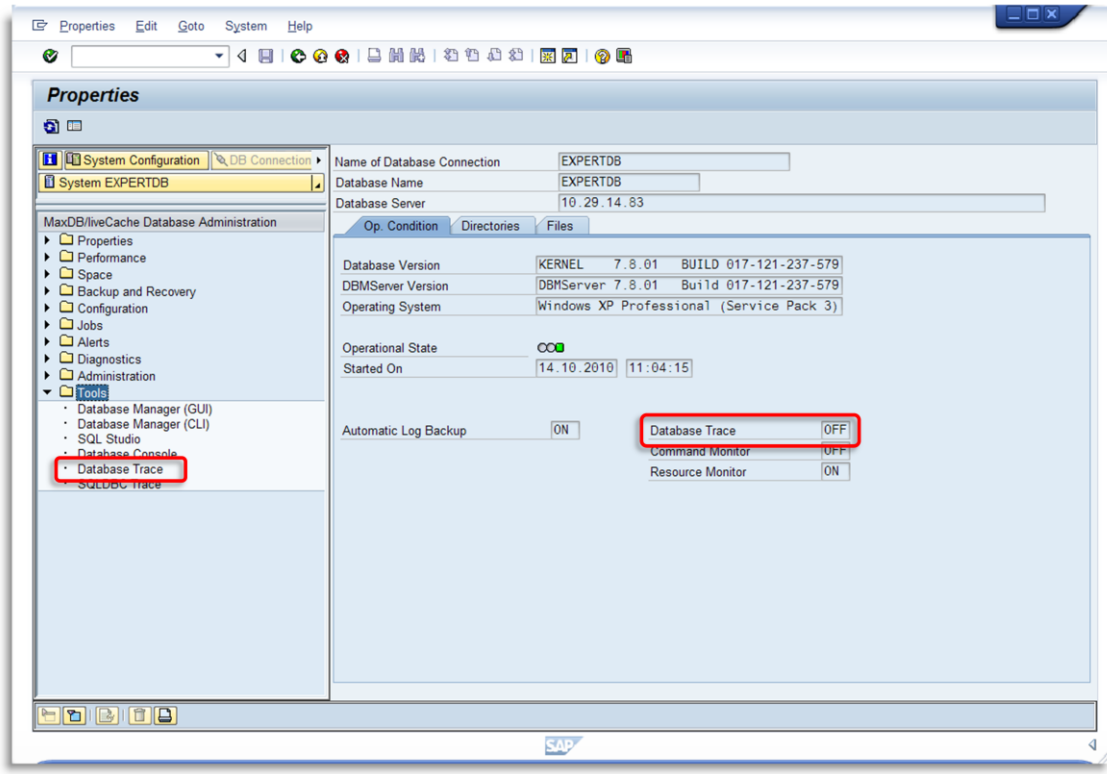
2.5.1. DB50 / LC10 [6]



The implementation of the Database Trace functionality in liveCache administration transaction LC10 is the same compared to DB50.

2. SAP MaxDB Database Trace

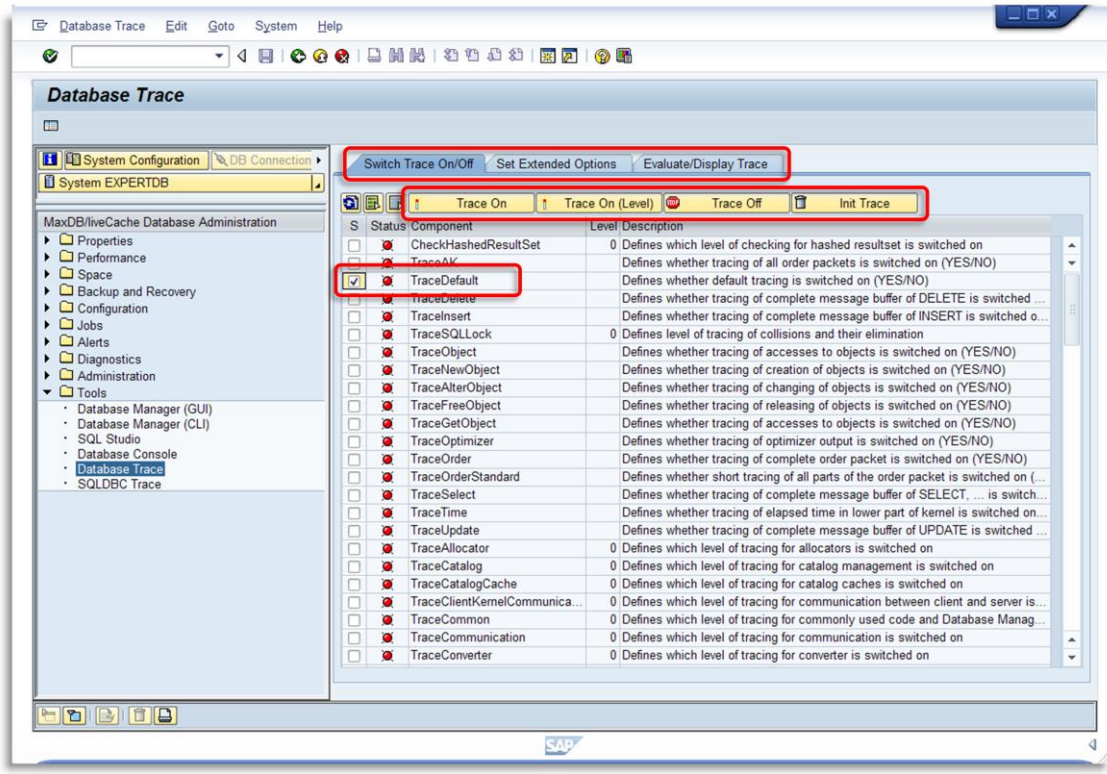
2.5.2. DBACOCKPIT [1]



Even in transaction DBACOCKPIT, which is the successor of DB50 / LC10, the implementation of the Database Trace functionality is all the same.

2. SAP MaxDB Database Trace

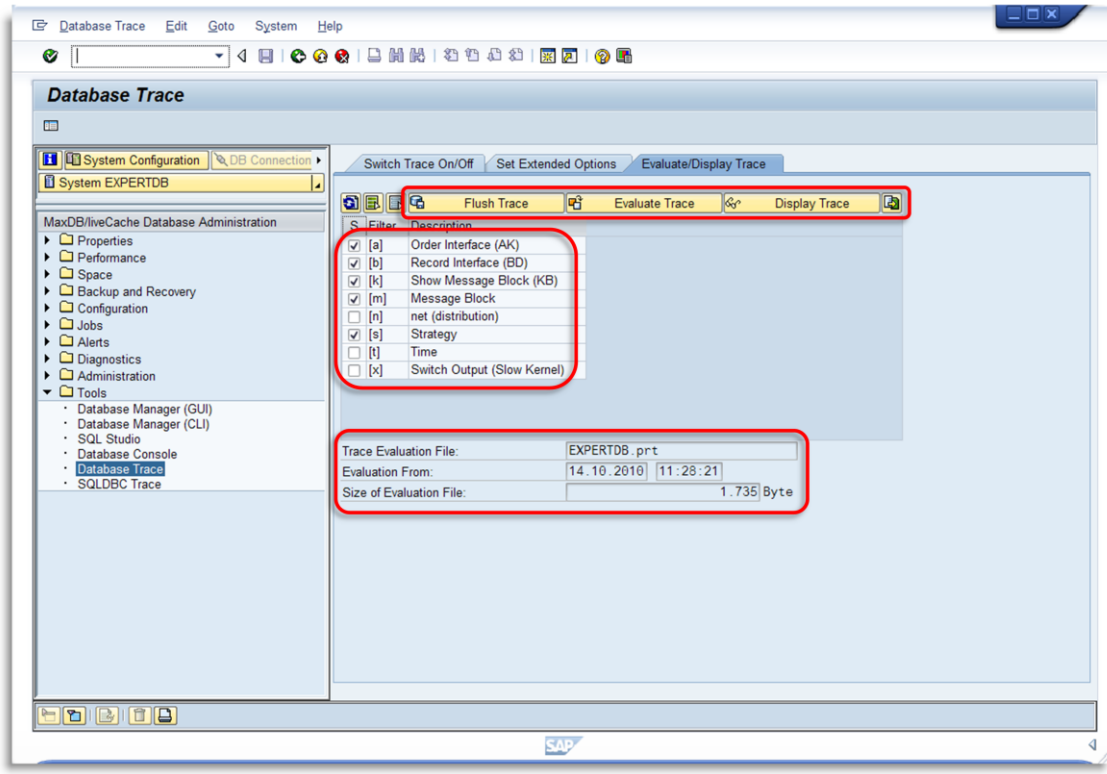
2.5.2. DBACOCKPIT [2]



Even in transaction DBACOCKPIT, which is the successor of DB50 / LC10, the implementation of the Database Trace functionality is all the same.

2. SAP MaxDB Database Trace

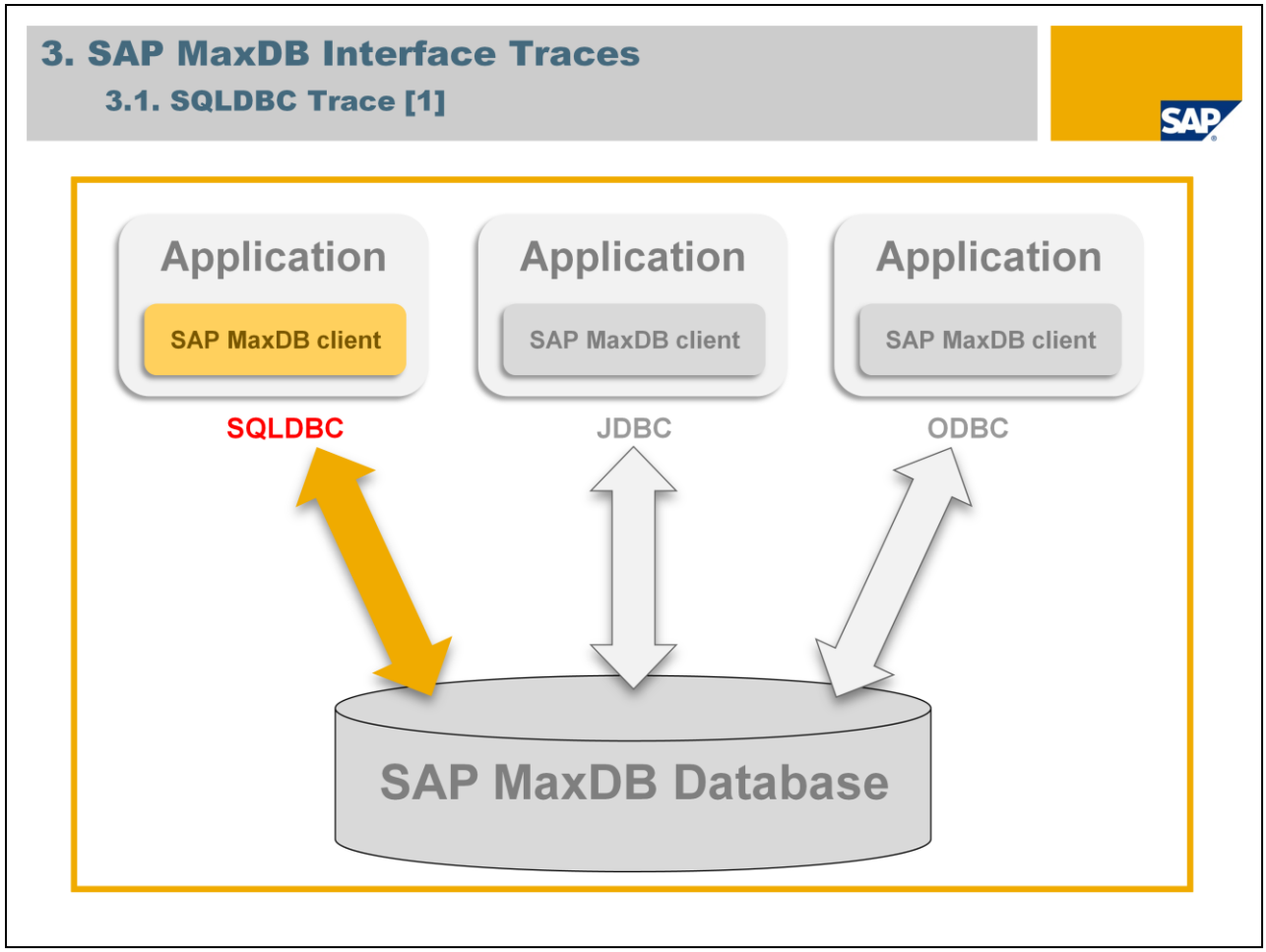
2.5.2. DBACOCKPIT [3]



Even in transaction DBACOCKPIT, which is the successor of DB50 / LC10, the implementation of the Database Trace functionality is all the same.

Agenda

1. Introduction
2. SAP MaxDB Database Trace
3. SAP MaxDB Interface Traces
4. Additional Traces
5. Useful Information Resources



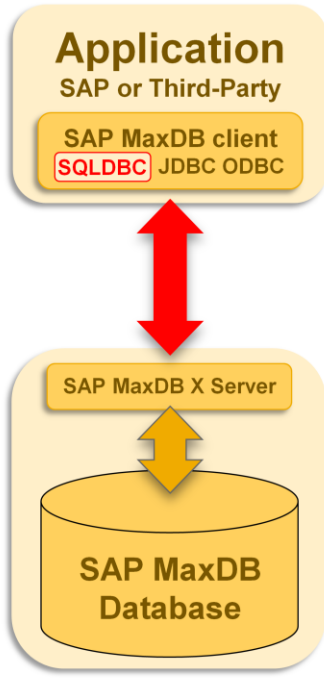
SQL Database Connectivity (SQLDBC) is a runtime library for the development of applications for MaxDB. Using SQLDBC applications can connect to MaxDB instances, execute SQL statements and process data. It consists of the runtime library libSQLDBC, the software development kit SQLDBC SDK and the tool sqldbc_cons for tracing.

The SQLDBC trace contains SQL statements sent by the application to the database instance, their parameters and results.

SAP kernel as of version 6.40 EXT-2 uses SQLDBC.

3. SAP MaxDB Interface Traces

3.1. SQLDBC Trace [2]



When it is indicated to use this trace?

- An error occurs in the interface between application and database
- Examples: SQL error, Connect error, Short dump

Which tools can be used to activate?

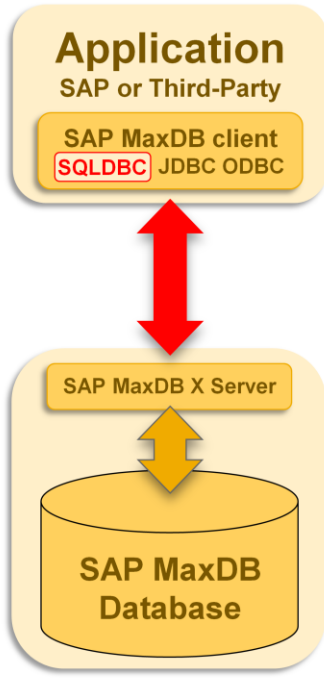
SAP MaxDB Tools	SAP Transactions
SQLDBC_CONS	DB50 / LC10
	DBACOCKPIT

Where is the trace log file located?

- in the directory <indep_data>\wrk (Microsoft Windows) respectively <user_home>/.sdb/<computer name> (UNIX/LINUX) as sqldbctrace-<pid>.prt file
- Example: C:\sdb\globaldata\wrk\sqldbctrace-125.prt
- Name, location and size of the trace log file can be changed

3. SAP MaxDB Interface Traces

3.1. SQLDBC Trace [3]

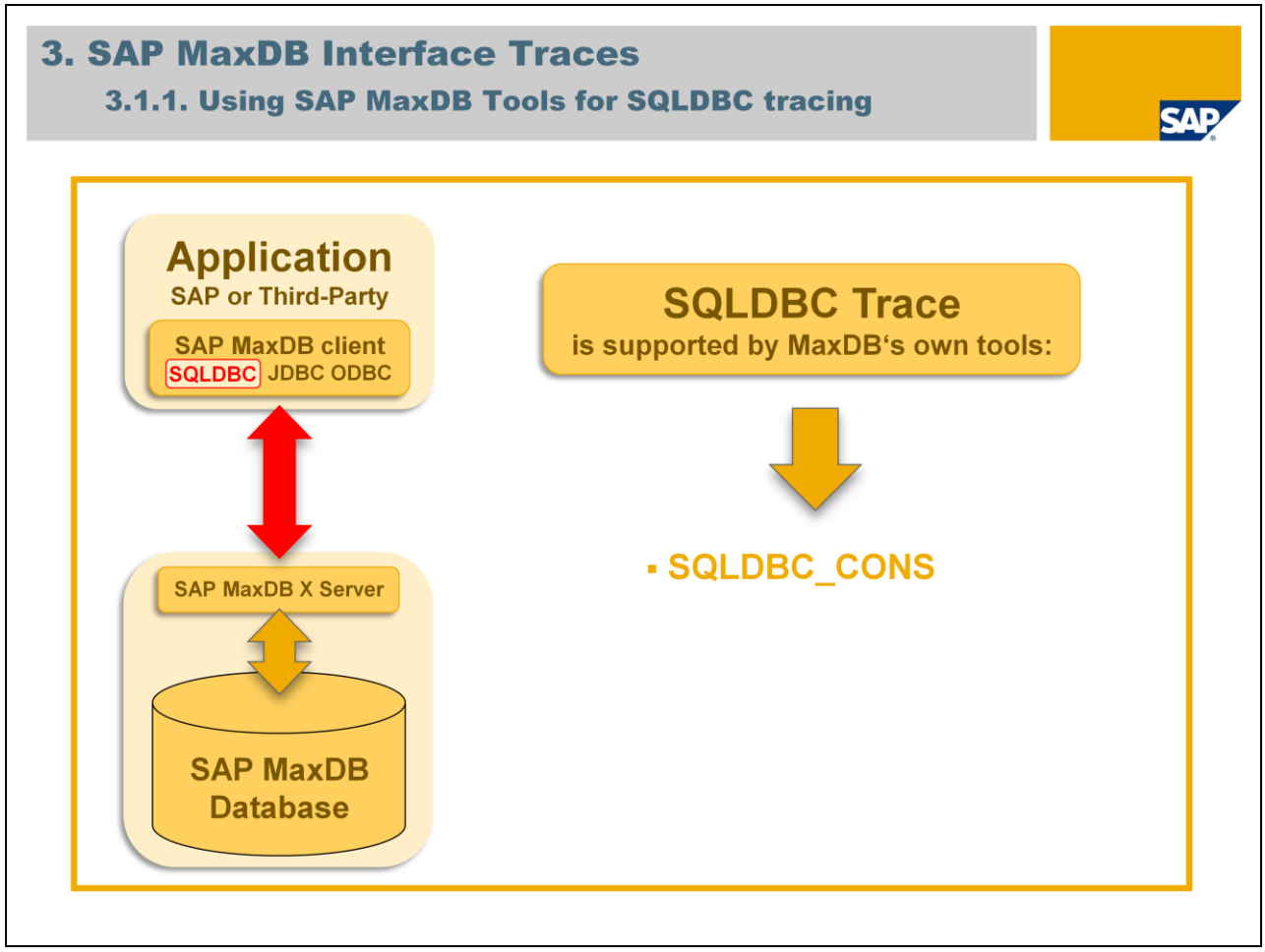


For which user the trace is written?

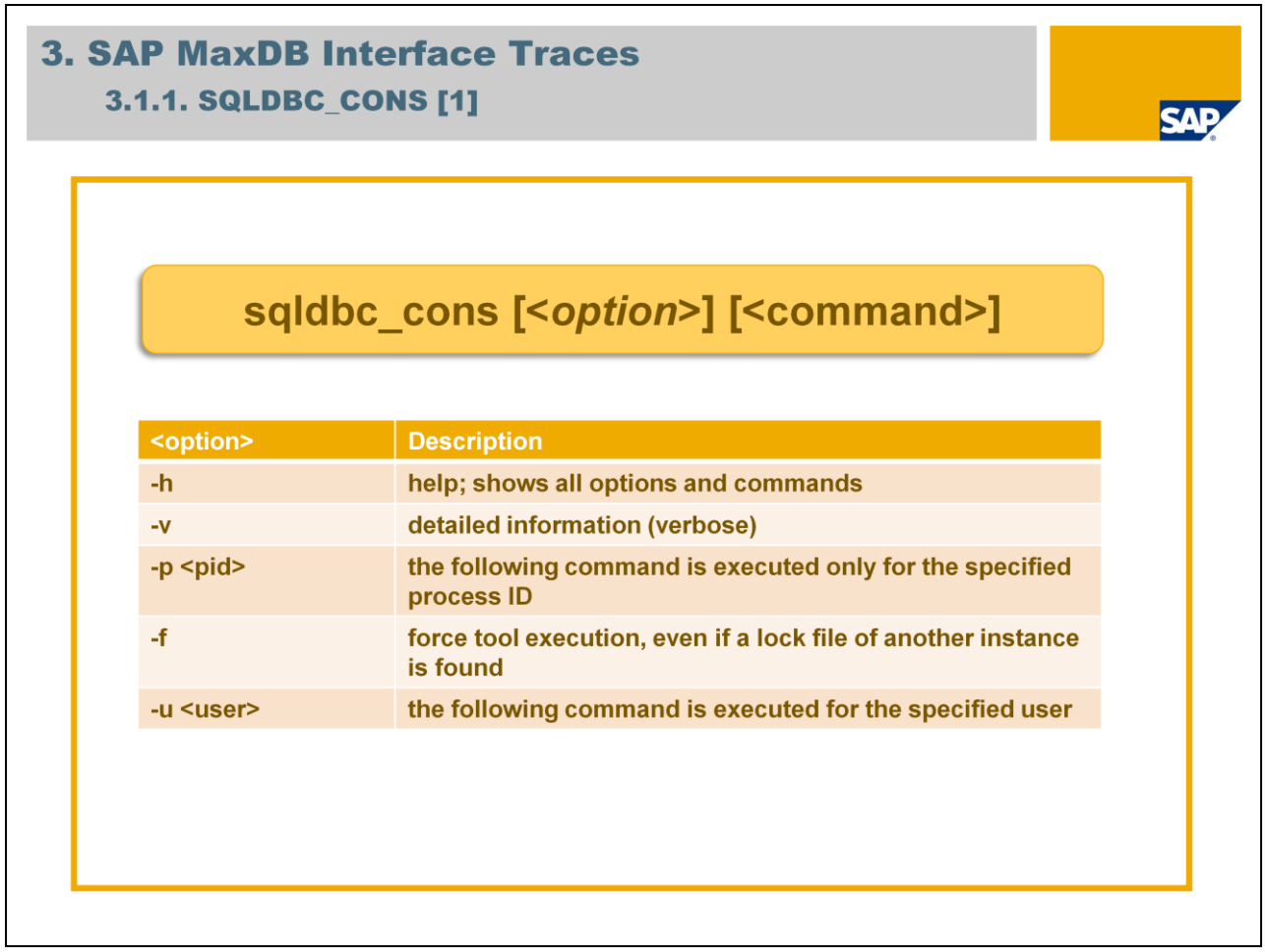
- The SQLDBC trace is written for the OS user who switches the trace on.
- On Windows the R/3 processes usually run as SAPService<SID>. Therefore you must specify the -u option of sqldbc_cons when you activate the trace at operating system level:

  sqldbc_cons -u SAPService<SID> <trace command>

- This is not required in DB50/LC10/DBACOCKPIT since the environment of the SAPService<SID> already exists there.



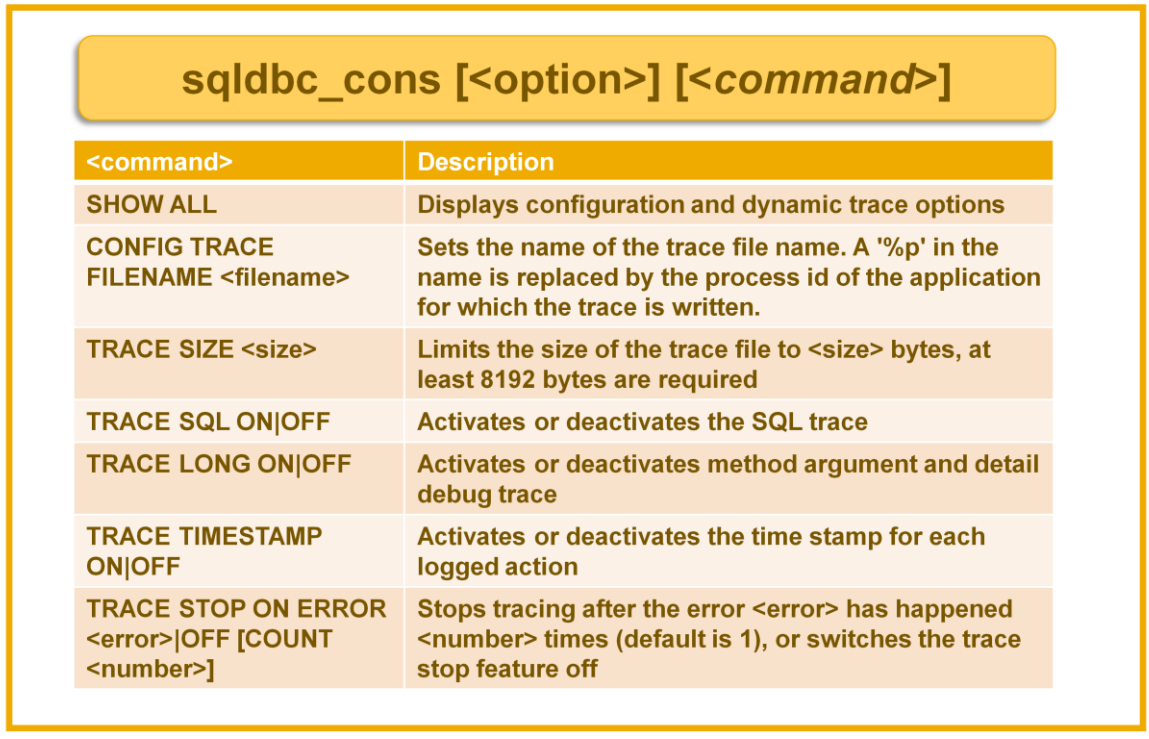
To configure the SQLDBC trace settings - e.g. to start or stop the trace - you can use sqldbc_cons.



Note that you can enter only one trace option at a time. They cannot be combined.

3. SAP MaxDB Interface Traces

3.1.1. SQLDBC_CONS [2]



sqldbc_cons [<option>] [<*command*>]

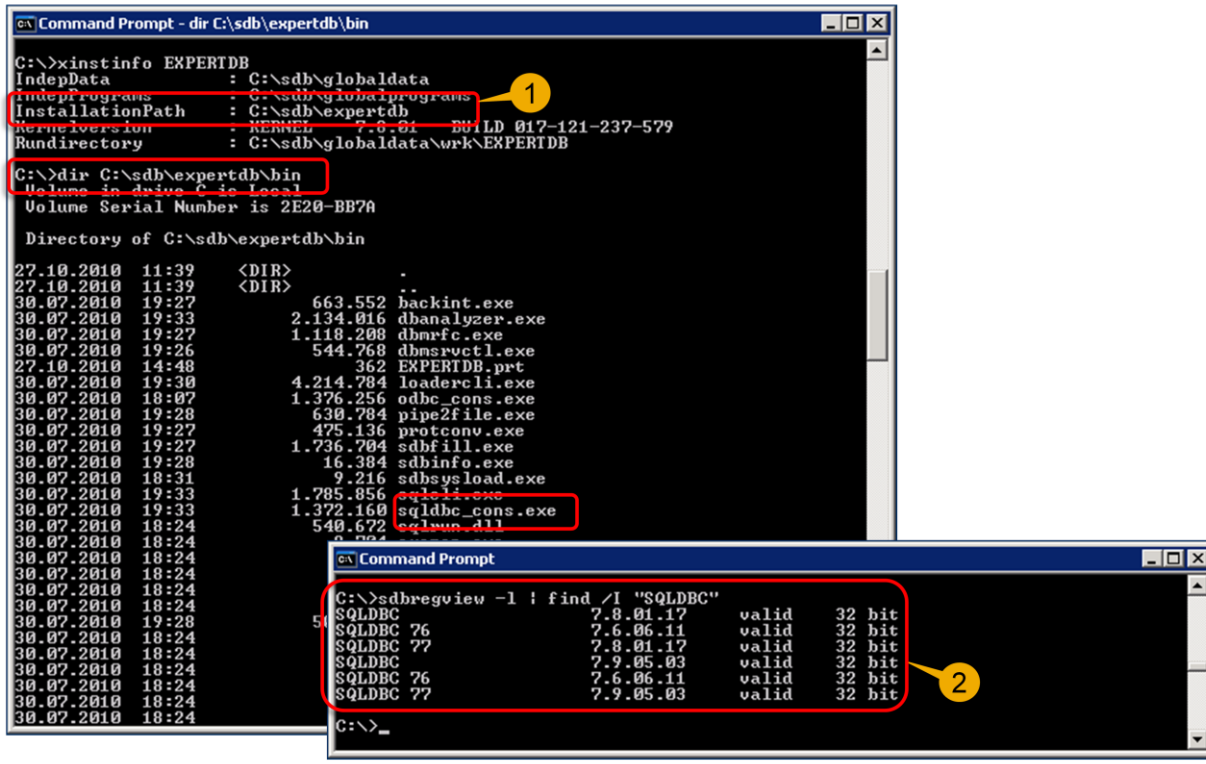
<command>	Description
SHOW ALL	Displays configuration and dynamic trace options
CONFIG TRACE FILENAME <filename>	Sets the name of the trace file name. A '%p' in the name is replaced by the process id of the application for which the trace is written.
TRACE SIZE <size>	Limits the size of the trace file to <size> bytes, at least 8192 bytes are required
TRACE SQL ON|OFF	Activates or deactivates the SQL trace
TRACE LONG ON|OFF	Activates or deactivates method argument and detail debug trace
TRACE TIMESTAMP ON|OFF	Activates or deactivates the time stamp for each logged action
TRACE STOP ON ERROR <error>|OFF [COUNT <number>]	Stops tracing after the error <error> has happened <number> times (default is 1), or switches the trace stop feature off

Only the most important commands are mentioned. To see all available commands option ‘-h’ can be used or have a look at MaxDB documentation.

Again only one command can be executed at the same time.

3. SAP MaxDB Interface Traces

3.1.1. SQLDBC_CONS [3]



Tool ‘sqldbc_cons’ is part of the MaxDB software installation and is located in <InstallationPath>/bin. To figure out the installation path command ‘xinstinfo <SID>‘ can be used (1).

You can use the program 'sdbregview' to determine which SQLDBC versions are installed on the relevant host.

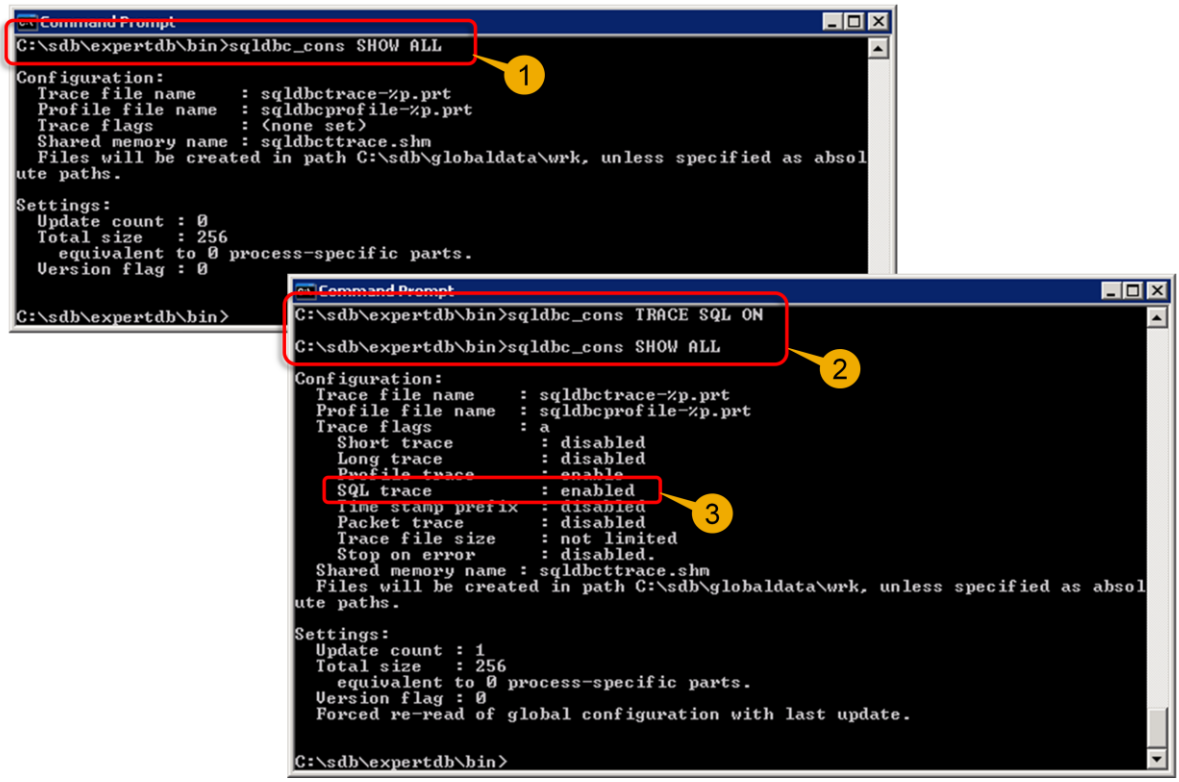
a) On UNIX platforms: 'sdbregview –l | grep -i SQLDBC'

b) On Windows platforms: 'sdbregview -l | find /I "SQLDBC"'

Several SQLDBC versions can be installed (2).

3. SAP MaxDB Interface Traces

3.1.1. SQLDBC_CONS [4]



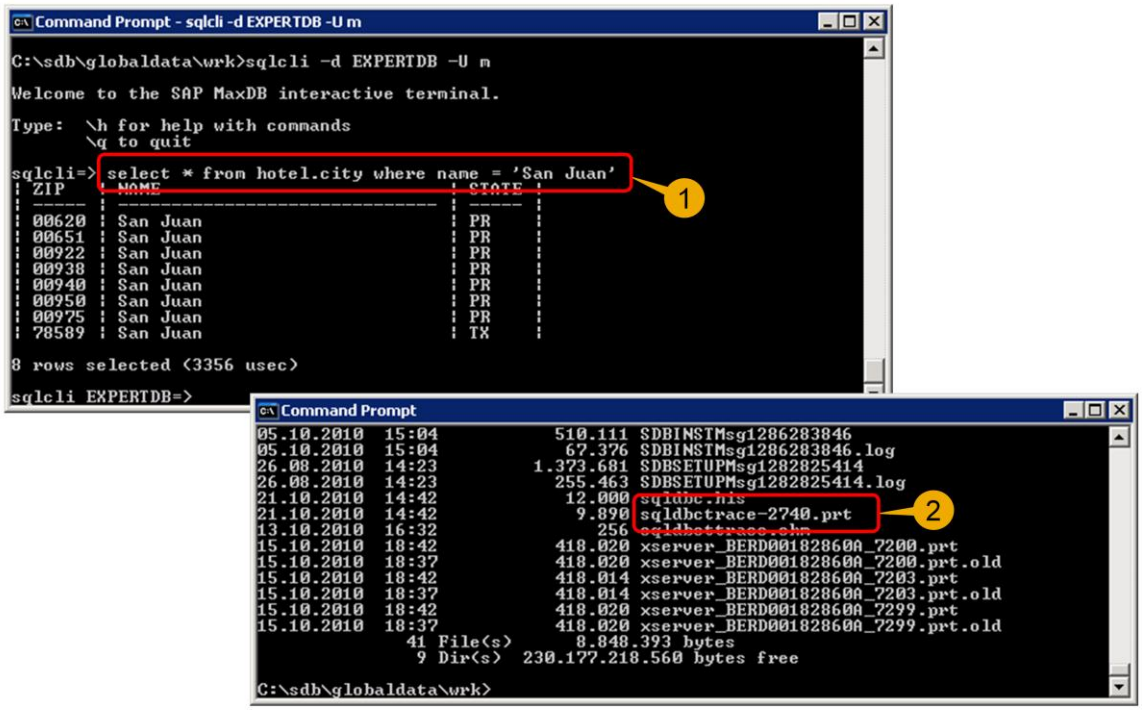
SQLDBC trace configuration is listed with command ‘sqldbc_cons SHOW ALL’ (1).

Trace file name is set to ‘sqldbctrace-%p.prt’. The ‘%p’ will be replaced by the process id. Trace file will be created in ‘C:\sdb\globaldata\wrk’.

To switch on SQL trace command ‘sqldbc_cons TRACE SQL ON’ is used. With command ‘sqldbc_cons SHOW ALL’ the new trace configuration is displayed (2) (3).

3. SAP MaxDB Interface Traces

3.1.1. SQLDBC_CONS [5]



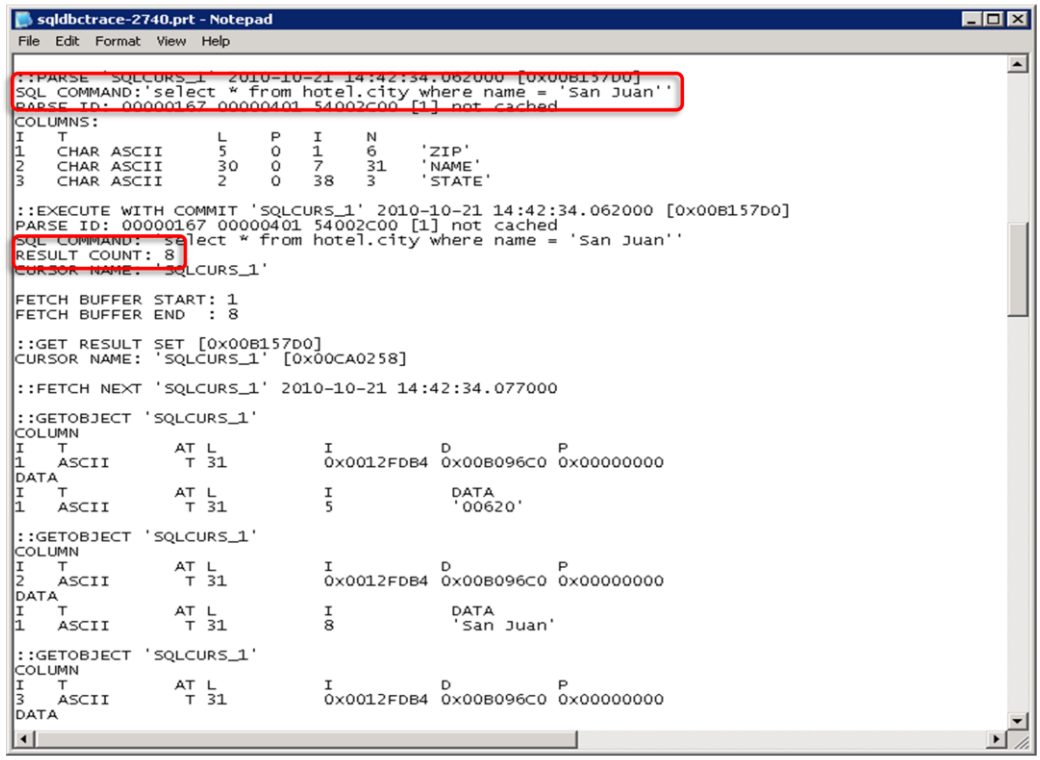
One application which is using SQLDBC to connect to MaxDB instances and execute SQL statements is the command line tool SQLCLI. It is a component of the MaxDB software.

To connect to database EXPERTDB command ‘sqlcli –d EXPERTDB –U m’ is used. SQL command ‘select * from hotel.city where name = ‘San Juan’ is executed (1).

In directoy ‘C:\sdb\globaldata\wrk’ SQLDBC trace file ‘sqldbctrace-2740.prt’ is created (2).

3. SAP MaxDB Interface Traces

3.1.1. SQLDBC_CONS [6]



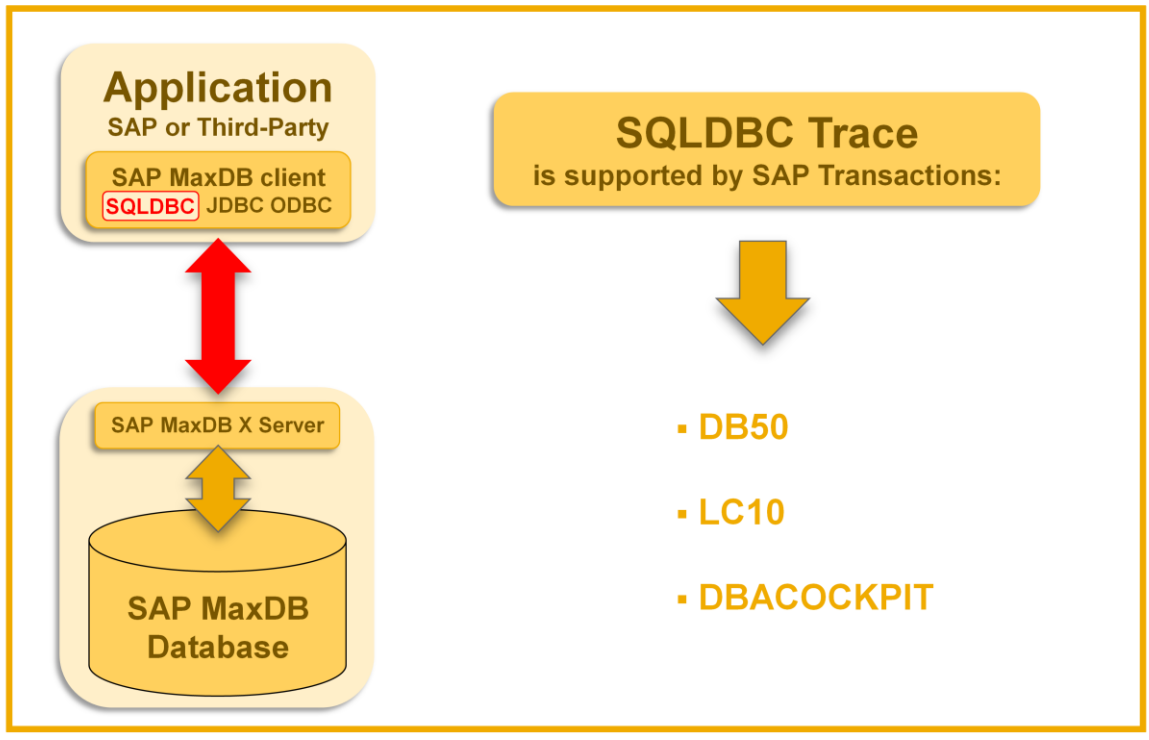
SQLDBC trace file ‘sqldbctrace-2740.prt’ contains the executed SQL statements, their parameters and results.

Reading the trace in detail is the job of MaxDB expert.

Don’t forget to switch off SQL trace with command ‘sqldbc_cons TRACE SQL OFF’.

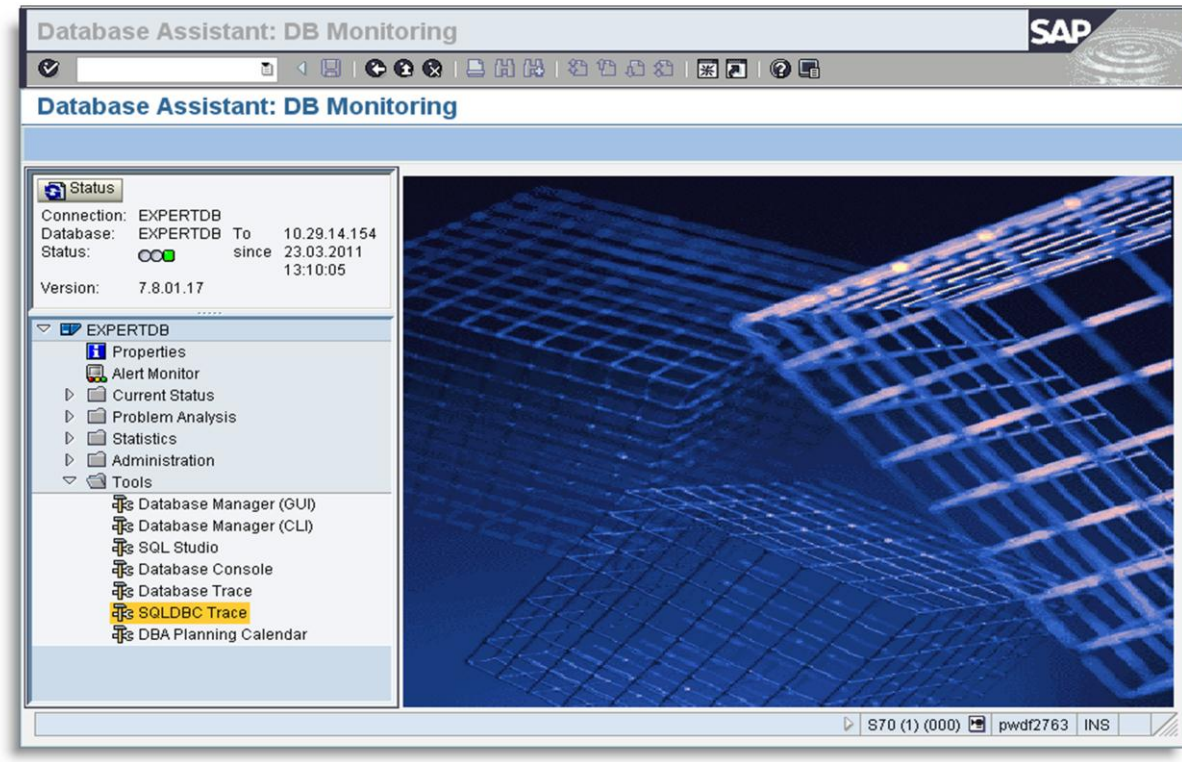
3. SAP MaxDB Interface Traces

3.1.2. Using SAP Transactions for SQLDBC tracing



3. SAP MaxDB Interface Traces

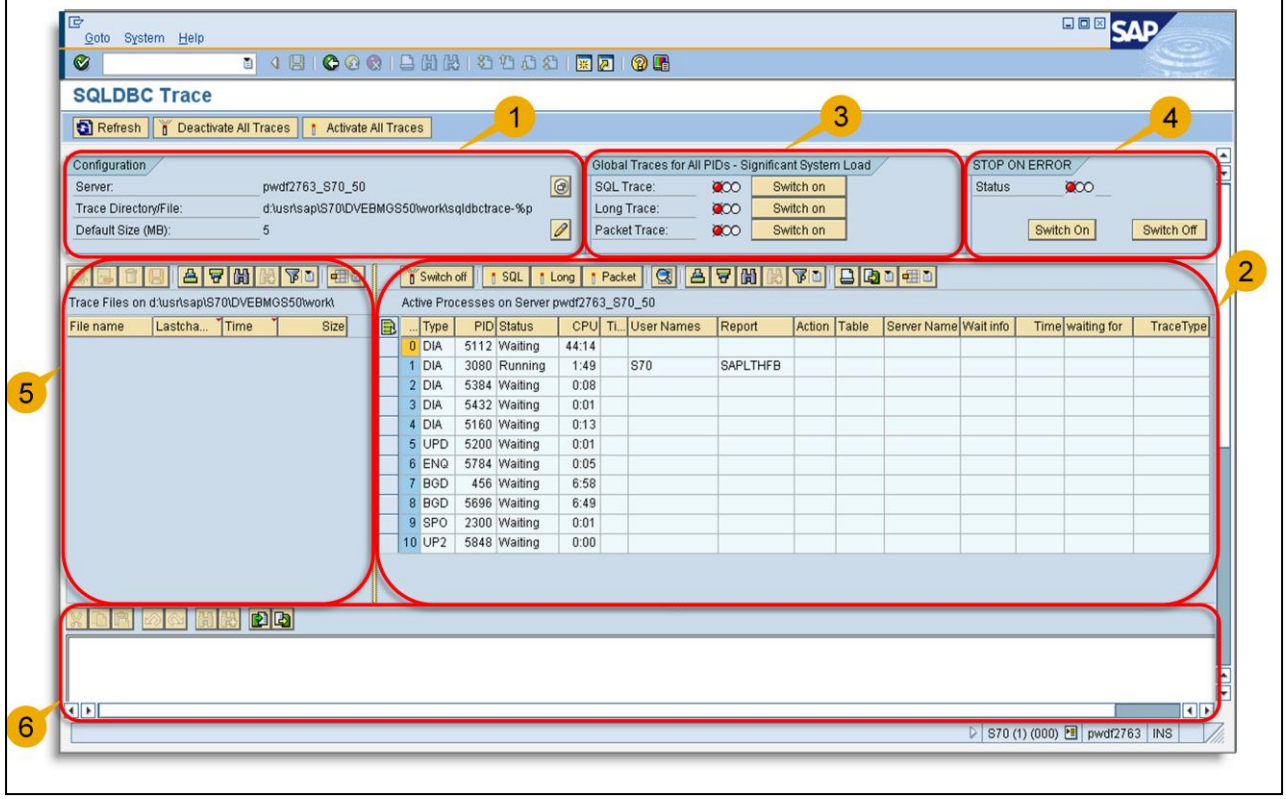
3.1.2. DB50 / LC10 [1]



To activate the SQLDBC trace in transaction DB50 or LC10 you need SAP Basis release 7.00 or higher. The SQLDBC Trace functionality can be reached via subtree ‘Tools‘.

3. SAP MaxDB Interface Traces

3.1.2. DB50 / LC10 [2]



On ‚Configuration' section application server name, trace file name, location and default size are listed. It is possible to switch the application server and to change the trace file size. For security reason trace file location and name can not be changed (1).

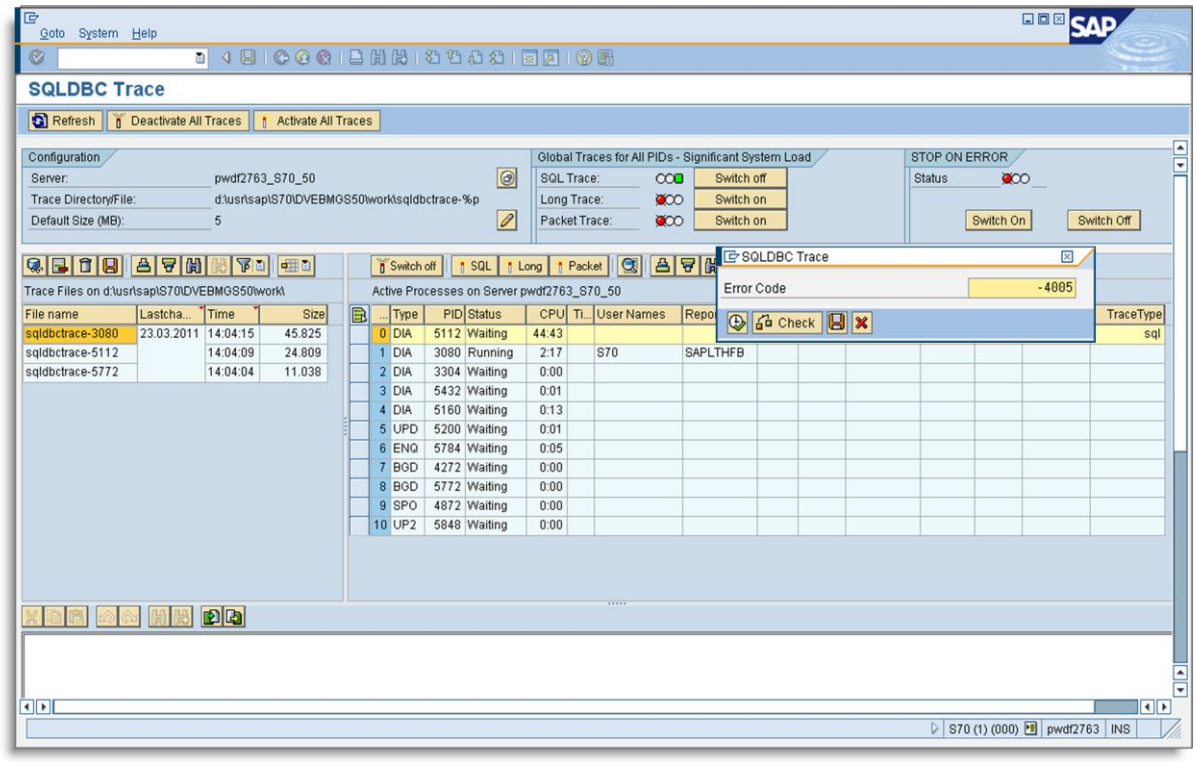
Below on the right the section shows an overview of all work processes as it is known from transaction SM50. Additionally the column ‚Trace Type' was added. For an selected workprocess three types of SQLDBC trace can be activated via the buttons at the top: SQL, Long or Packet trace (2). To switch on these types of traces for all work processes buttons on section ‚Global Traces' can be used (3). It can be lead to a significant system load.

On section ‚STOP ON ERROR' exists the possibility to request the trace to stop automatically in the event of a certain error (4).

Section (5) shows all existing SQLDBC trace files in the displayed trace directory. Via the buttons at the top one trace file can be shown in the section (6) at the bottom.

3. SAP MaxDB Interface Traces

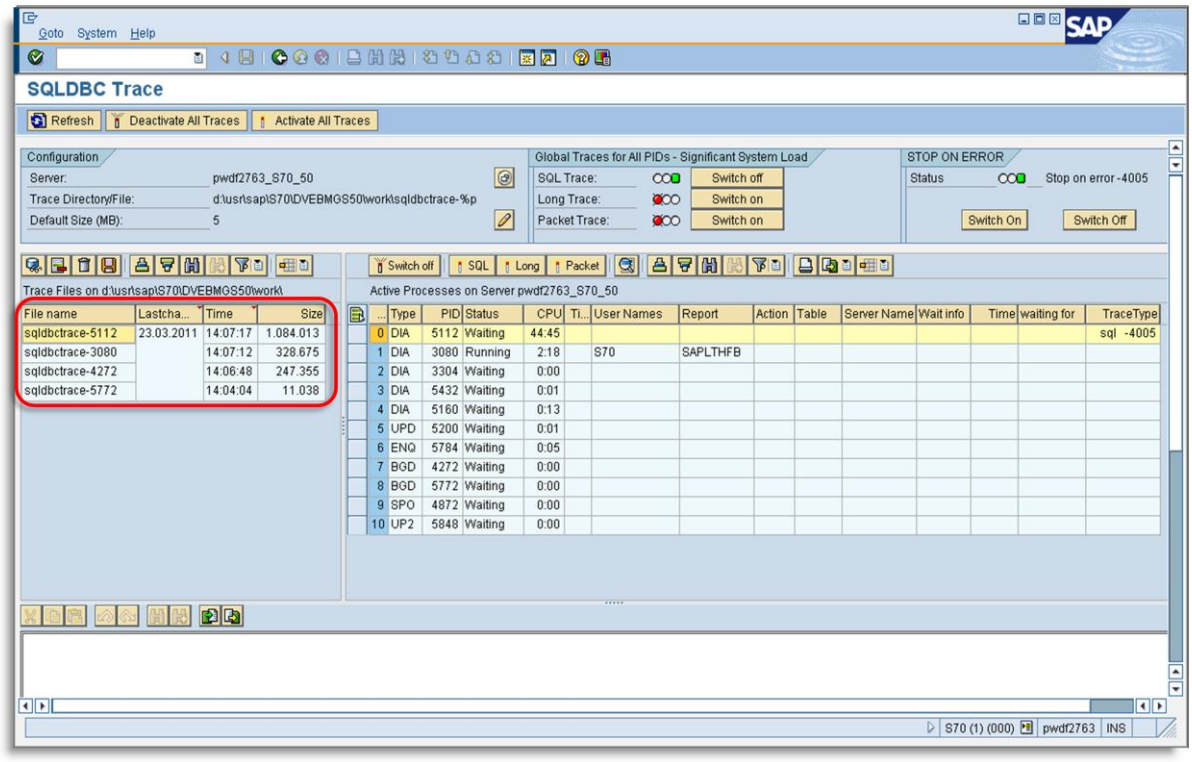
3.1.2. DB50 / LC10 [3]



SQLDBC trace type SQL is already activated for all workprocesses. Tracing should be stopped automatically in the event of database error ‚-4005 Unknown column name ‘ for instance.

3. SAP MaxDB Interface Traces

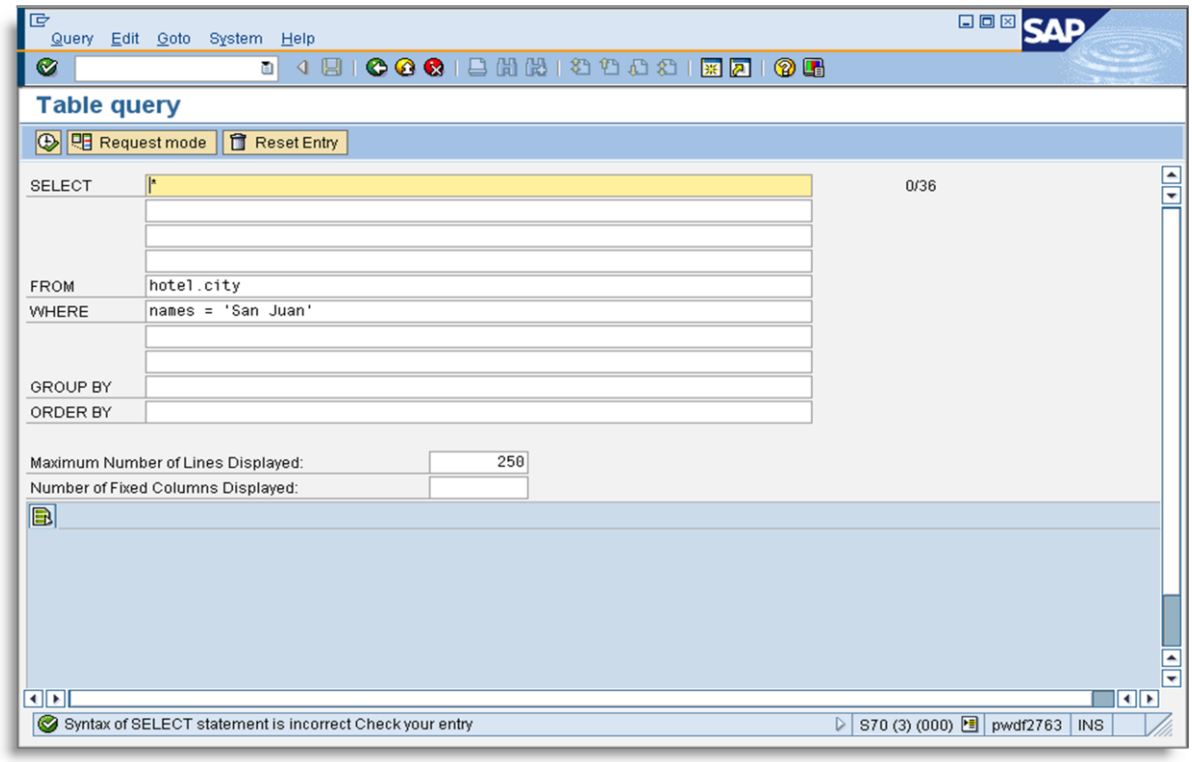
3.1.2. DB50 / LC10 [4]



For each workprocces which has sent SQL statements to the database in the meantime a SQLDBC trace file was already created.

3. SAP MaxDB Interface Traces

3.1.2. DB50 / LC10 [5]



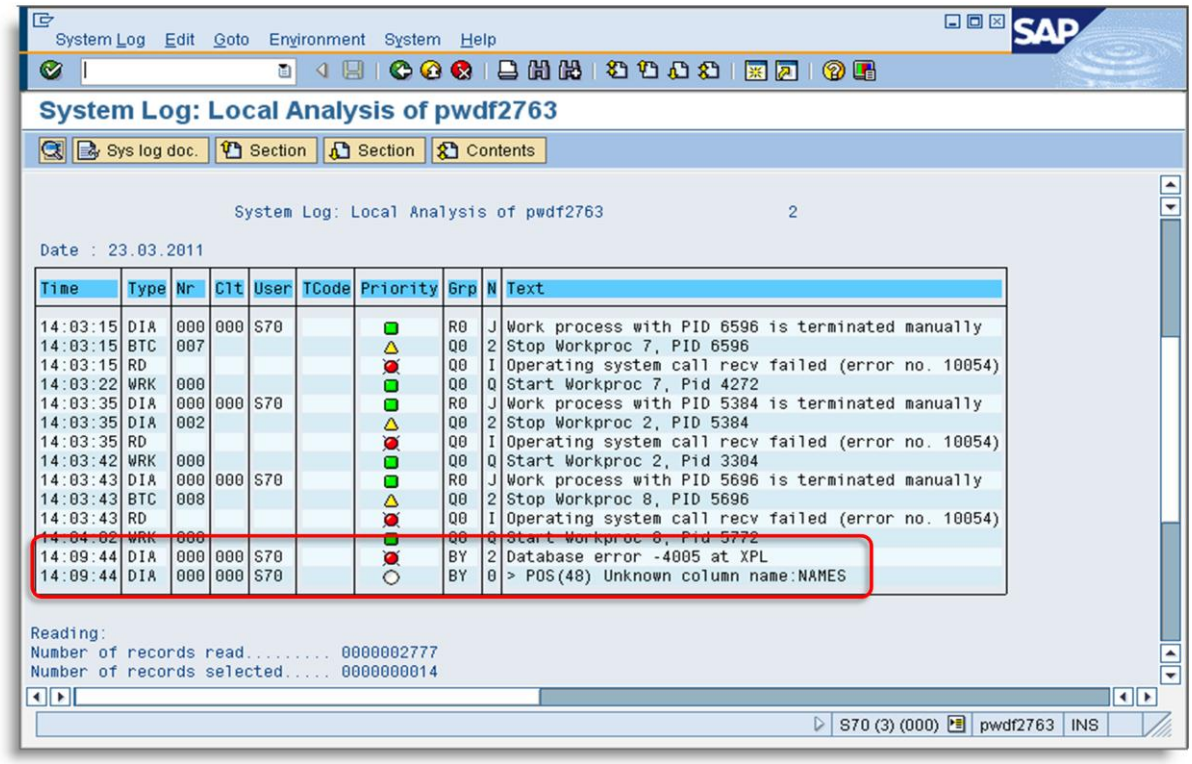
With the following SQL statement the error ‚-4005 Unknown column name' could be generated (column ‚names' instead of ‚name'):

SELECT * FROM hotel.city WHERE names = ‚San Juan'

This SQL statement was executed with menu ‚Utilities' -> ‚Free Table Query' of transaction DB50 (mode SAPEXP).

3. SAP MaxDB Interface Traces

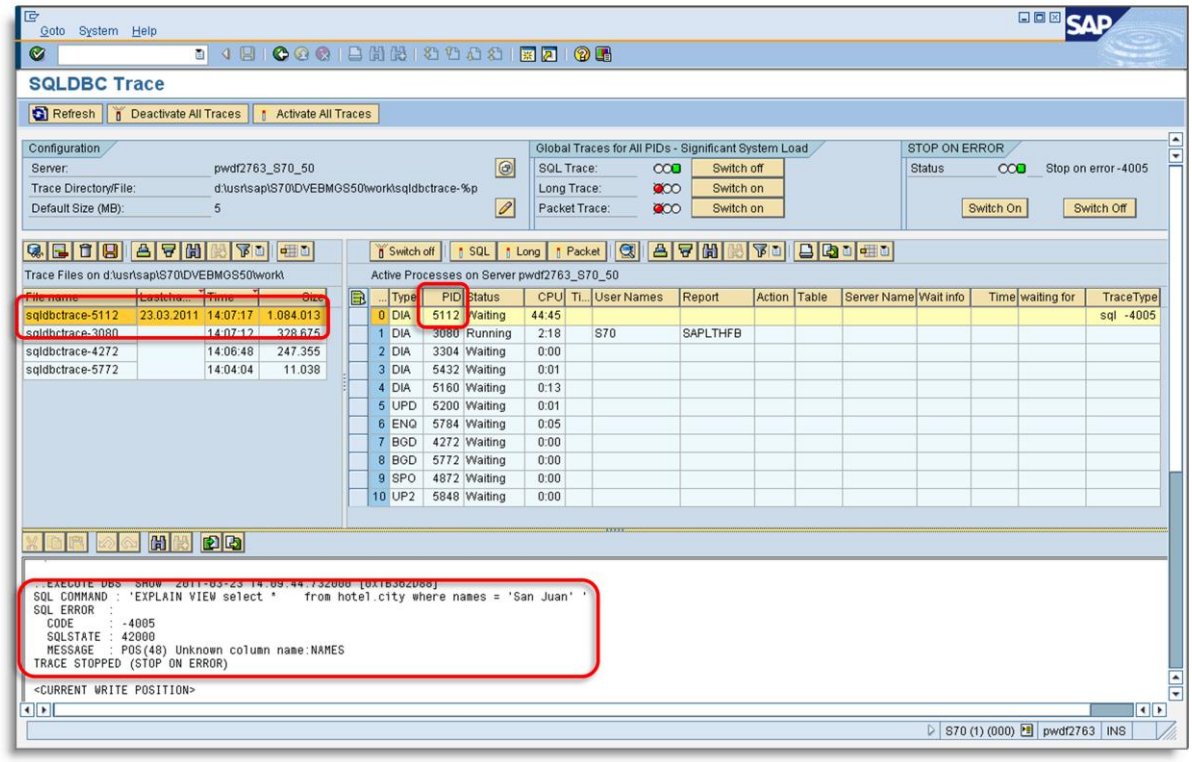
3.1.2. DB50 / LC10 [6]



System log overview (transaction SM21): Database error ‚-4005' occured in work process number 0.

3. SAP MaxDB Interface Traces

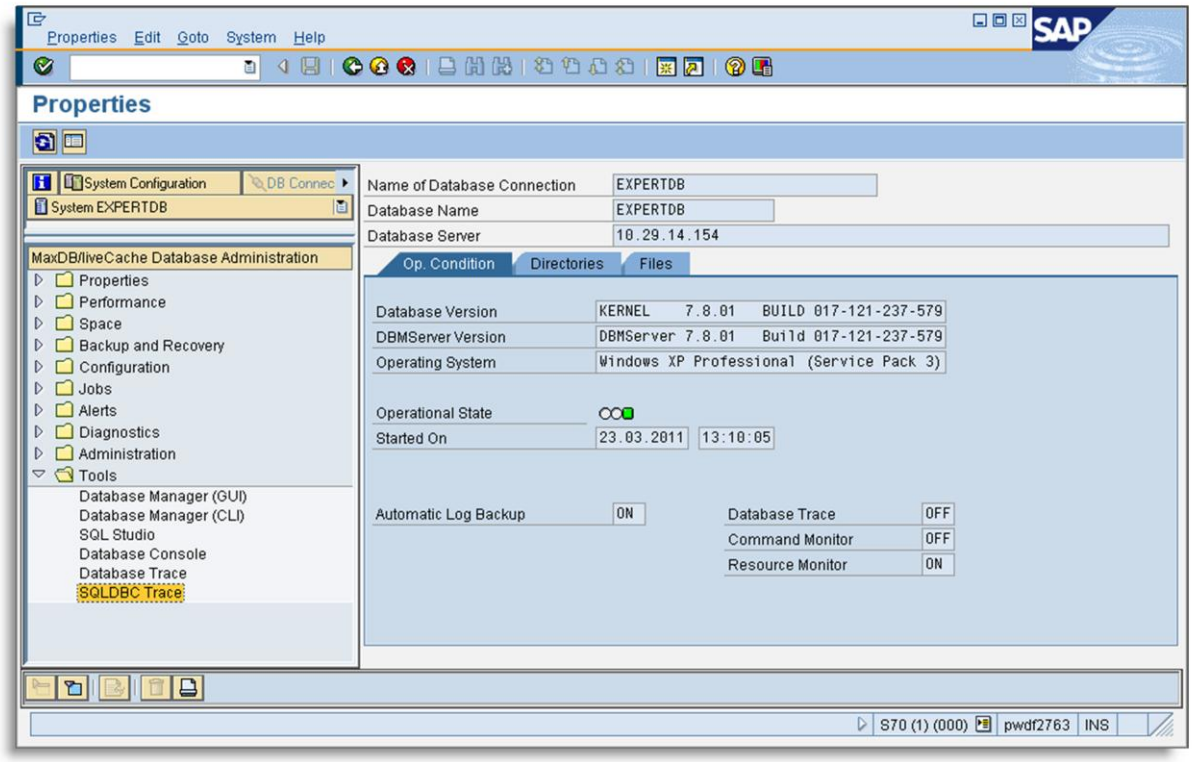
3.1.2. DB50 / LC10 [7]



Work process number 0 has process id 5112. In corresponding SQLDBC trace file sqldbctrace-5112 SQL statement together with error code is logged. Tracing was stopped immediately after the error occured but only for work process number 0.

3. SAP MaxDB Interface Traces

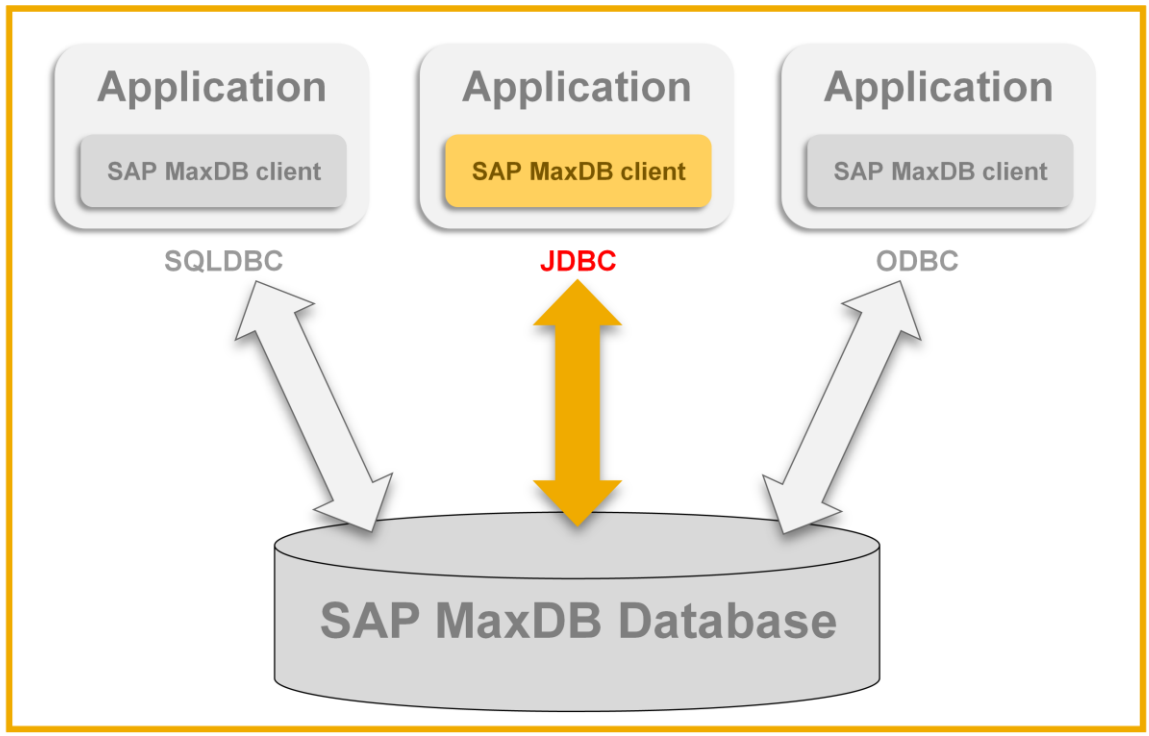
3.1.2. DBACOCKPIT [1]



Even in transaction DBACOCKPIT, which is the successor of DB50 / LC10, the implementation of the SQLDBC Trace functionality is all the same.

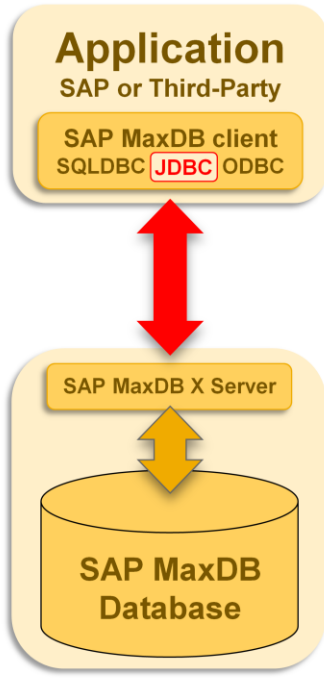
3. SAP MaxDB Interface Traces

3.2. JDBC Trace [1]



3. SAP MaxDB Interface Traces

3.2. JDBC Trace [2]



When it is indicated to use this trace?

- An error occurs in the Java interface between application and database
- Examples: Connect error, SQL error,

Which tools can be used to activate?

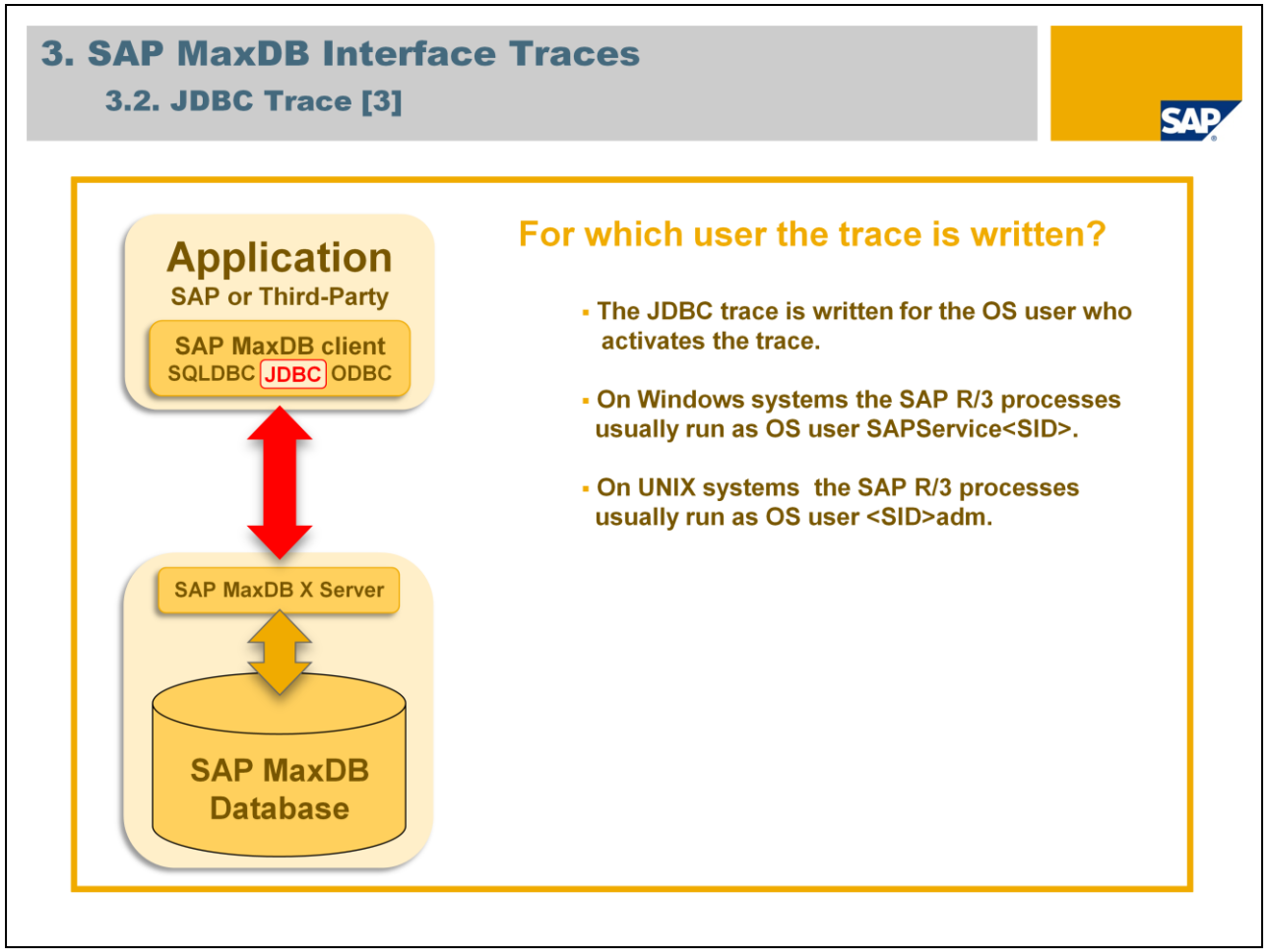
SAP MaxDB Tools	SAP Transactions
SAP MaxDB JDBC driver	

Where is the trace log file located?

- No fix location, can be configured
- Name of the trace log files can also be changed
- Example: C:\ jdbctrace3945742732322446323.prt

The JDBC trace logs JDBC API calls from the JDBC application including call parameters.

Furthermore executed SQL statements and their results are logged.



The JDBC trace is always user dependend.

User SAPService<SID> on Windows systems is a service user. This means it is not an interactive user by default (no logon possible).

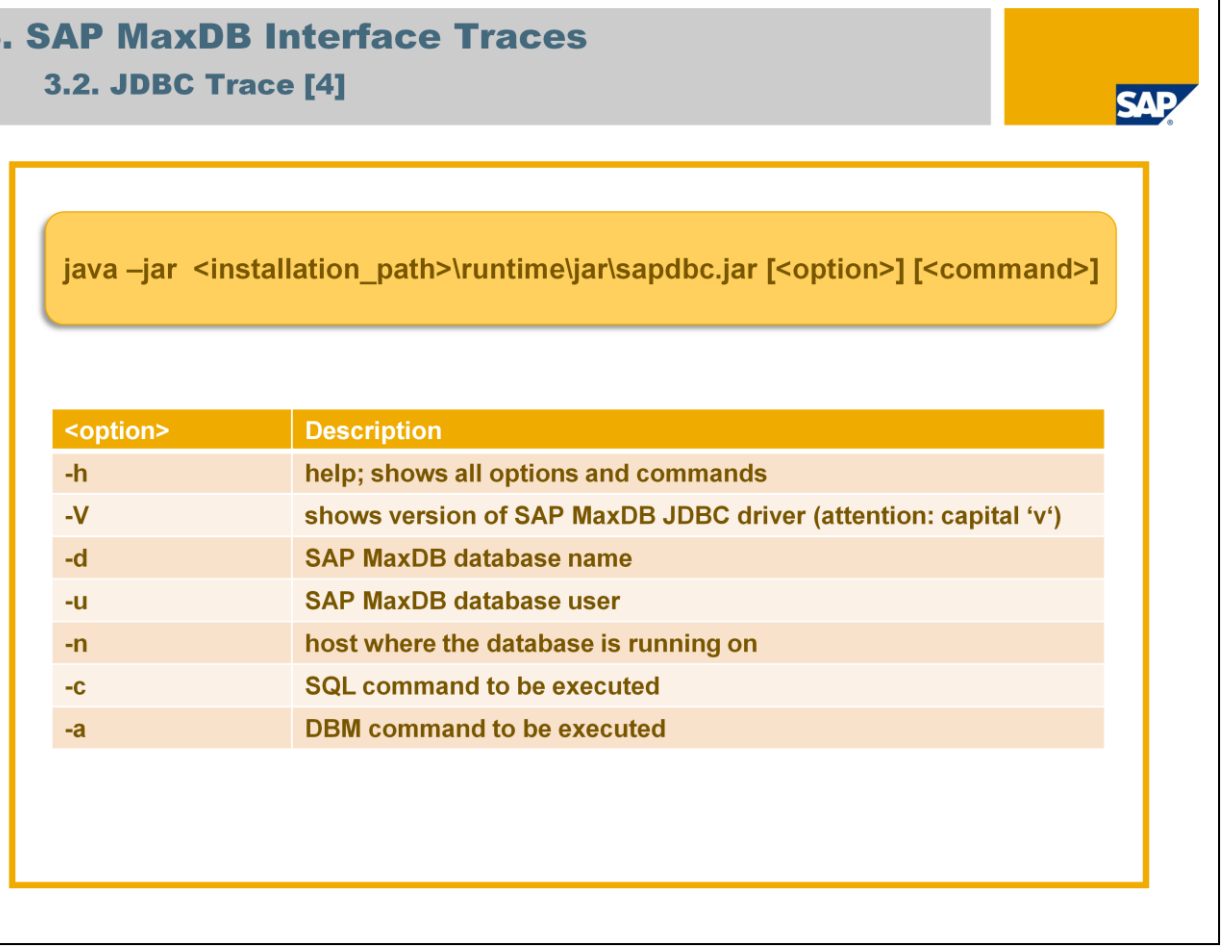
If a JDBC trace is required to analyze an issue in such an environment the properties of user SAPService<SID> could be changed temporarily to allow a logon.

By all means this should be considered as a temporary change which has to be set back after tracing.

On UNIX systems this situation doesn‘t exist as user <SID>adm is a normal user which can be used for logon by default.

3. SAP MaxDB Interface Traces

3.2. JDBC Trace [4]

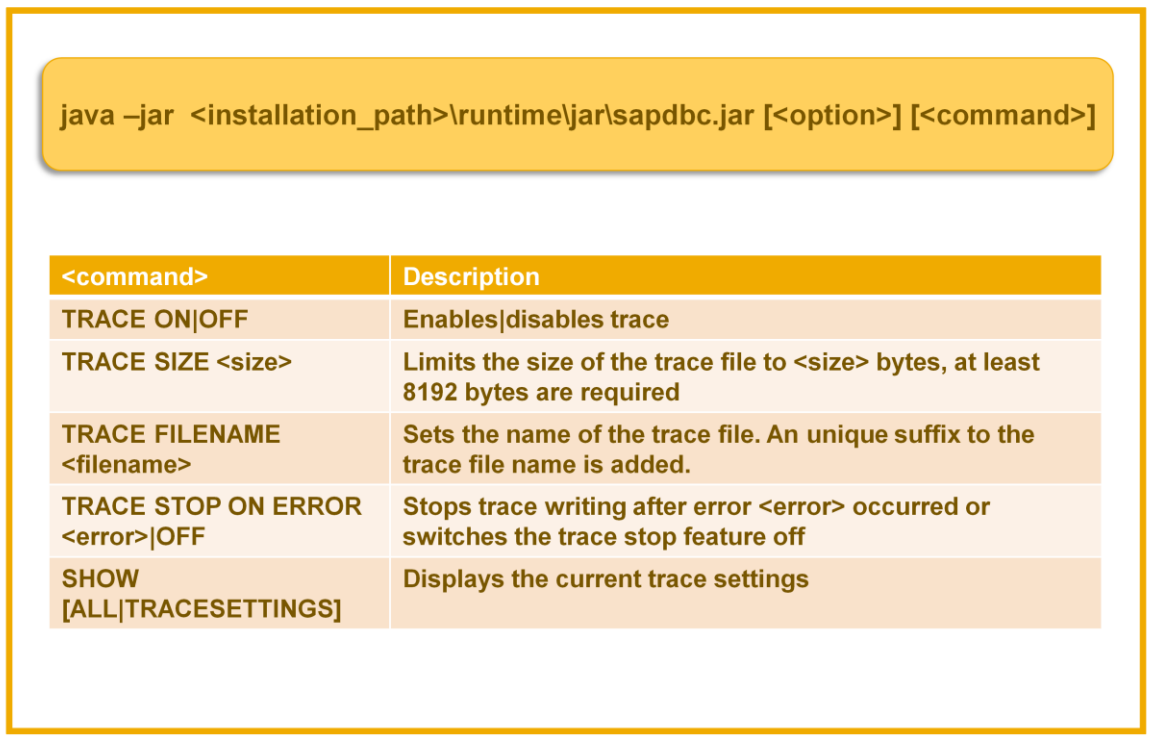


This is a selection of most important options of SAP MaxDB JDBC driver. Via these options SAP MaxDB JDBC trace cannot be administrated.

If a DBM or SQL command is executed via JDBC driver it is required to specify a corresponding user with option ‘-u’ .

3. SAP MaxDB Interface Traces

3.2. JDBC Trace [5]

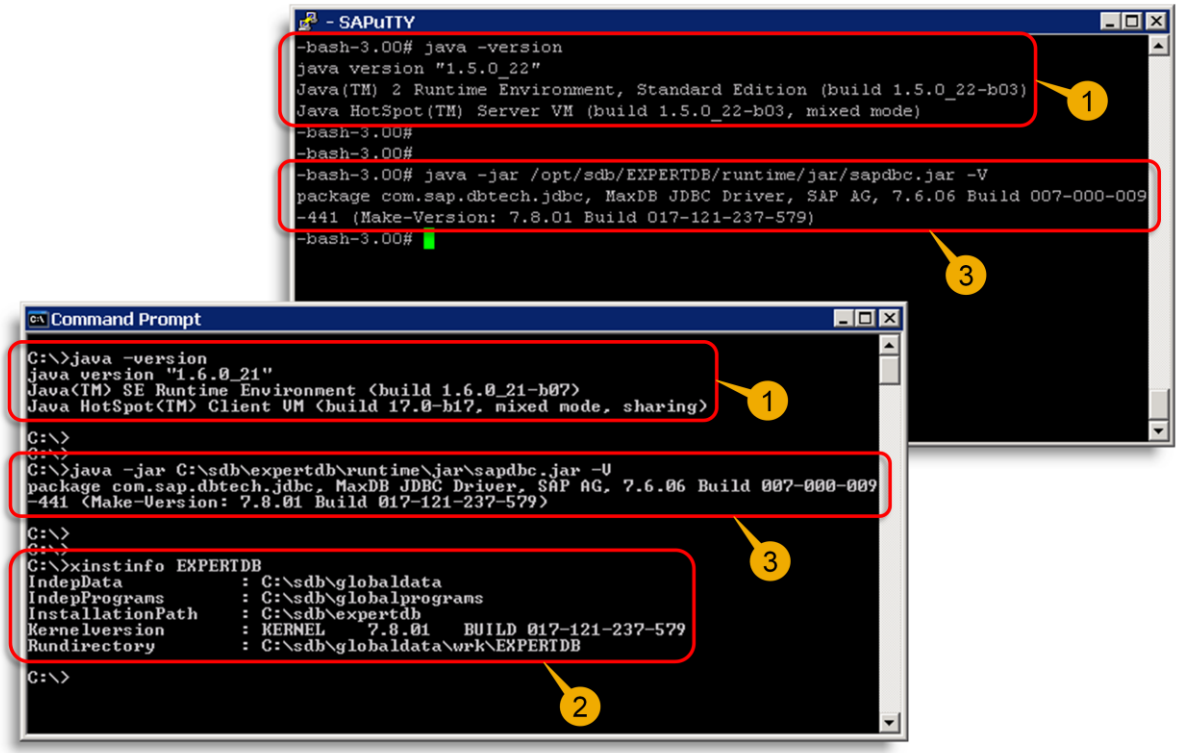


The above listed commands are used for the JDBC trace settings. It is possible to limit the trace file size to avoid space trouble in file system. As for the Database Trace and the SQLDBC Trace a ‘STOP ON ERROR’ feature is available. Command ‘SHOW’ allows to have a look at the current settings including name and location of the trace file.

Only one command can be executed at the same time. Multiple commands have to be executed successively.

3. SAP MaxDB Interface Traces

3.2. JDBC Trace [6]



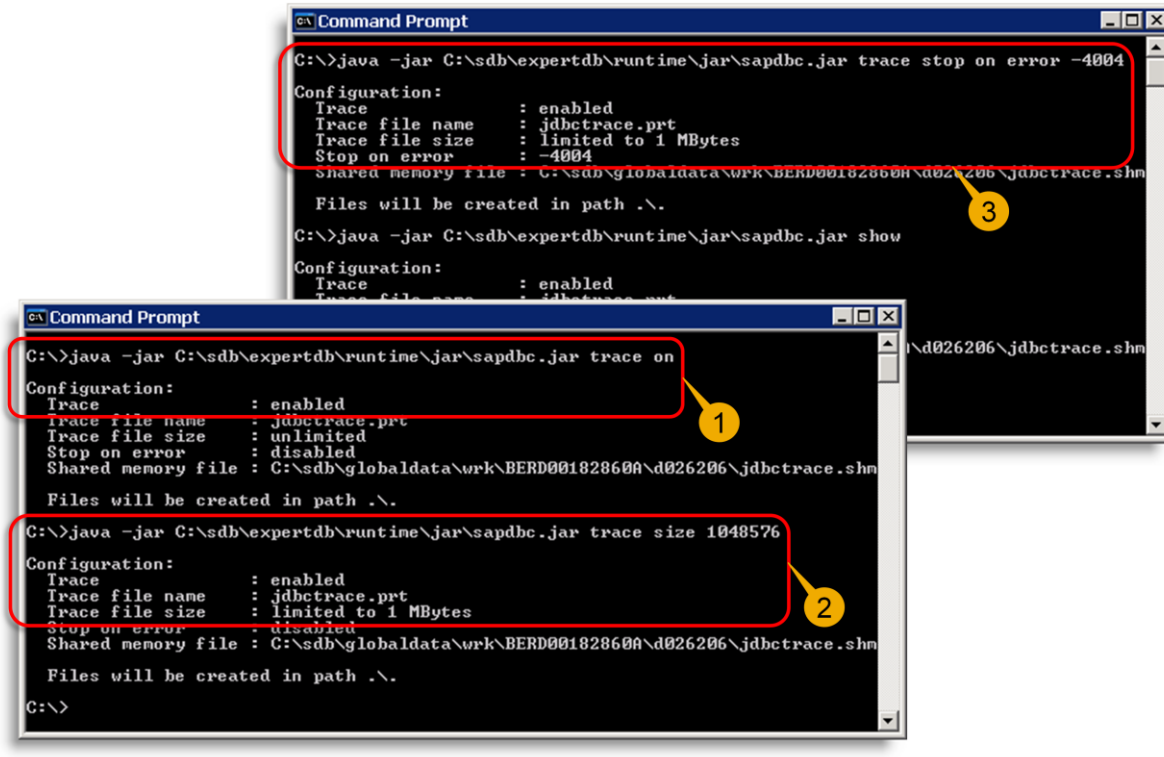
To check the installed java runtime version command ‘java -version‘ is used on Windows systems as well as on UNIX systems. At least version 1.5 is required (1).

SAP MaxDB JDBC driver is called ‘sapdbc.jar‘ and is installed in directory <InstallationPath>/runtime/jar. To figure out the installation path command ‘xinstinfo <SID>‘ can be used. Tool ‘xinstinfo‘ is part of the MaxDB software installation and is located in <IndepPrograms>/bin. Independend programs path can be listed via ‘dbmcli dbm_getpath IndepProgPath‘ (2).

To check the version of the installed SAP MaxDB JDBC driver command ‘java -jar <InstallationPath>/runtime/jar/sapdbc.jar –V‘ is used (pay attention to use a capital ‘v‘) (3).

3. SAP MaxDB Interface Traces

3.2. JDBC Trace [7]

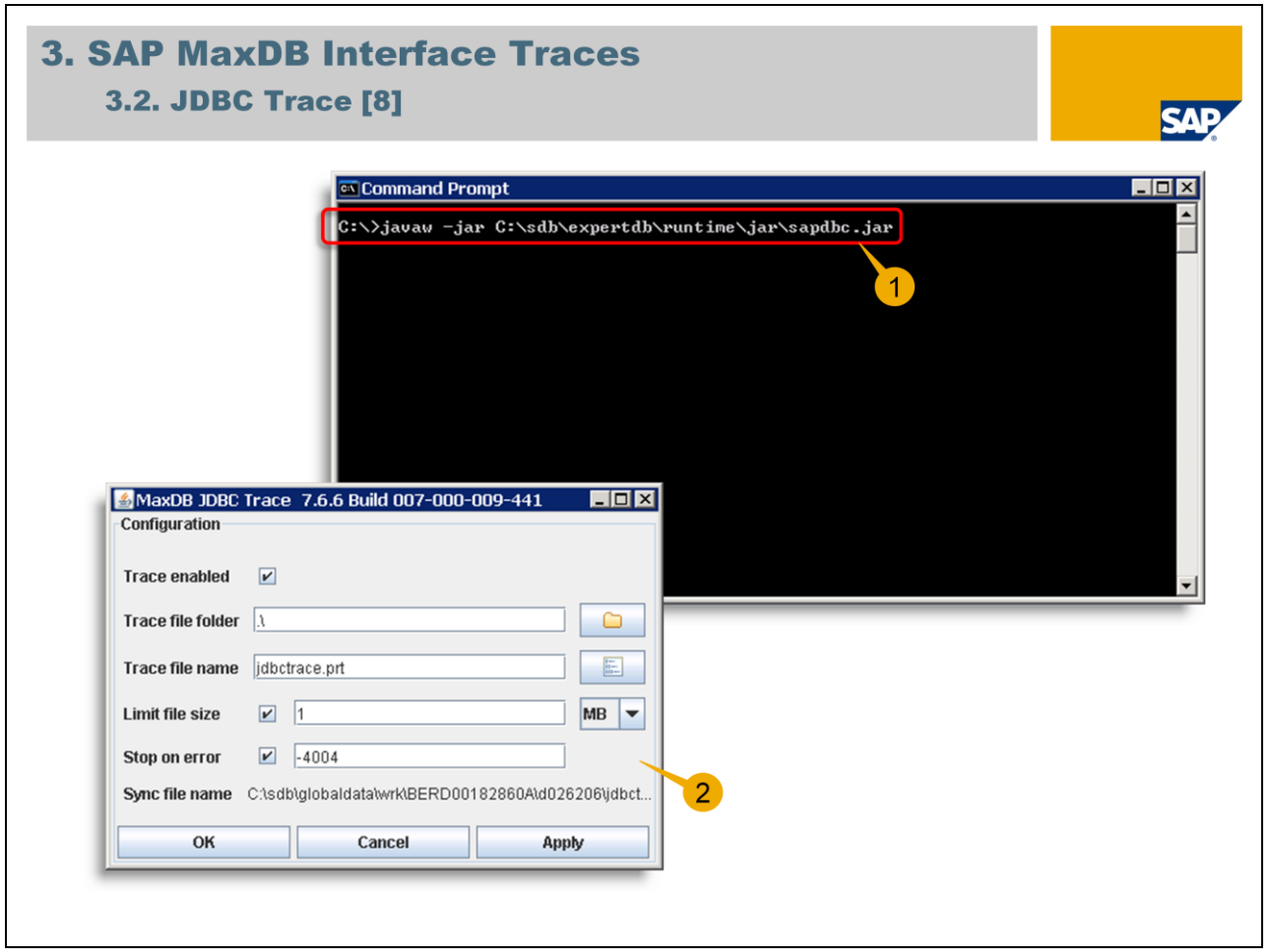


To switch on the JDBC trace command ‘java -jar <InstallationPath>/runtime/jar/sapdbc.jar trace on‘ is used. The trace configuration is listed automatically when executing this command. It now shows that the trace has been enabled (1).

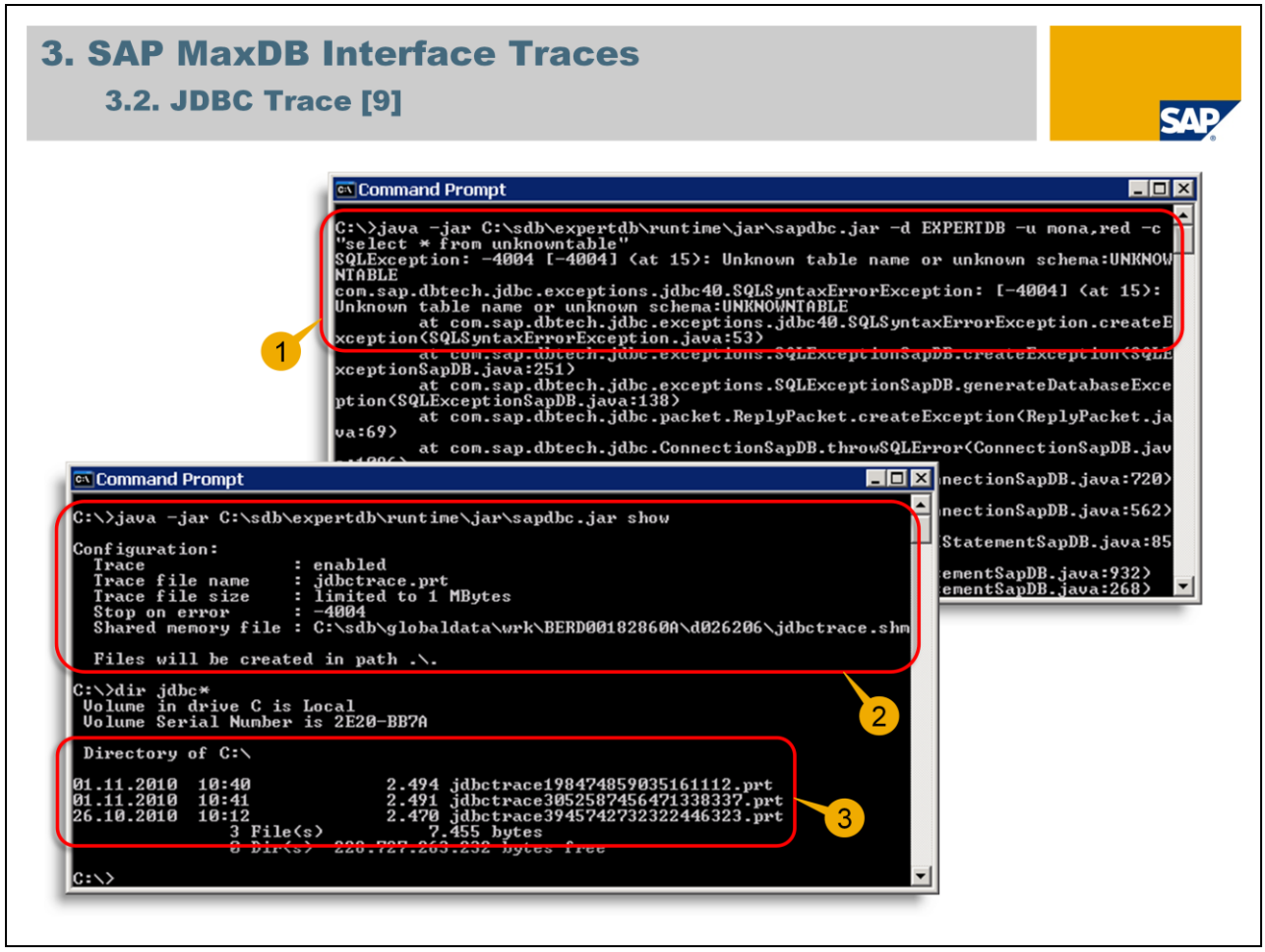
To avoid getting trace files which maybe cause trouble with file system space it is recommended to limit the trace file size. This can be done via command ‘java -jar <InstallationPath>/runtime/jar/sapdbc.jar trace size <size>‘. Unit of <size> is bytes. In this example the trace file size has been limited to 1 MB (2).

It is also possible to get trace writing stopped automatically in the event of occurrence of a specific error. To activate this feature command ‘java -jar <InstallationPath>/runtime/jar/sapdbc.jar trace stop on error <error>‘ has to be executed (3).

Please note: The order of these commands makes no difference.



On Windows systems a more comfortable possibility exists to change the JDBC trace configuration. Executing command ‘javaw -jar <InstallationPath>/runtime/jar/sapdbc.jar‘ opens a graphical interface to view all trace options at a glance and change them if necessary (1) (2).



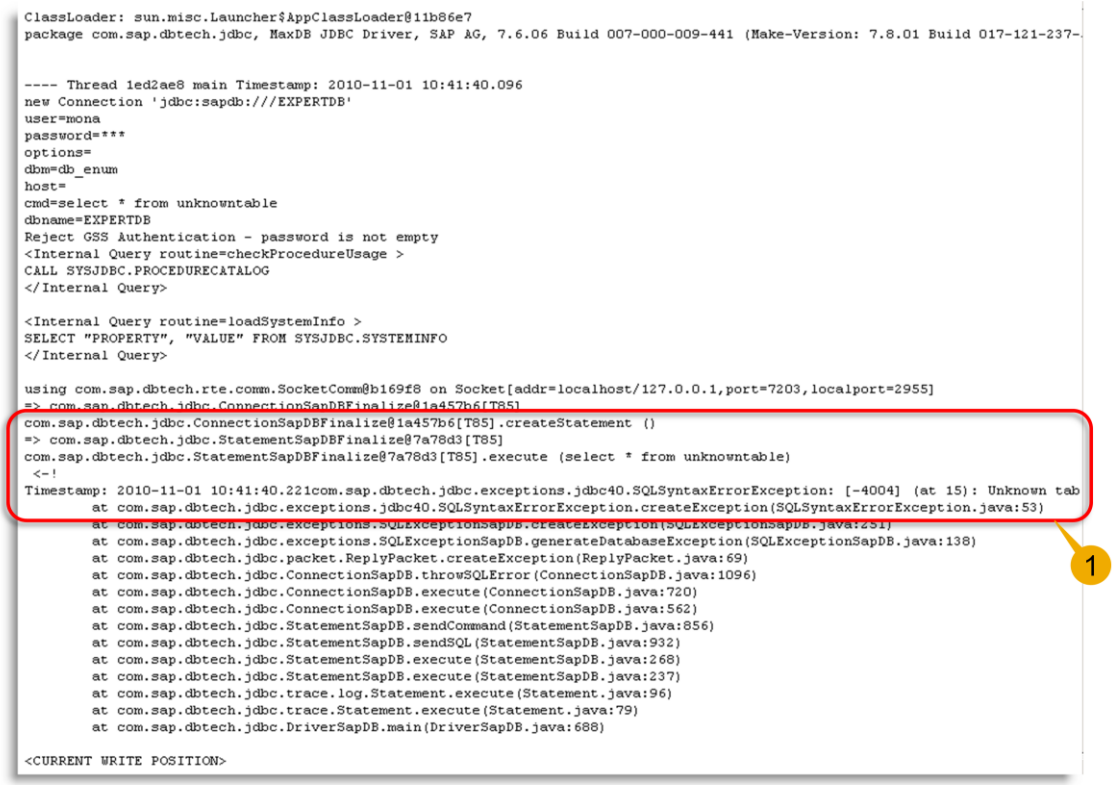
To execute an SQL command via JDBC it is required to specify database name, SQL user and password and of course the SQL command itself. This is done using the options ‘–d‘, ‘-u‘ and ‘–c‘. As a not existing table was chosen here the corresponding error message appears (1).

Command ‘java -jar <InstallationPath>/runtime/jar/sapdbc.jar show‘ always allows to display the current trace configuration including trace file name and the location of the trace files. In this example the files are located in root directory C:\ . Additional hint: In case of option ‘stop on error <error>‘ was set and the configured error occurred the trace configuration will continue to show ‘Trace : enabled‘. So the trace isn‘t switched off but trace writing has been stopped. If a new action is started afterwards which creates trace output a new trace file (with a different number, see below) is generated (2).

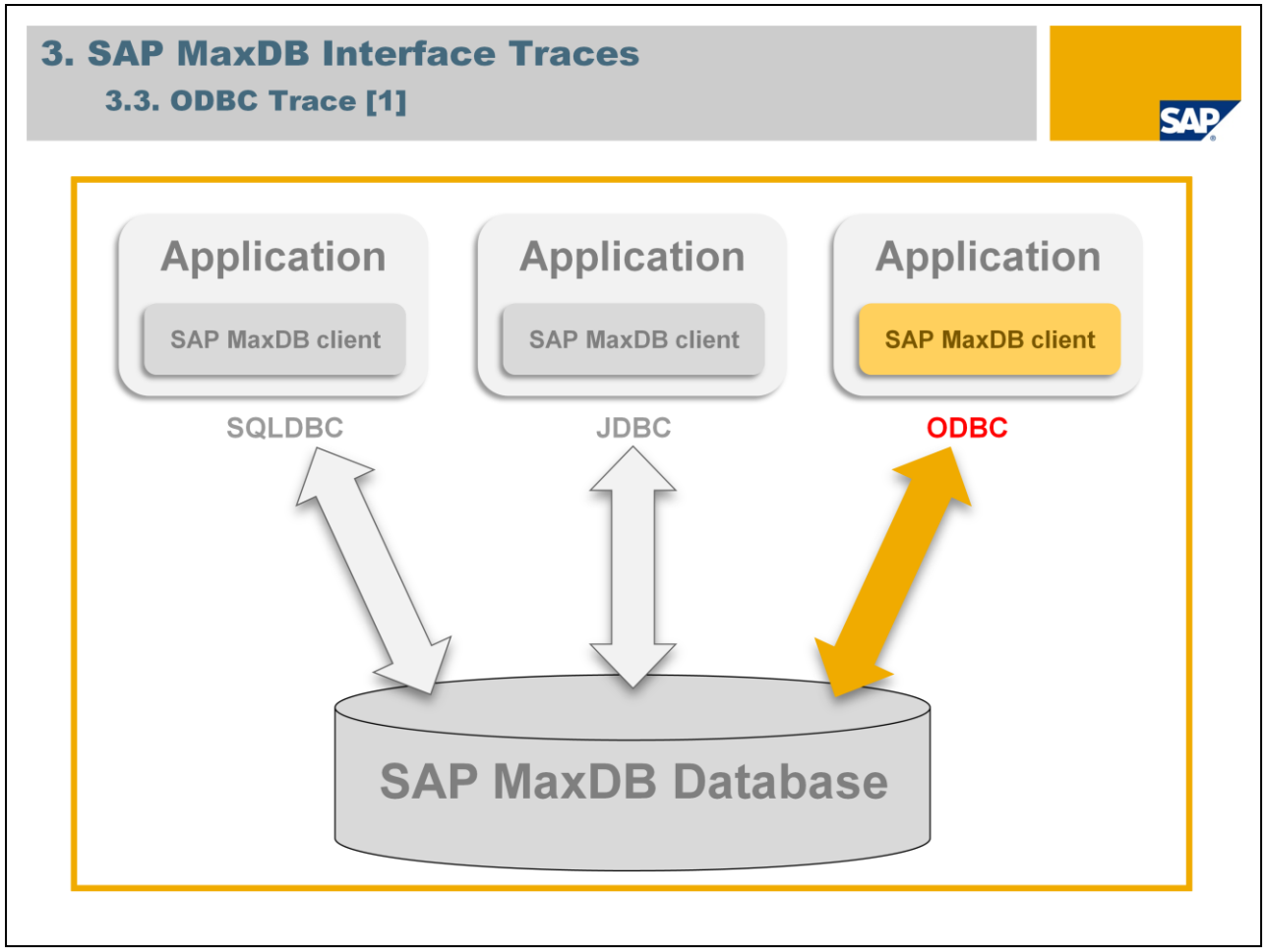
The trace file name gets a number as an appendix. This number looks strange and no context can be recognized at first. It represents the hashcode of the java classloader. This hashcode is used to distinguish the different java threads as there is only one java process. If several trace files are available it is hardly possible to use this number to assign the trace file to a specific trace record. So it is recommended to use the timestamp instead. If several trace files are created in succession it might be a good idea to rename the last ones before creating a new one (3).

3. SAP MaxDB Interface Traces

3.2. JDBC Trace [10]



The trace file is written including HTML tags. So a clear structure is displayed when opening this file e.g. with a reader/editor which considers HTML. Reading the file using a plain text reader is a little bit incommodious. But as said before reading the trace is definitely not expected to be your task. If possible looking for a specific error message is helpful to provide the relevant trace file to the experts (1).



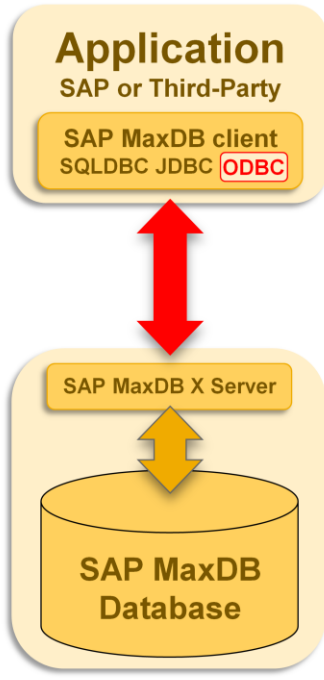
You can use the SAP MaxDB ODBC driver to access SAP MaxDB databases via the ODBC interface.

The SAP Content Server, which is a server component of the Knowledge Provider (Kpro), uses ODBC as an interface for SAP MaxDB.

The SAP MaxDB Database Analyzer requires ODBC in order to connect to the SAP MaxDB database.

3. SAP MaxDB Interface Traces

3.3. ODBC Trace [2]



When it is indicated to use this trace?

- An error occurs in the interface between application and database
- Examples: SQL error, Connect error, HTTP error

Which tools can be used to activate?

SAP MaxDB Tools	SAP Transactions
ODBC_CONS	

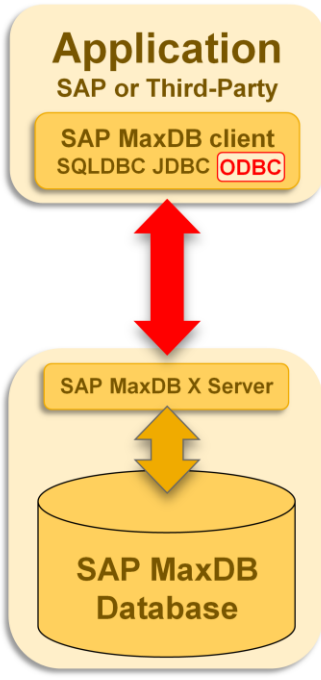
Where is the trace log file located?

- in the directory <indep_data>\wrk (Microsoft Windows) respectively <user_home>/.sdb/<computer name> (UNIX/LINUX) as odbctrace-<pid>.prt file
- Example: C:\sdb\globaldata\wrk\odbctrace-125.prt
- Name, location and size of the trace log file can be changed

The SAP MaxDB ODBC trace logs SQL statements, communication packages and method calls that the database receives and sends via the SAP MaxDB ODBC interface.

3. SAP MaxDB Interface Traces

3.3. ODBC Trace [3]

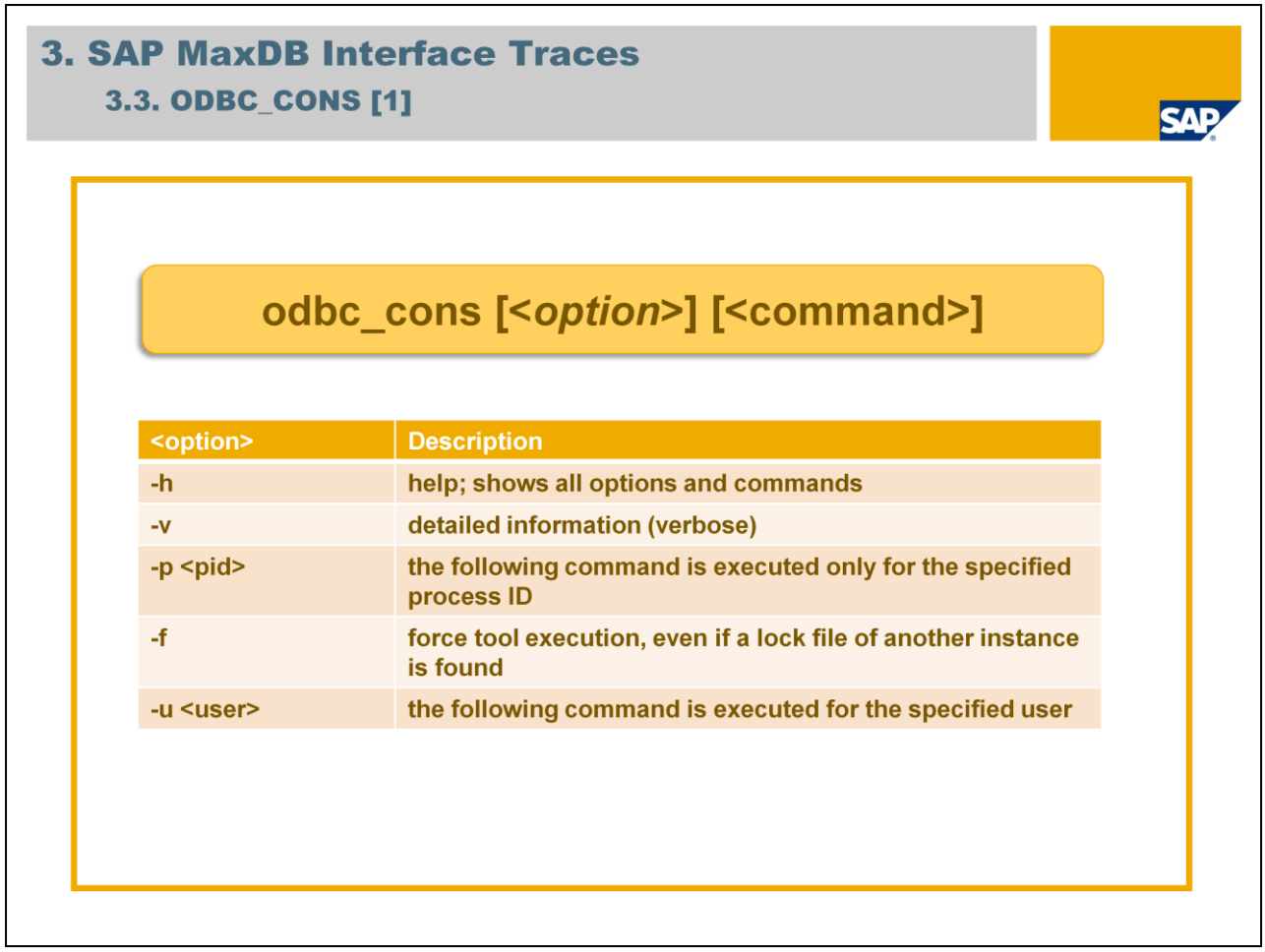


For which user the trace is written?

- The ODBC trace is written for the OS user who switches the trace on.

3. SAP MaxDB Interface Traces

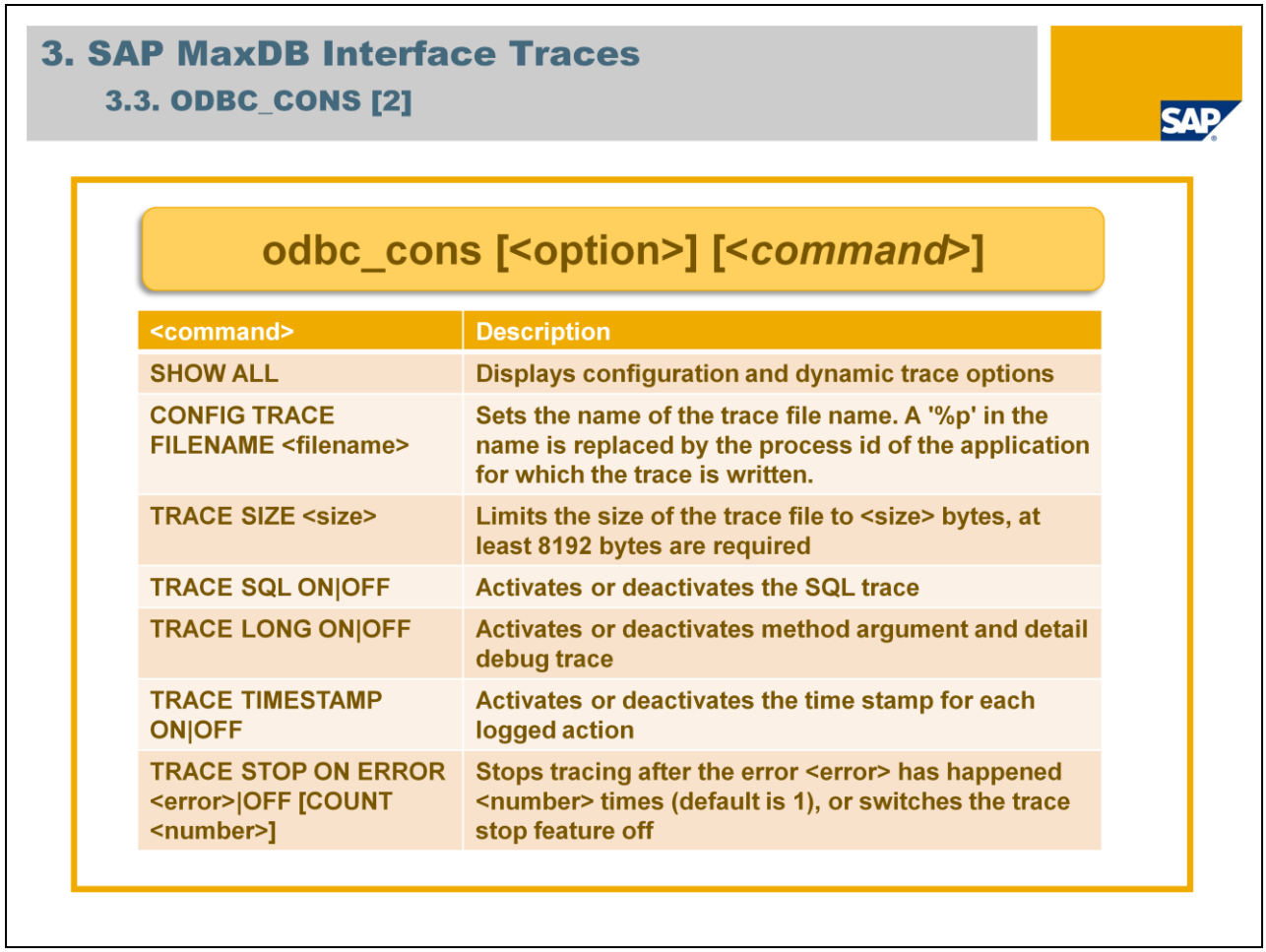
3.3. ODBC_CONS [1]



Note that you can enter only one trace option at a time. They cannot be combined.

3. SAP MaxDB Interface Traces

3.3. ODBC_CONS [2]



odbc_cons [<option>] [<*command*>]

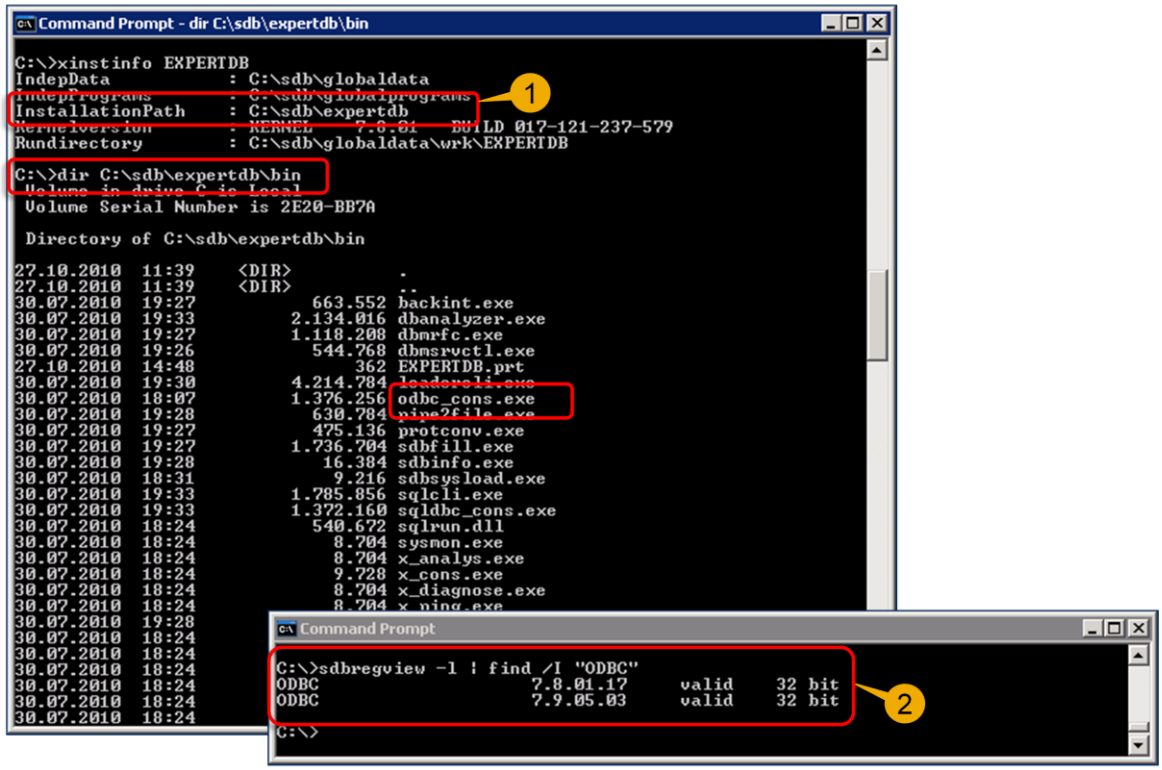
<command>	Description
SHOW ALL	Displays configuration and dynamic trace options
CONFIG TRACE FILENAME <filename>	Sets the name of the trace file name. A '%p' in the name is replaced by the process id of the application for which the trace is written.
TRACE SIZE <size>	Limits the size of the trace file to <size> bytes, at least 8192 bytes are required
TRACE SQL ON|OFF	Activates or deactivates the SQL trace
TRACE LONG ON|OFF	Activates or deactivates method argument and detail debug trace
TRACE TIMESTAMP ON|OFF	Activates or deactivates the time stamp for each logged action
TRACE STOP ON ERROR <error>|OFF [COUNT <number>]	Stops tracing after the error <error> has happened <number> times (default is 1), or switches the trace stop feature off

Only the most important commands are mentioned. To see all available commands option ‘-h’ can be used or have a look at MaxDB documentation.

Again only one command can be executed at the same time.

3. SAP MaxDB Interface Traces

3.3. ODBC_CONS [3]



Tool ‘odbc_cons’ is part of the MaxDB software installation and is located in <InstallationPath>/bin. To figure out the installation path command ‘xinstinfo <SID>‘ can be used (1).

You can use the program 'sdbregview' to determine which ODBC versions are installed on the relevant host.

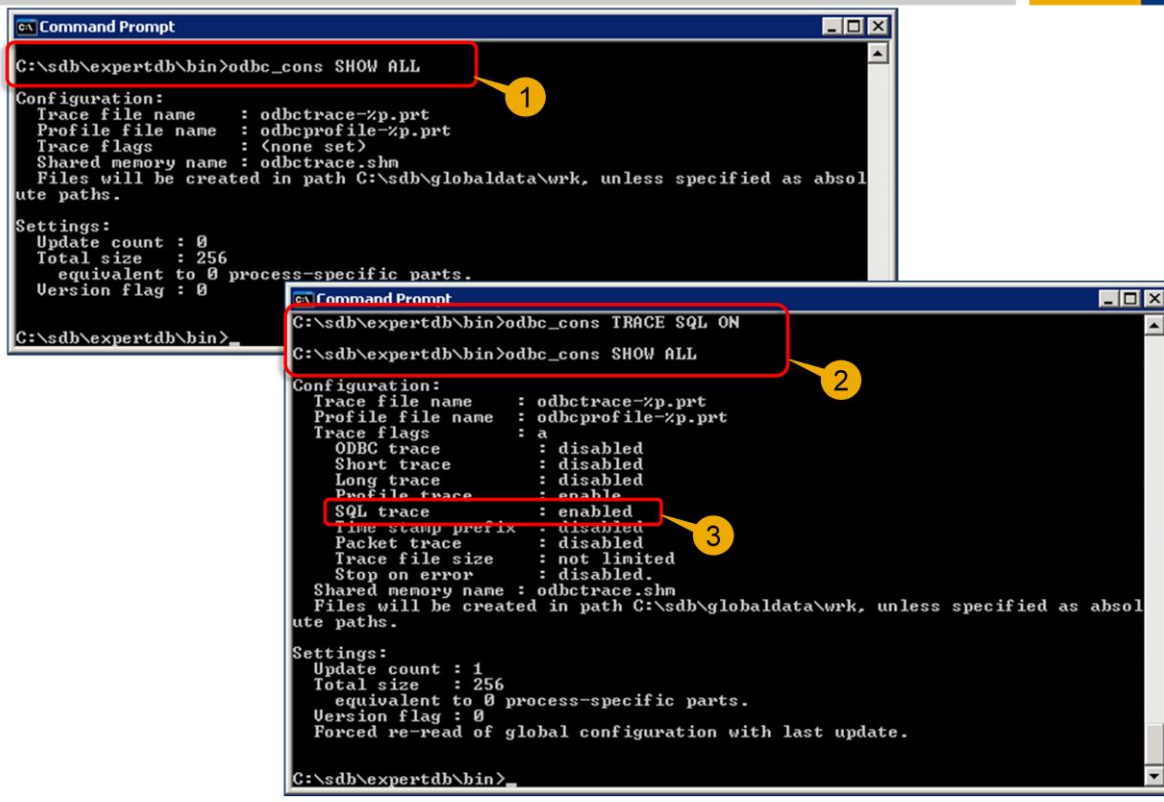
a) On UNIX platforms: 'sdbregview –l | grep –I ODBC'

b) On Windows platforms: 'sdbregview -l | find /I “ODBC"'

Several ODBC versions can be installed (2).

3. SAP MaxDB Interface Traces

3.3. ODBC_CONS [4]



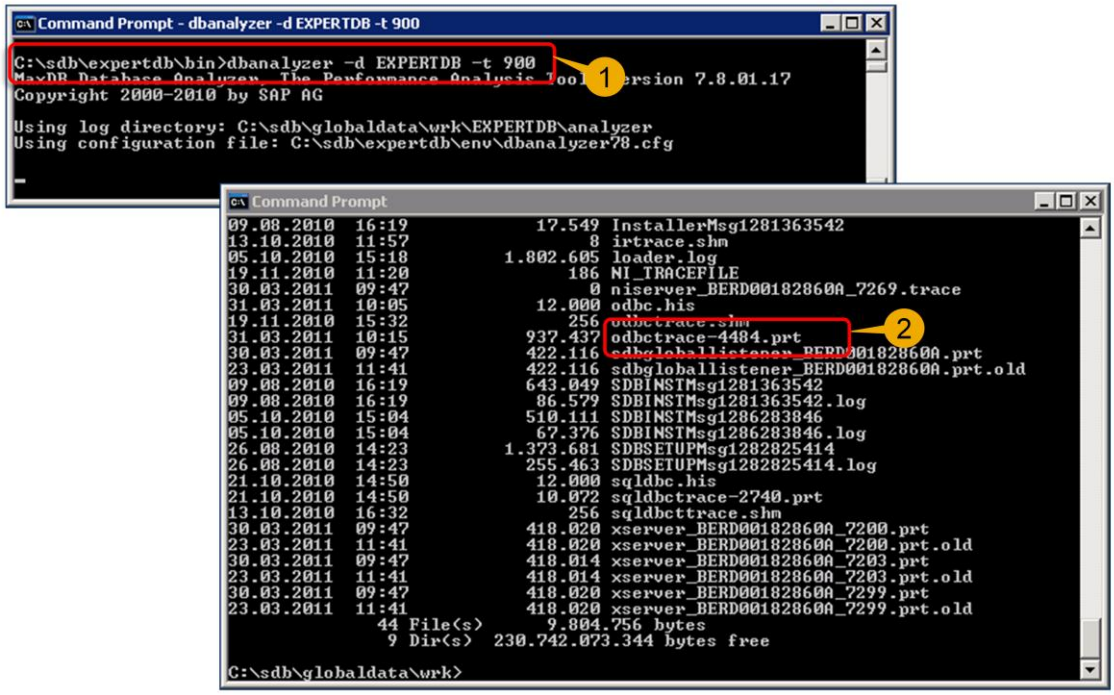
ODBC trace configuration is listed with command ‘odbc_cons SHOW ALL’ (1).

Trace file name is set to ‘odbctrace-%p.prt’. The ‘%p’ will be replaced by the process id. Trace file will be created in ‘C:\sdb\globaldata\wrk’.

To switch on SQL trace command ‘odbc_cons TRACE SQL ON’ is used. With command ‘odbc_cons SHOW ALL’ the new trace configuration is displayed (2) (3).

3. SAP MaxDB Interface Traces

3.3. ODBC_CONS [5]



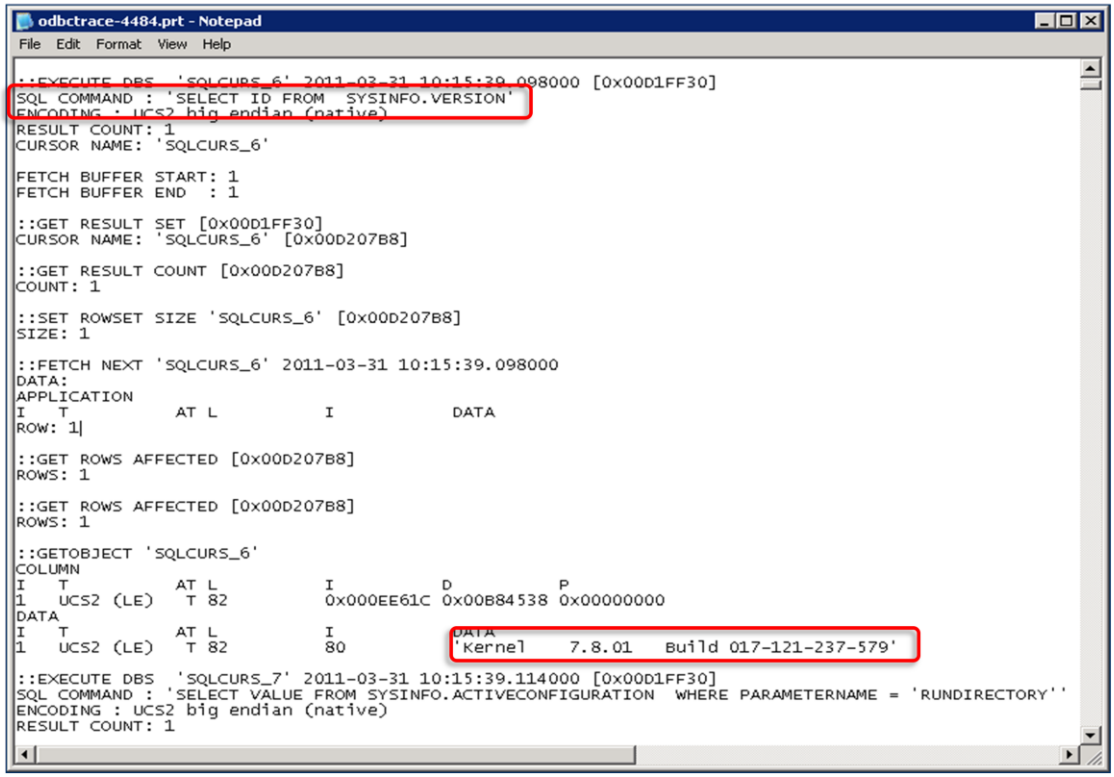
One application which is using ODBC driver to connect to MaxDB instances and execute SQL statements is the performance analyzing tool SAP MaxDB Database Analyzer. It is a component of the MaxDB software.

To start SAP MaxDB Database Analyzer for database EXPERTDB command ‘dbanalyzer –d EXPERTDB –t 900’ is used (1). The option ‘-t 900’ specifies the time interval between two analyses (in seconds).

In directory ‘C:\sdb\globaldata\wrk’ ODBC trace file ‘odbctrace-4484.prt’ is created (2).

3. SAP MaxDB Interface Traces

3.3. ODBC_CONS [6]

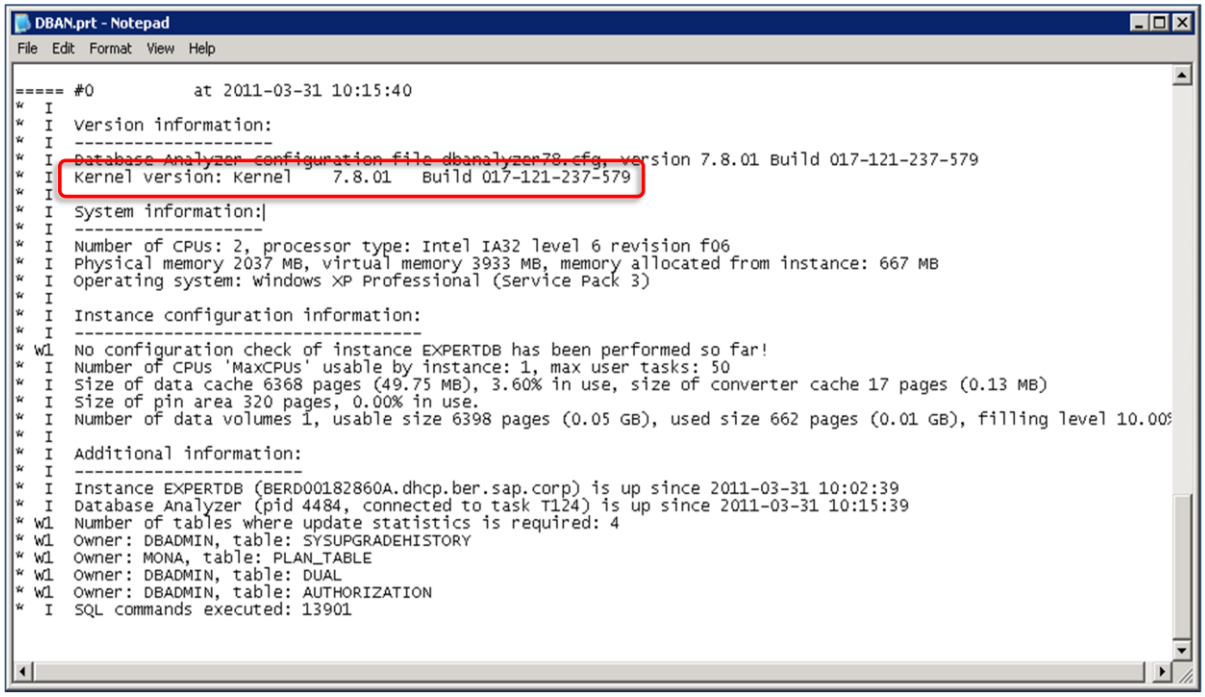


ODBC trace file ‘odbctrace-4484.prt’ contains the executed SQL statements, their parameters and results. SAP MaxDB Database Analyzer reads the database kernel version from system table SYSINFO.VERSION for instance.

Reading the trace in detail is the job of MaxDB expert.

3. SAP MaxDB Interface Traces

3.3. ODBC_CONS [7]



SAP MaxDB Database Analyzer logs the selected information in file DBAN.prt in directory <RUNDIRECTORY>\analyzer\<date>, for instance C:\sdb\globaldata\wrk\EXPERTDB\analyzer\20110331.

Don’t forget to switch off ODBC trace with command ‘odbc_cons TRACE SQL OFF’.

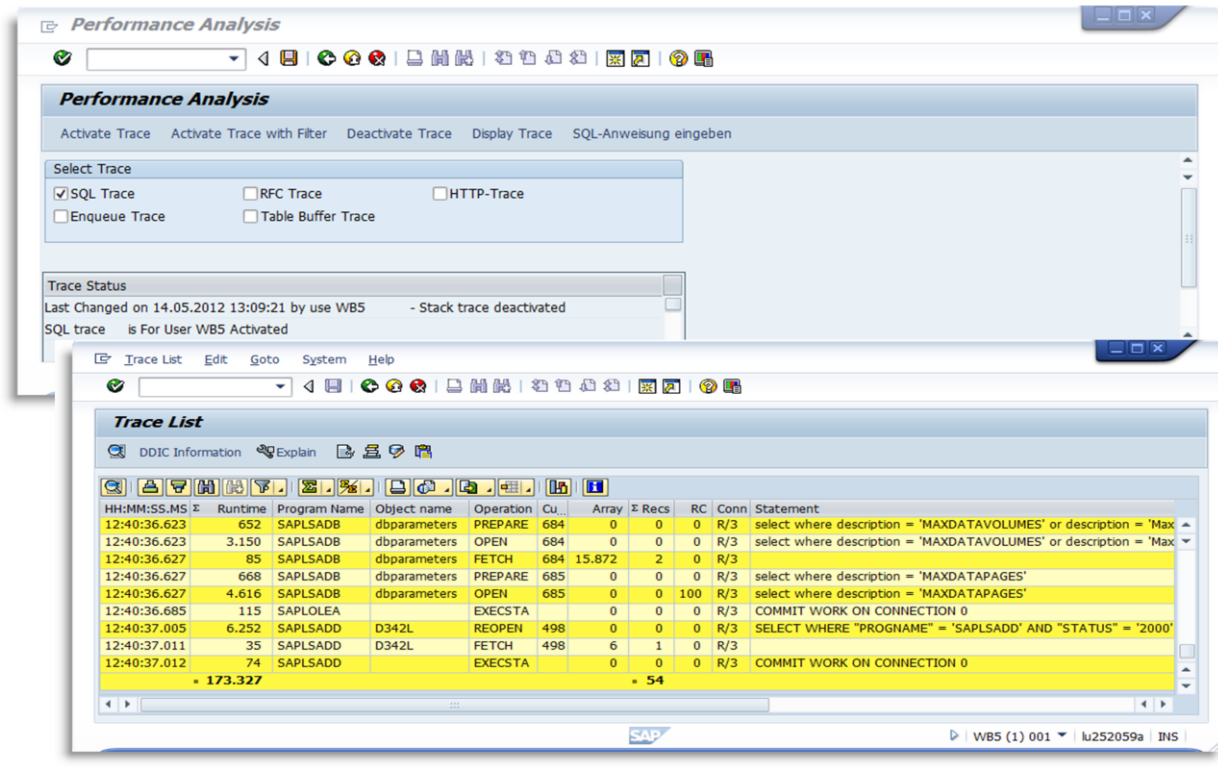
Agenda

1. Introduction
2. SAP MaxDB Database Trace
3. SAP MaxDB Interface Traces
4. Additional Traces
5. Useful Information Resources

This chapter provides information about traces in the SAP system which are not MaxDB specific but often used for an analysis of problems with a MaxDB instance (SQL trace and developer trace). Additionally one more MaxDB specific trace - DBMRFC trace – is mentioned briefly. This interface is only be used in SAP systems.

4. Additional Traces

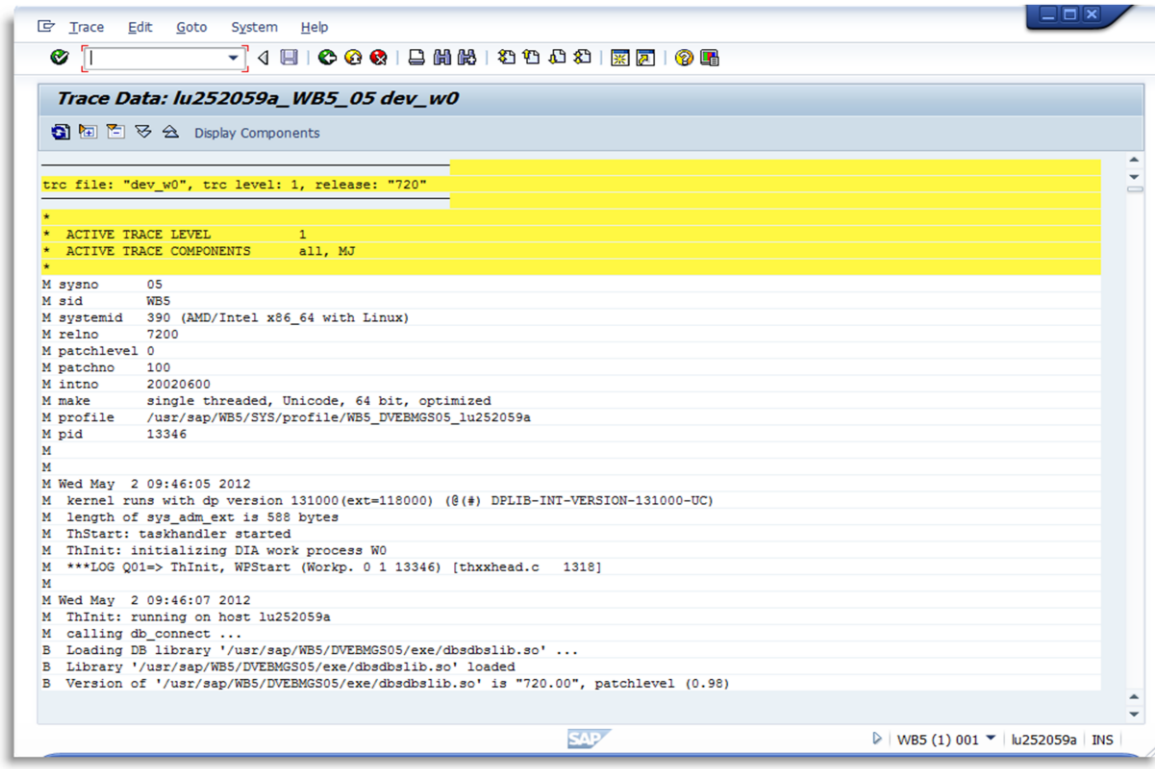
4.1. SQL Trace



The SAP system provides the possibility to write a trace which logs all SQL statements sent to the database. This SQL trace can be activated via transaction ST05.
The SQL trace is mainly used to analyze performance problems. Furthermore it is used to determine which native SQL statement is responsible for an error during the execution of a specific transaction.

4. Additional Traces

4.2. Developer Trace

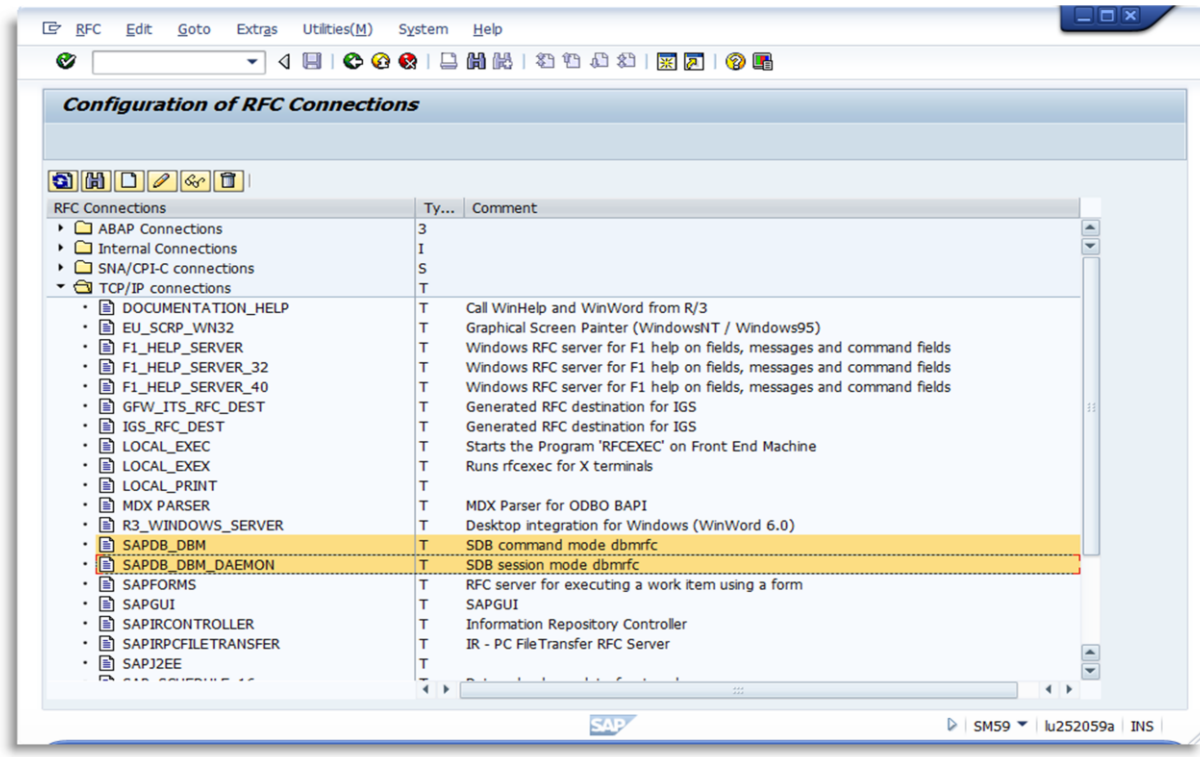


The developer trace (files dev_w* in directory /usr/sap/<SID>/DVEBMG<SID>/work) is always written - you don't need to activate it manually. In these trace files you can find e.g. information about connects from the work processes to the database instance. You can also find information about the used SAP kernel version, DBSL version, precompiler or SQLDBC version and the database version.
It is possible to configure different trace levels to log more detailed information in these trace files.

To display the trace output you can use either transaction AL11 (directory <DIR_HOME>) or transaction SM50.

4. Additional Traces

4.3. DBMRFC Trace [1]



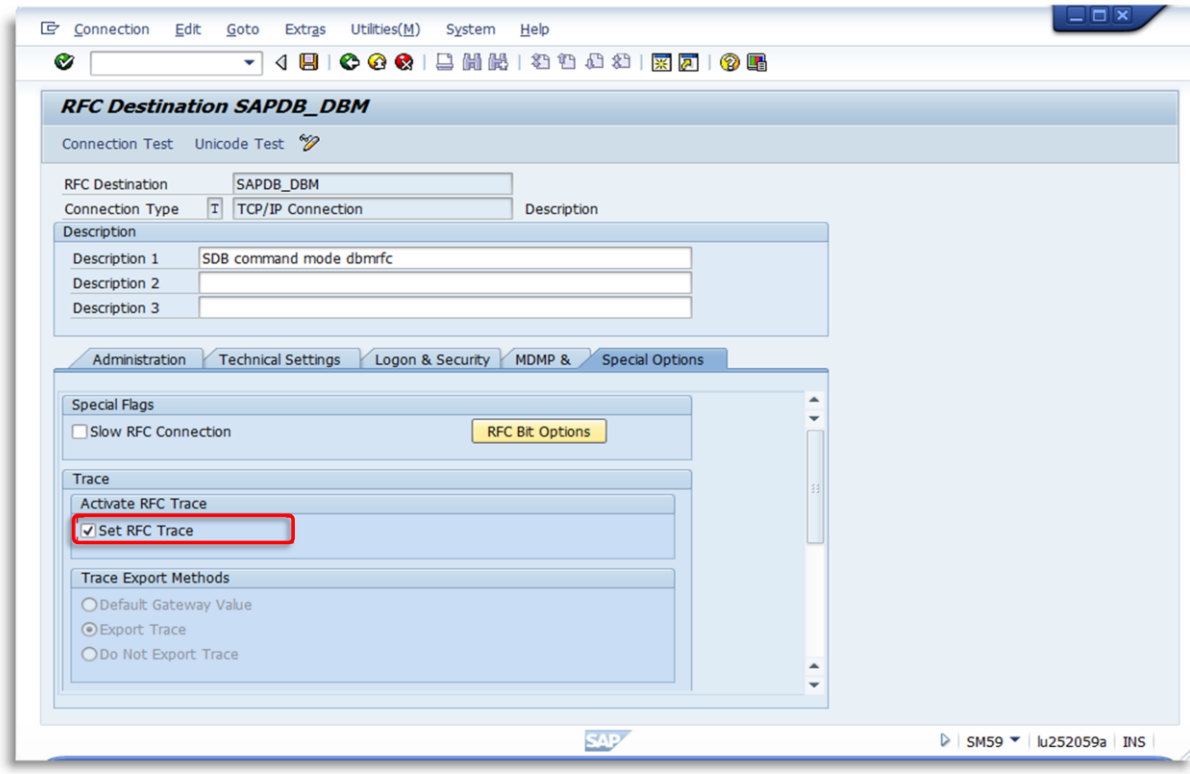
The DBMRFC trace is mainly used to analyze connection problems in the SAP MaxDB-specific CCMS transactions (LC10, DB50, DBACOCKPIT). SAP application servers connect via DBMRFC to DBMServer processes of a SAP MaxDB database server. In the meaning of the SAP RFC feature DBMRFC is an RFC server.

There are two kinds of TCP/IP connections (transaction SM59) used for DBMRFC to connect to a SAP MaxDB database:

- SAPDB_DBM is used for DBMRFC in "command mode“
- SAPDB_DBM_DAEMON is used for DBMRFC in "session mode"

4. Additional Traces

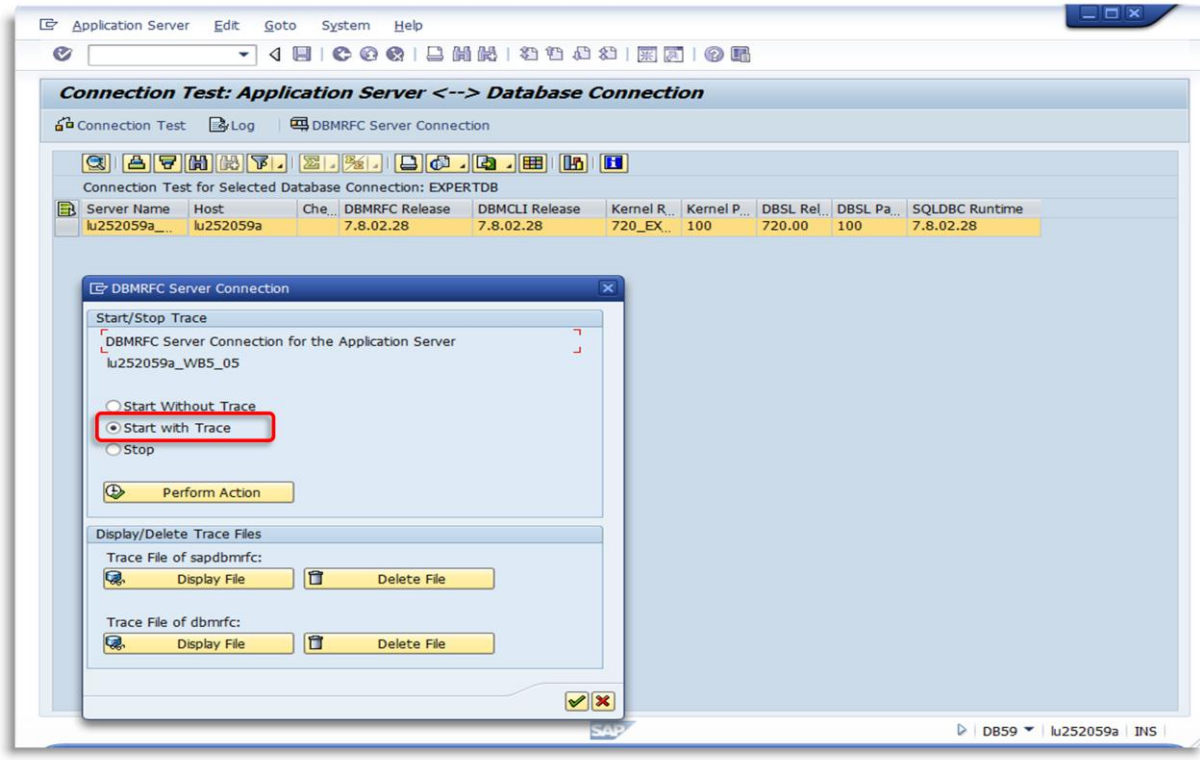
4.3. DBMRFC Trace [2]



To turn on the DBMRFC trace for destination SAPDB_DBM transaction SM59 has to be used. Choose tab "Special Options“ and activate check box "Set RFC Trace“.

4. Additional Traces

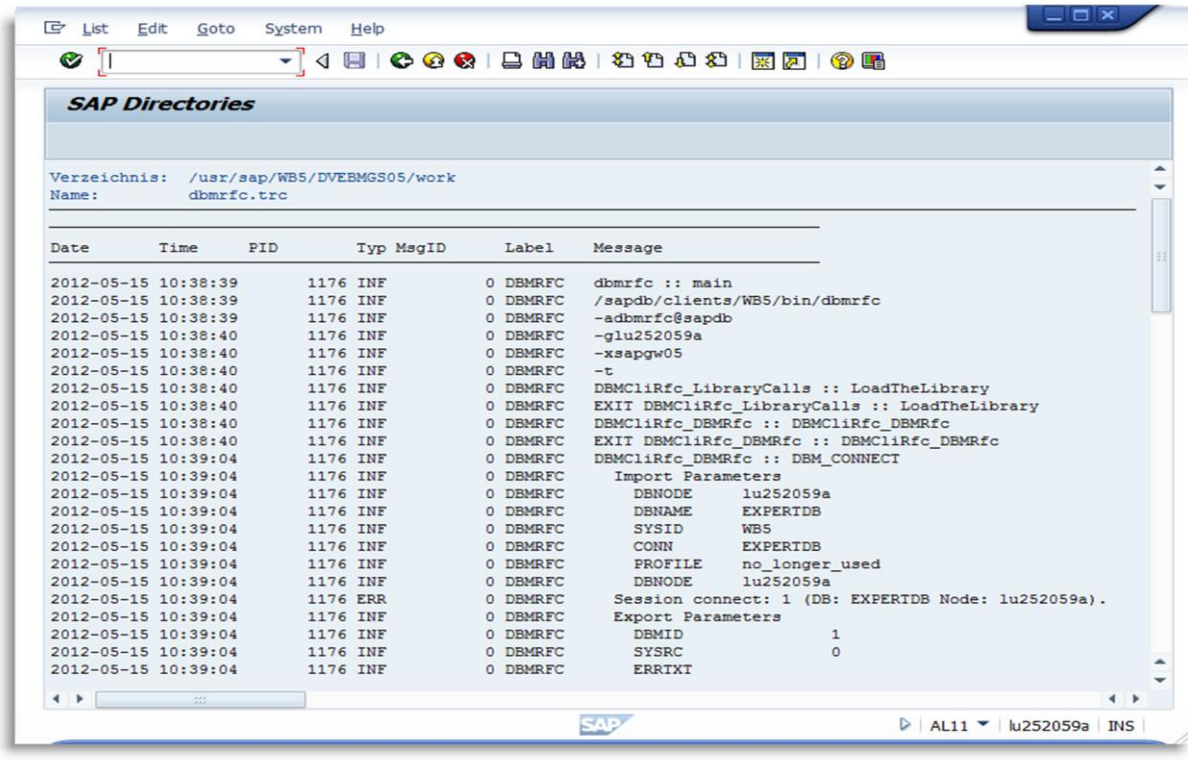
4.3. DBMRFC Trace [3]



To turn on the DBMRFC trace for destination SAPDB_DBM_DAEMON transaction DB59 has to be used. Select the line for the corresponding database connection and choose "Connection Test" (e.g. EXPERTDB). Next step is to choose "DBMRFC Server Connection" and radio button "Start with Trace".

4. Additional Traces

4.3. DBMRFC Trace [4]



The output of the DBMRFC trace is logged in file dbmrfc.trc which can be found in SAP work directory DIR_HOME (transaction AL11).

Agenda

1. Introduction
2. SAP MaxDB Database Trace
3. SAP MaxDB Interface Traces
4. Additional Traces
5. Useful Information Resources

5. Useful Information Resources

- **SAP MaxDB documentation:**

  http://maxdb.sap.com

  following ➡ “Documentation” ➡ “Version 7.8”

  ➡ “Glossary” ➡ “Trace”

- **SAP notes:**

  837385 “FAQ: SAP MaxDB database trace (VTRACE)”

  822239 “FAQ: SAP MaxDB Interfaces”

  1428709 “Creating ODBC trace as of SAP MaxDB Version 7.7“

  903018 “SAP MaxDB: JDBC trace“

- **SAP Community Network (SCN):**

  http://wiki.sdn.sap.com/wiki/display/MaxDB/SAP+MaxDB+traces

  (SAP MaxDB traces)

Questions and Answers

Thank You!
Bye, Bye – And Remember Next Session

August 14, 2012	**SAP MaxDB No-Reorganization Principle**
	All Expert Sessions: http://maxdb.sap.com/training

Disclaimer

This presentation reflects current planning.

Contents may be changed without prior notice, and are in no way binding upon SAP.

101