Managing MaxDB

SPC150

Version 7.6

Content

- What Has to be Monitored?
- Standby Database
- Central Monitoring – Transaction DB59
- Database Assistant – Transaction DB50
- Performance Analysis Tools
- Alert Monitor – Transaction RZ20
- Snapshot

Learning Objectives

As a result of this workshop, you will be able to:

- Integrate your MaxDB instances into your monitoring landscape in transaction DB59.
- Use transaction DB50 to monitor your MaxDB instances.
- Use the MaxDB performance analysis tools to determine performance bottlenecks.
- Create a standby database and snapshots.
- Activate the Alert Monitor for your MaxDB instances.
- Schedule backups and other administrative tasks using the DBA Planning Calendar.

Chapter

What Has to be Monitored?

Standby Database

Central Monitoring – Transaction DB59

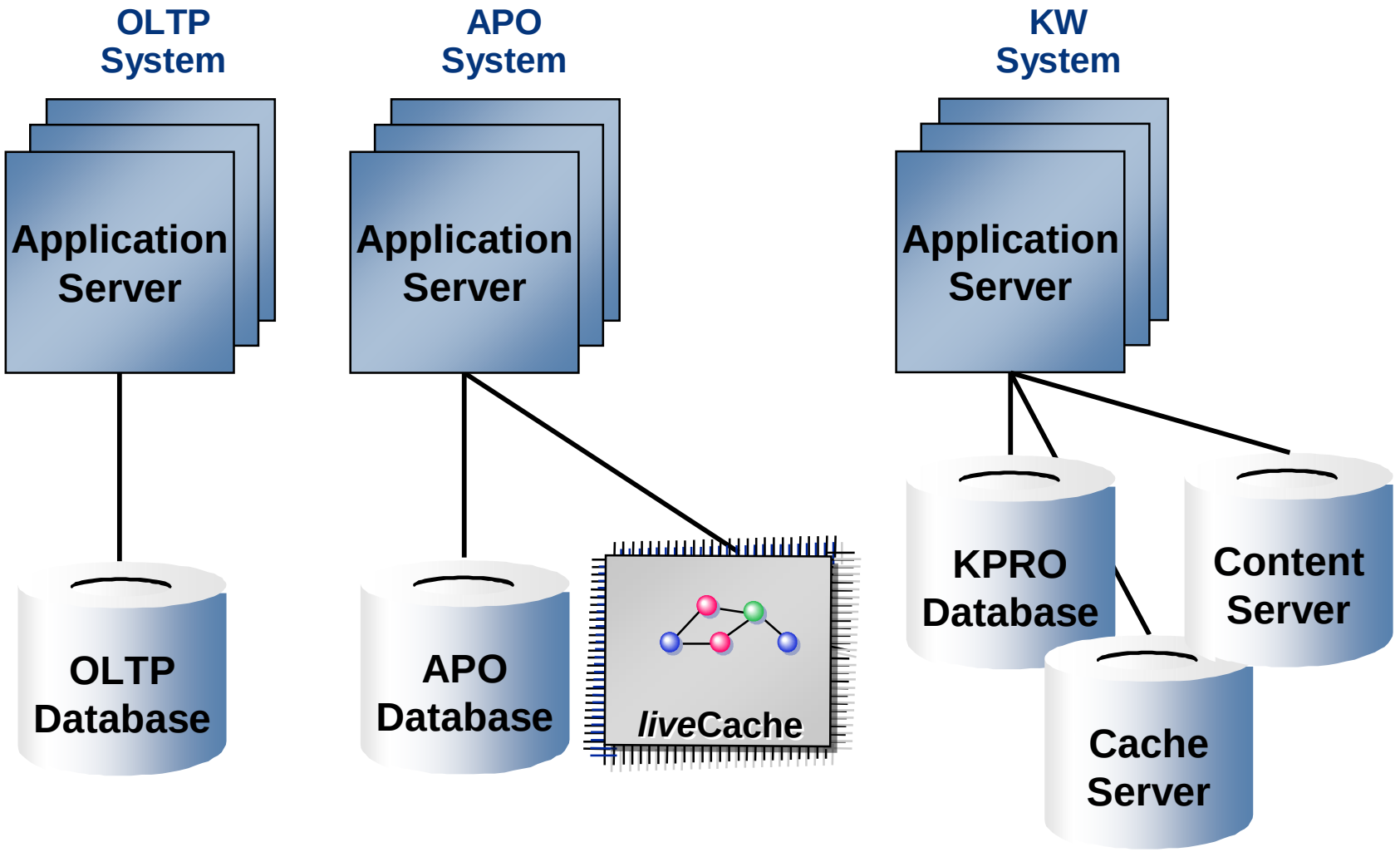
Database Assistant – Transaction DB50

Performance Analysis Tools

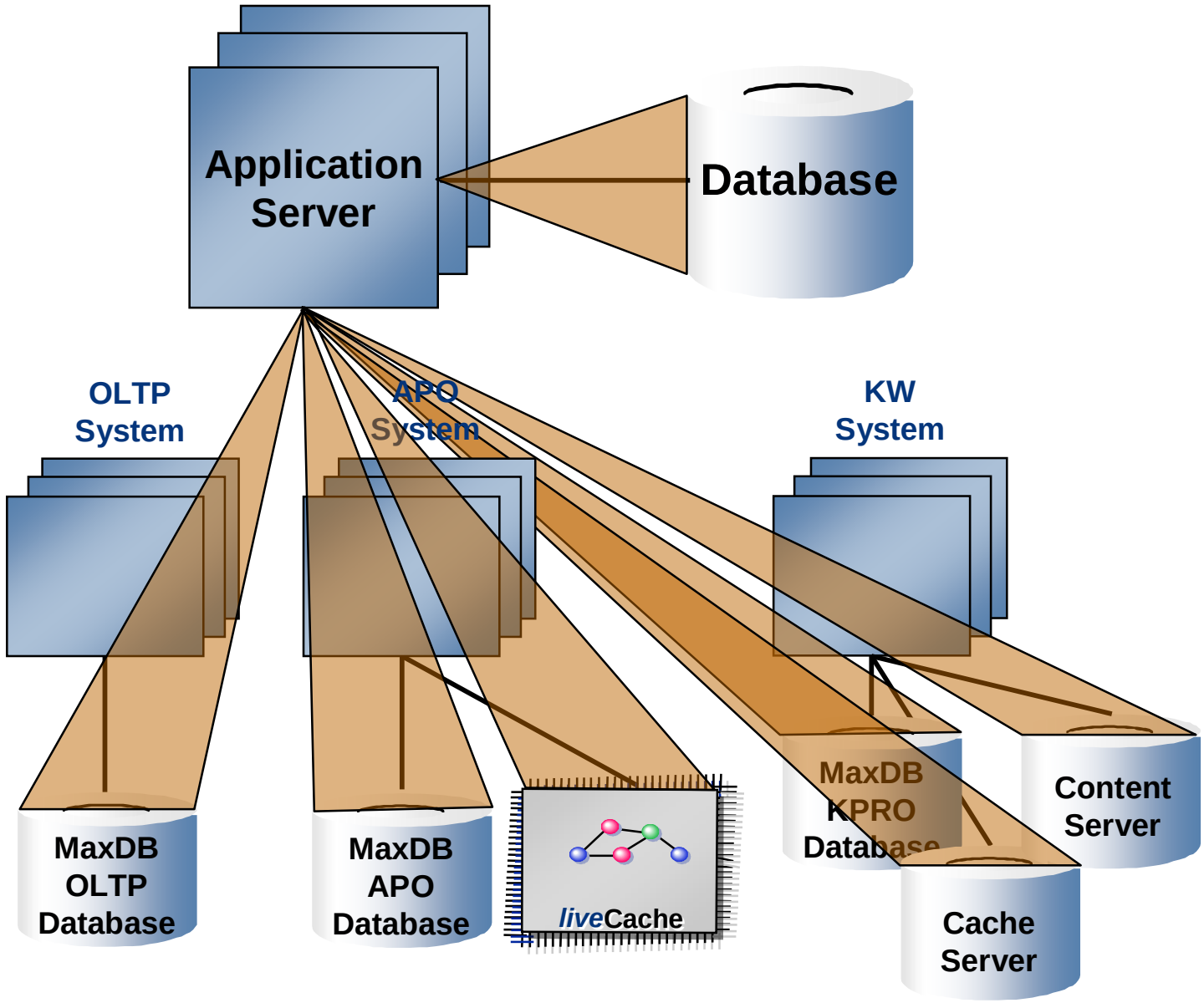
Alert Monitor – Transaction RZ20

Snapshot

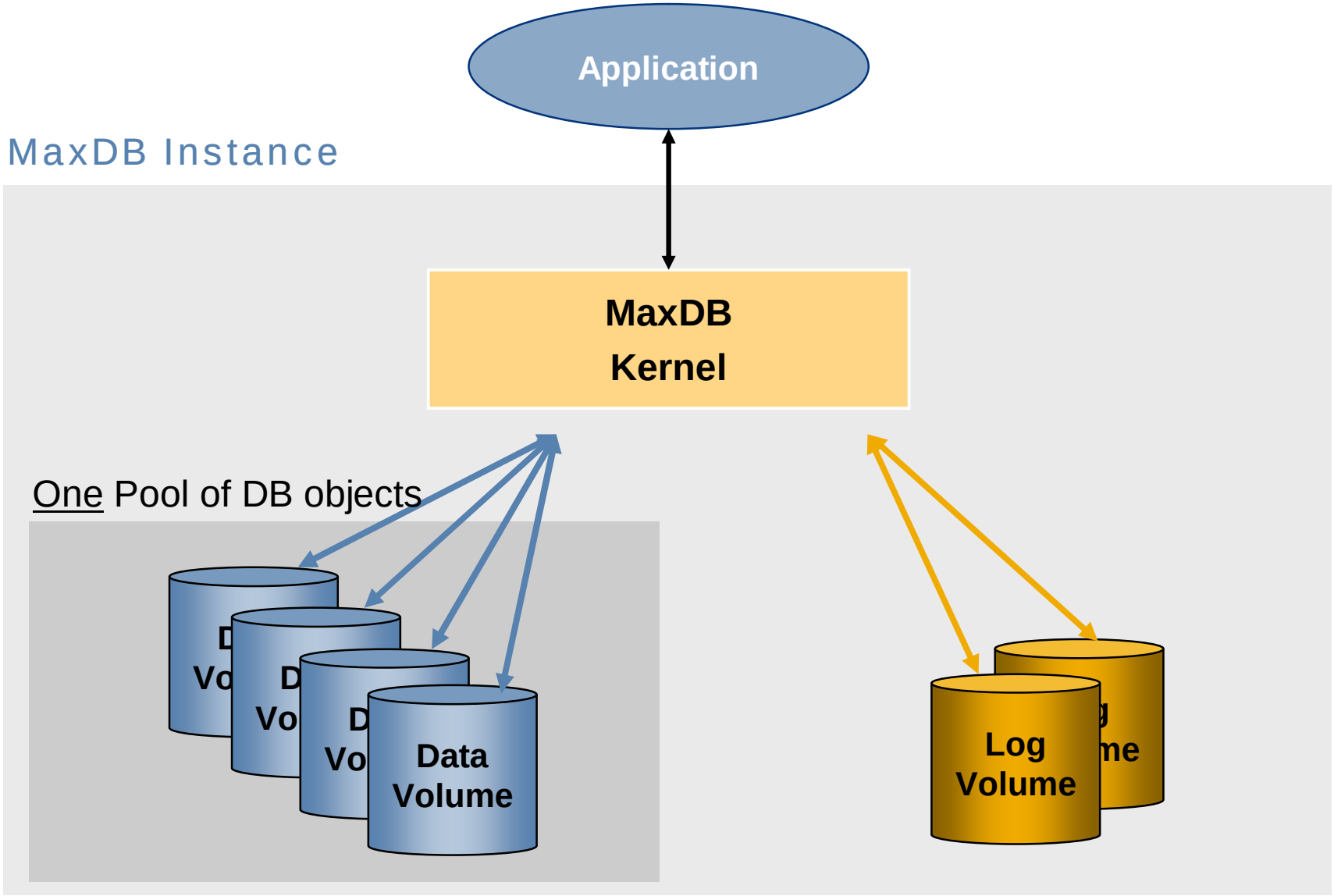
Which Instances can be Monitored?



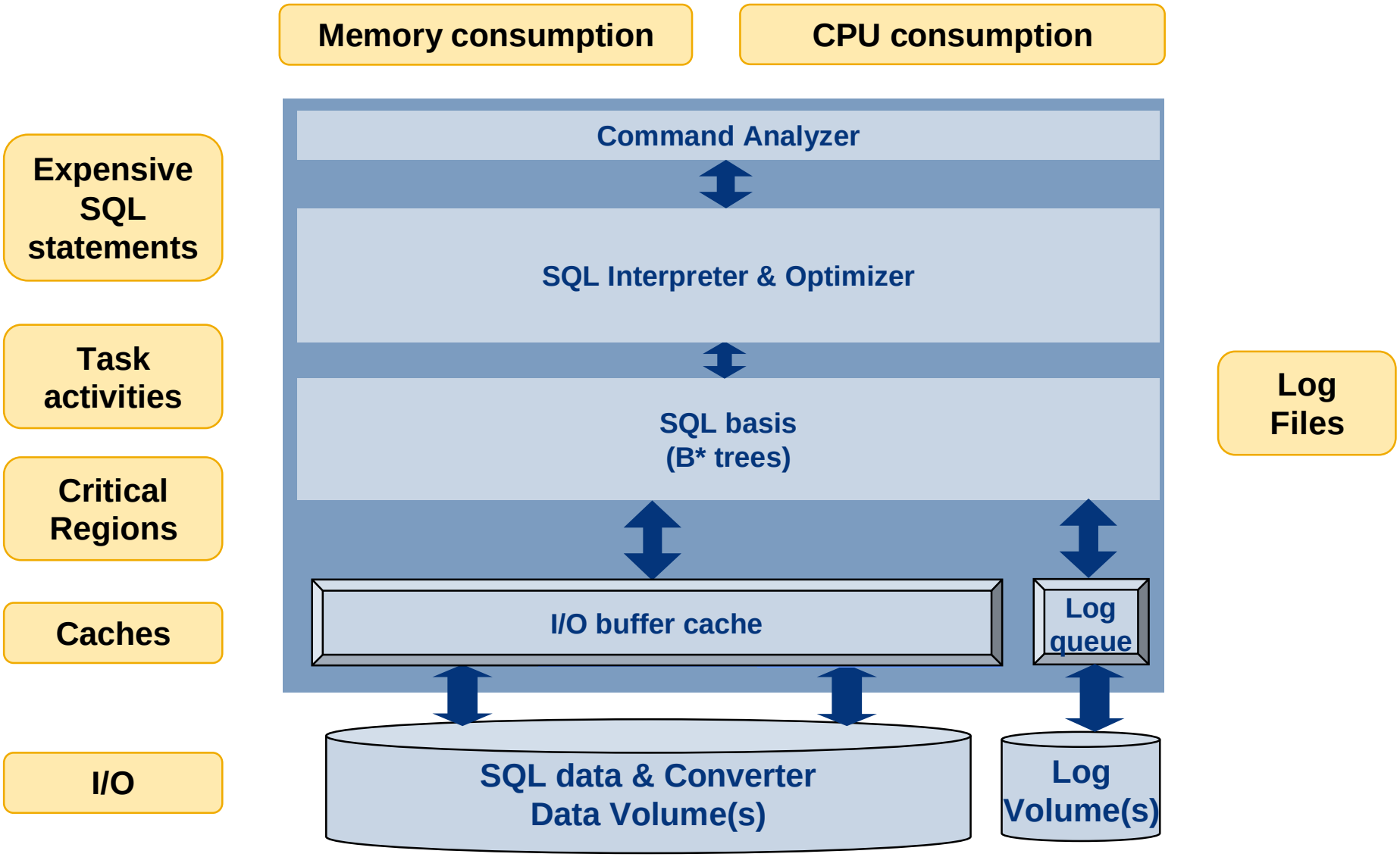
Central Monitoring



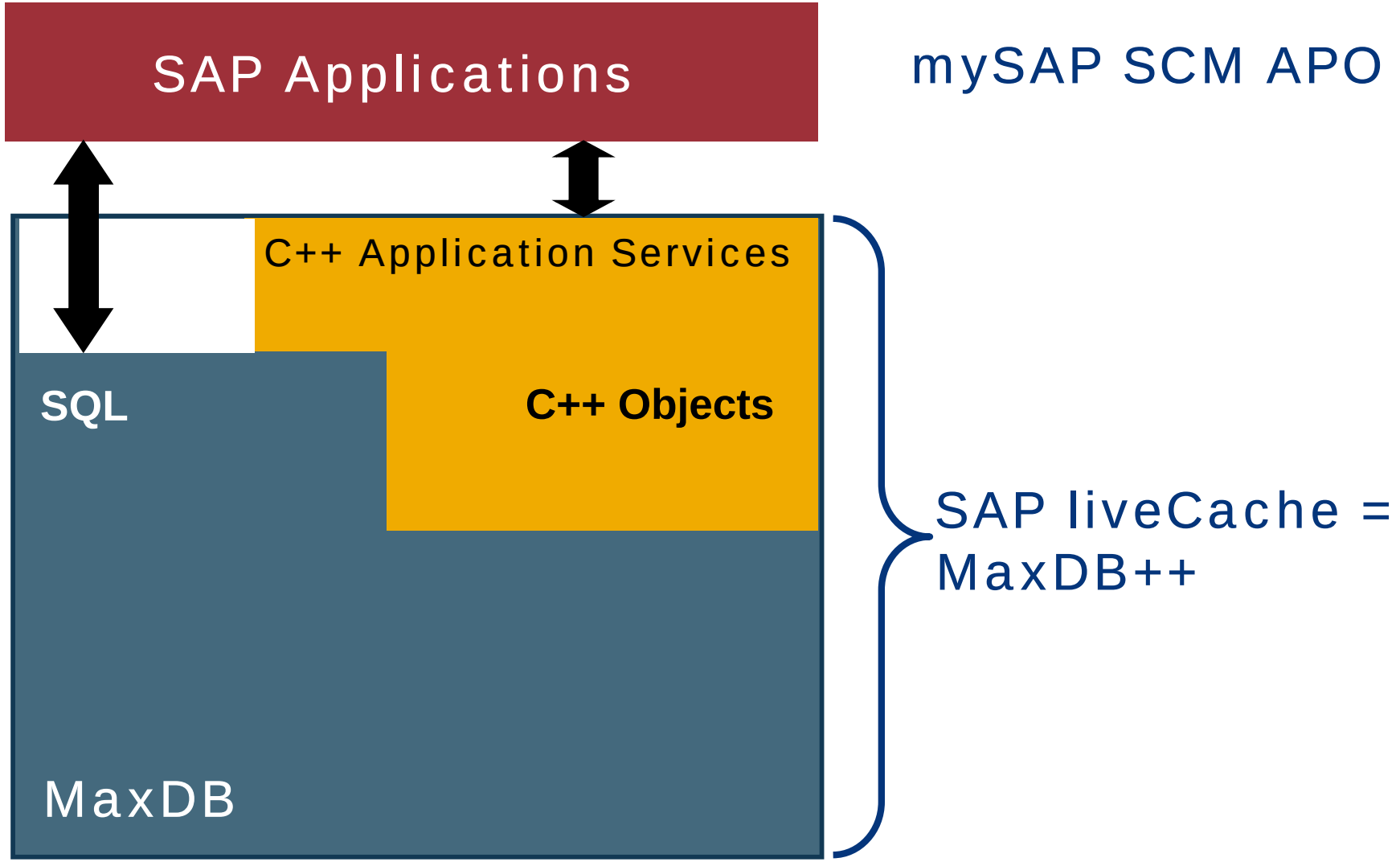
Anatomy of a MaxDB Instance



MaxDB Monitoring



SAP liveCache Technology



Transaction Availability

	DB50	DB59 / DB50N	LC10	RZ20	DB13C
4.6C	**GA**	**Basis SP 44**	**GA**	**Basis SP 44**	**NA**
4.6D	**GA**	**Basis SP 33**	**GA**	**Basis SP 33**	**NA**
6.10	**GA**	**Basis SP 32**	**GA**	**Basis SP 32**	**Basis SP 28**
6.20	**GA**	**Basis SP 21**	**GA**	**Basis SP 21**	**Basis SP 18**
6.40	**GA**	**GA**	**GA**	**GA**	**GA**
7.00	**GA**	**GA**	**GA**	**GA**	**NA – new: DBACOCKPIT**

GA: **general available; NA: not available**

DB50 and DB50N **work exclusively with SAP DB/MaxDB instances (as of version 7.3).**

DB59 and DB13C **are database independent.**

LC10 **works exclusively with liveCache instances. To start/stop/initialize the liveCache this transaction should only be used in the corresponding APO/SCM system - but liveCache instances can be monitored in any SAP system using the mentioned Basis SPs.**

RZ20 **is database independent and available as of SAP release 4.6C. The integration of any SAP DB/MaxDB and liveCache instances is possible as of the mentioned SPs.**

Target

- The following slides can be used as a reference book –
  they contain screenshots of the used transactions and additional information.

Chapter

What Has to be Monitored?

Standby Database

Central Monitoring – Transaction DB59

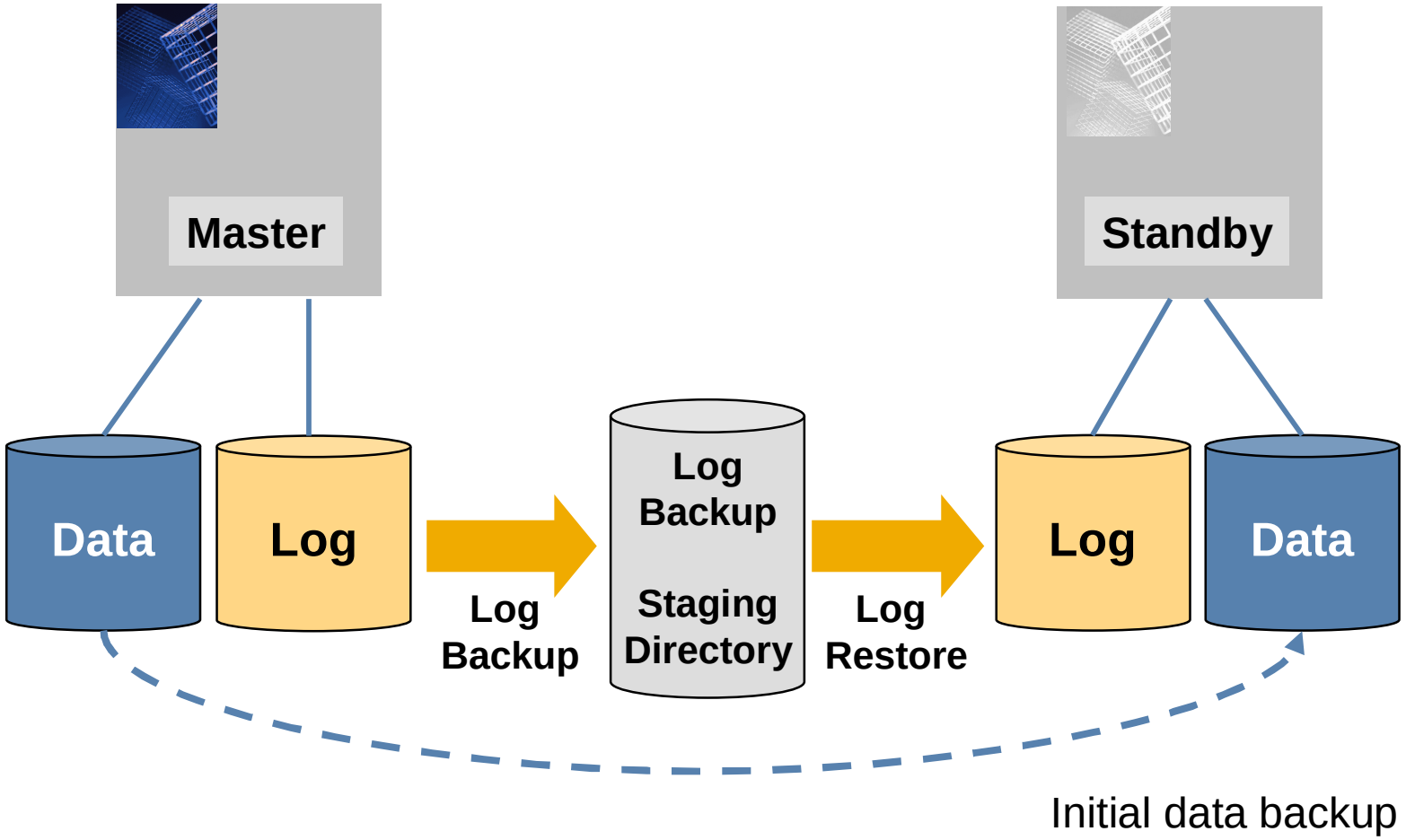
Database Assistant – Transaction DB50

Performance Analysis Tools

Alert Monitor – Transaction RZ20

Snapshot

Standby Database (w/ Log Shipping)



Standby Database: Slave Steps

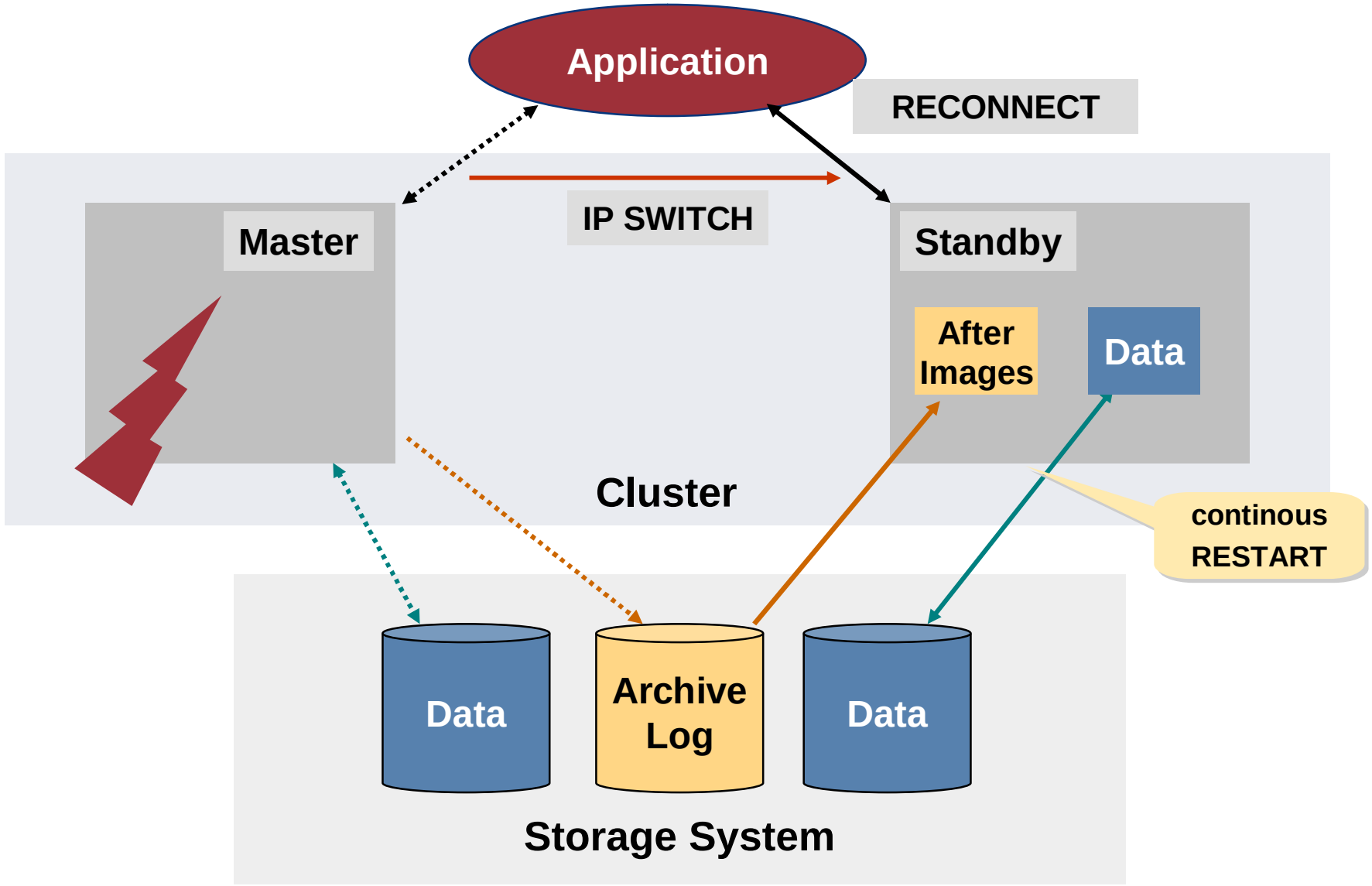
Standby Mode

- Initialize once with complete backup from Master
- Redo logs as they appear

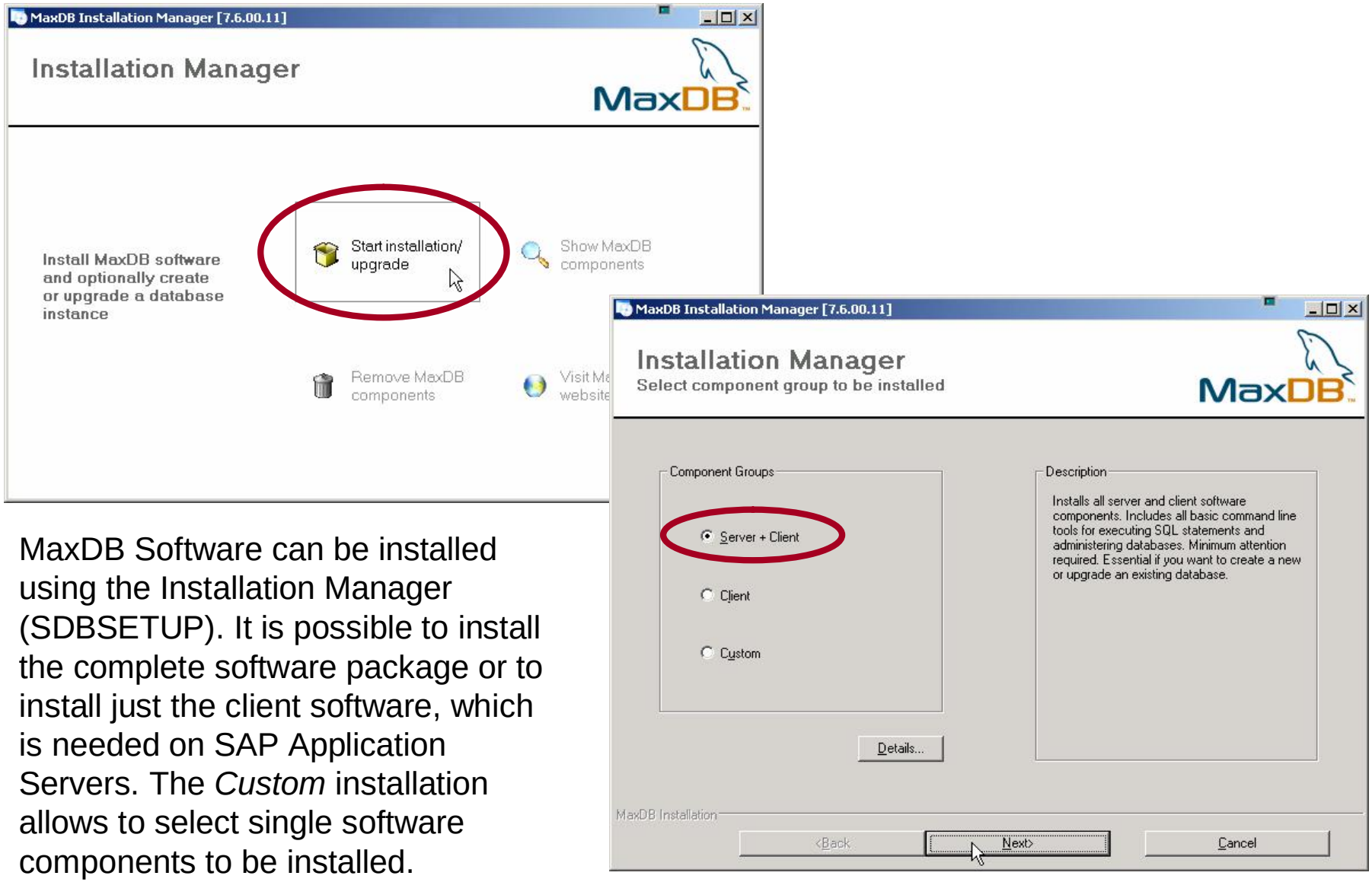
Start Slave to online mode in case of emergency

- If possible, back up last piece of log from master
- Redo all 'open' log backups (there should be none)
- Redo final log piece
- Restart slave to be the new master

Hot Standby – Standby Within Seconds

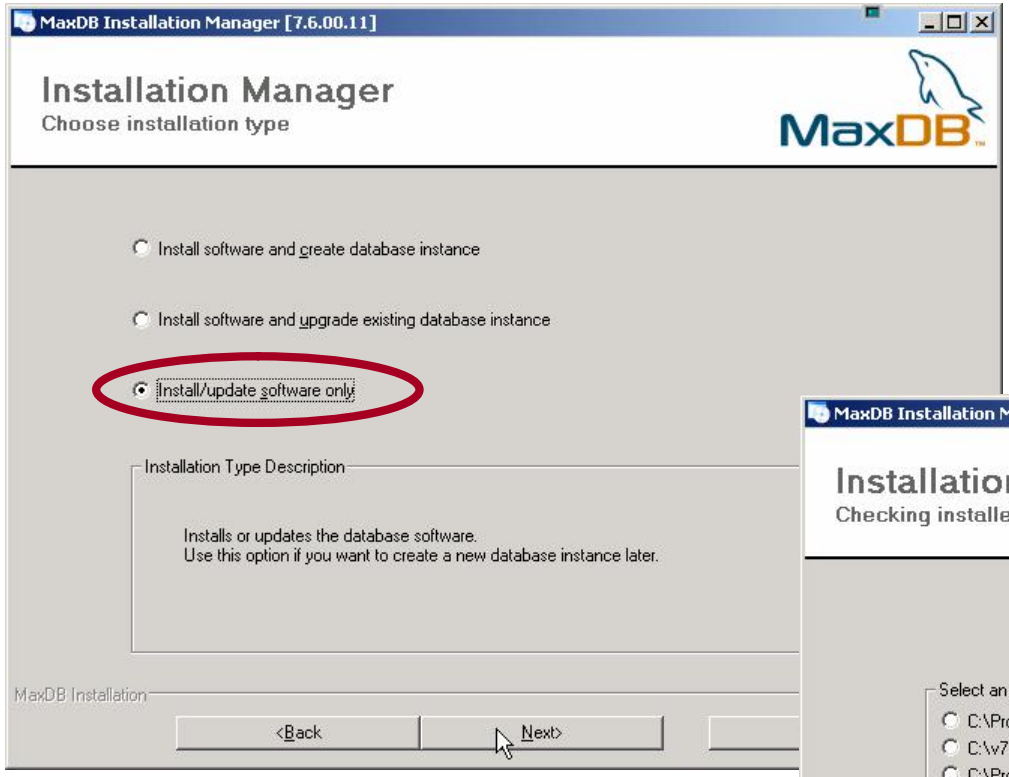


Standby Database: Installing Database Software I



MaxDB Software can be installed using the Installation Manager (SDBSETUP). It is possible to install the complete software package or to install just the client software, which is needed on SAP Application Servers. The *Custom* installation allows to select single software components to be installed.

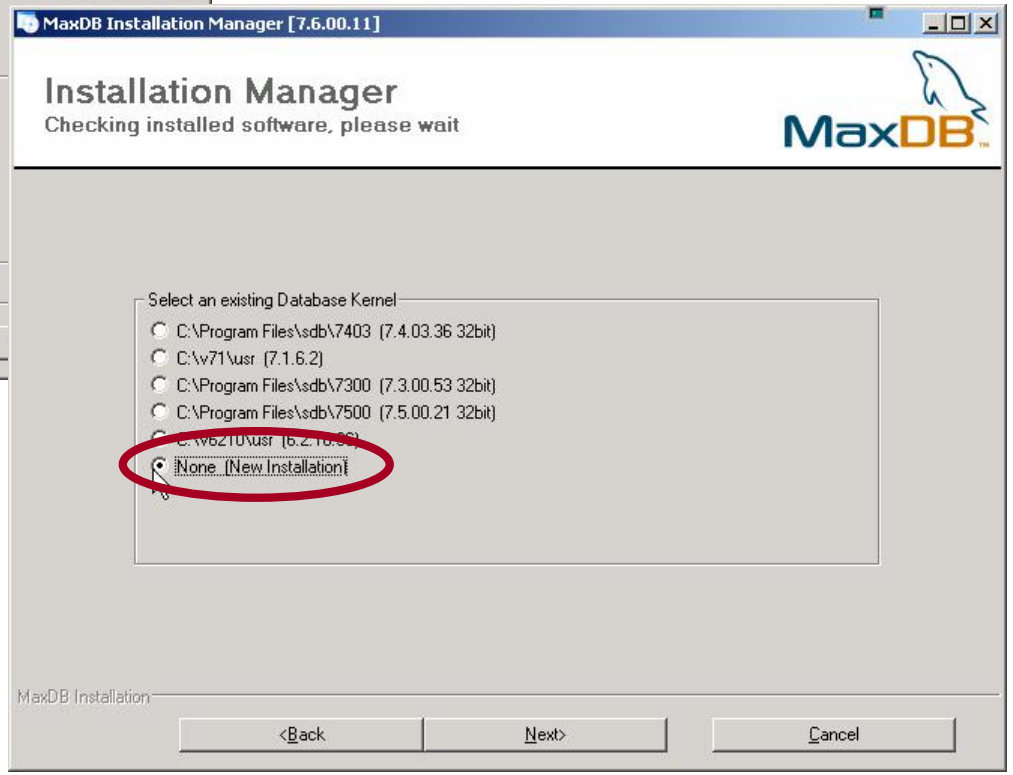
Standby Database: Installing Database Software II



You can choose if you would like to

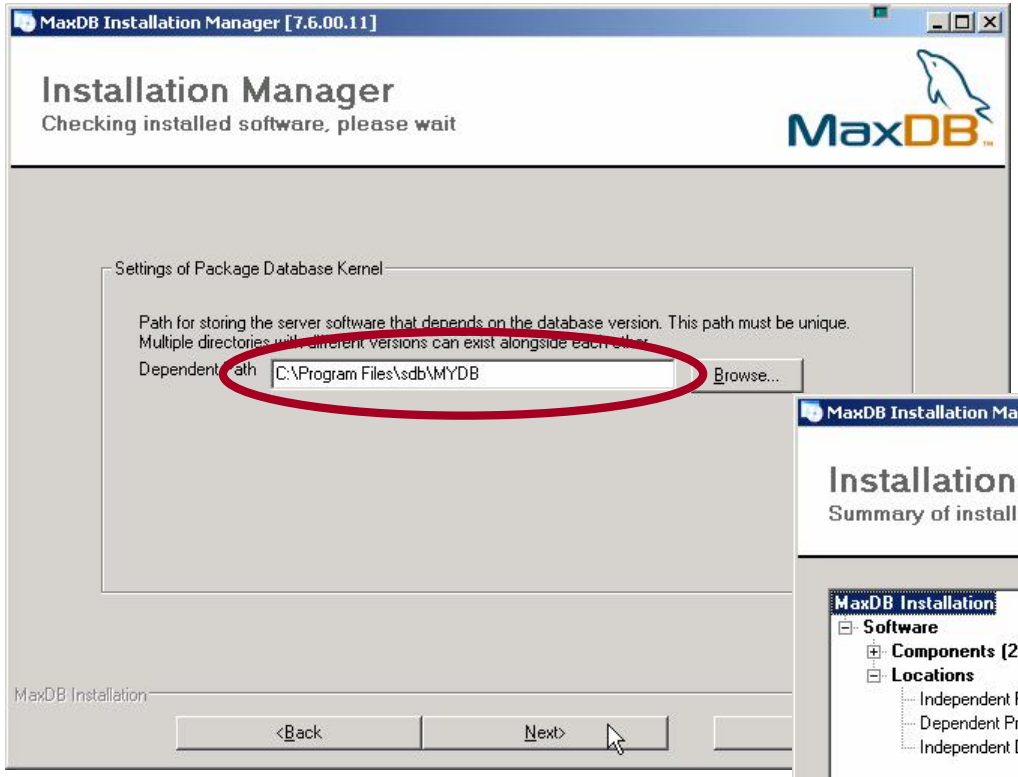
- install new software or
- update existing software

It is possible to install a new or upgrade an existing database instance.



You have to choose if you would like to upgrade an existing software installation of if you would like to install the software into a new directory.

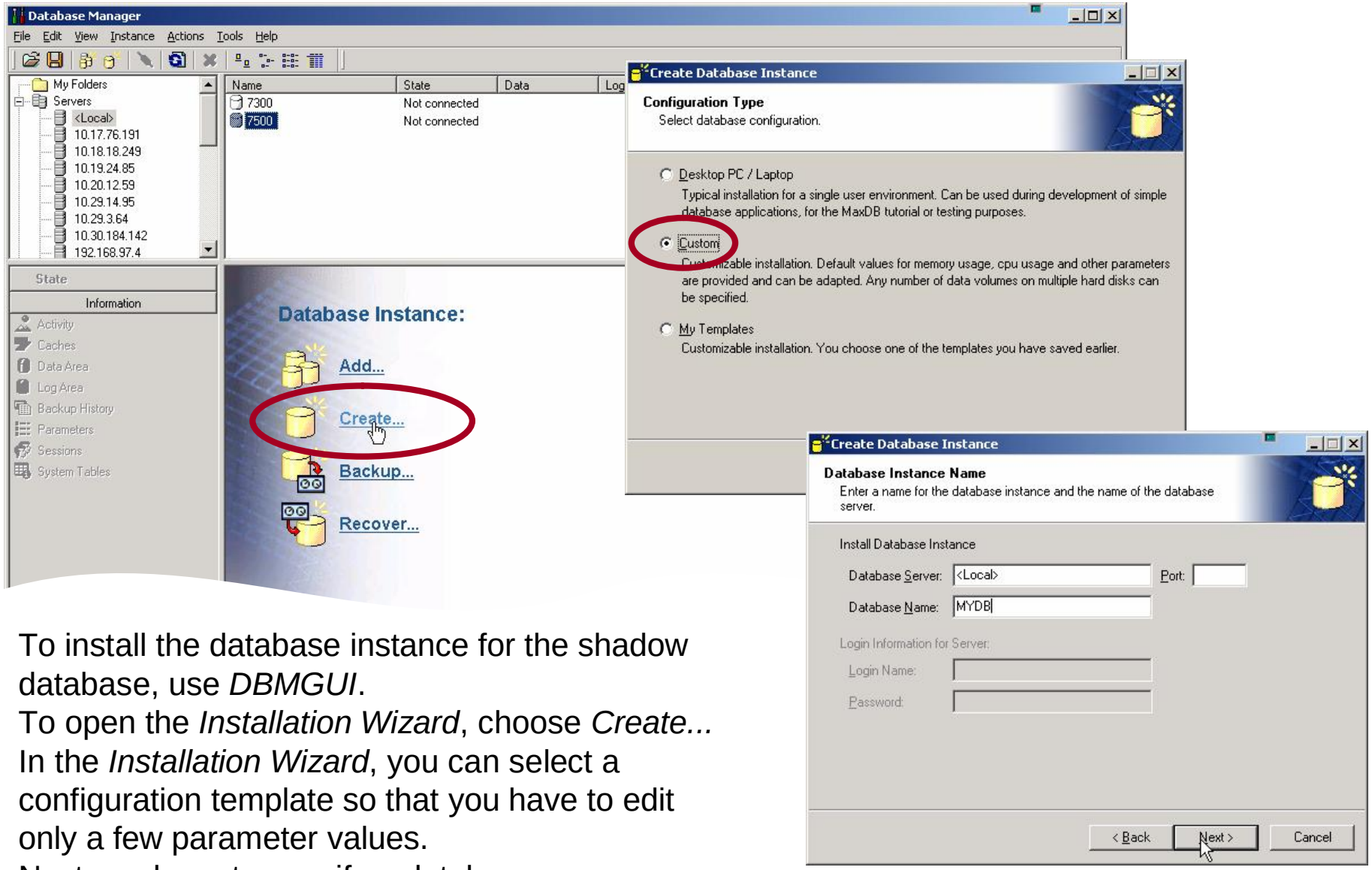
Standby Database: Installing Database Software III



If you would like to install the software into a new directory, you have to enter this directory.

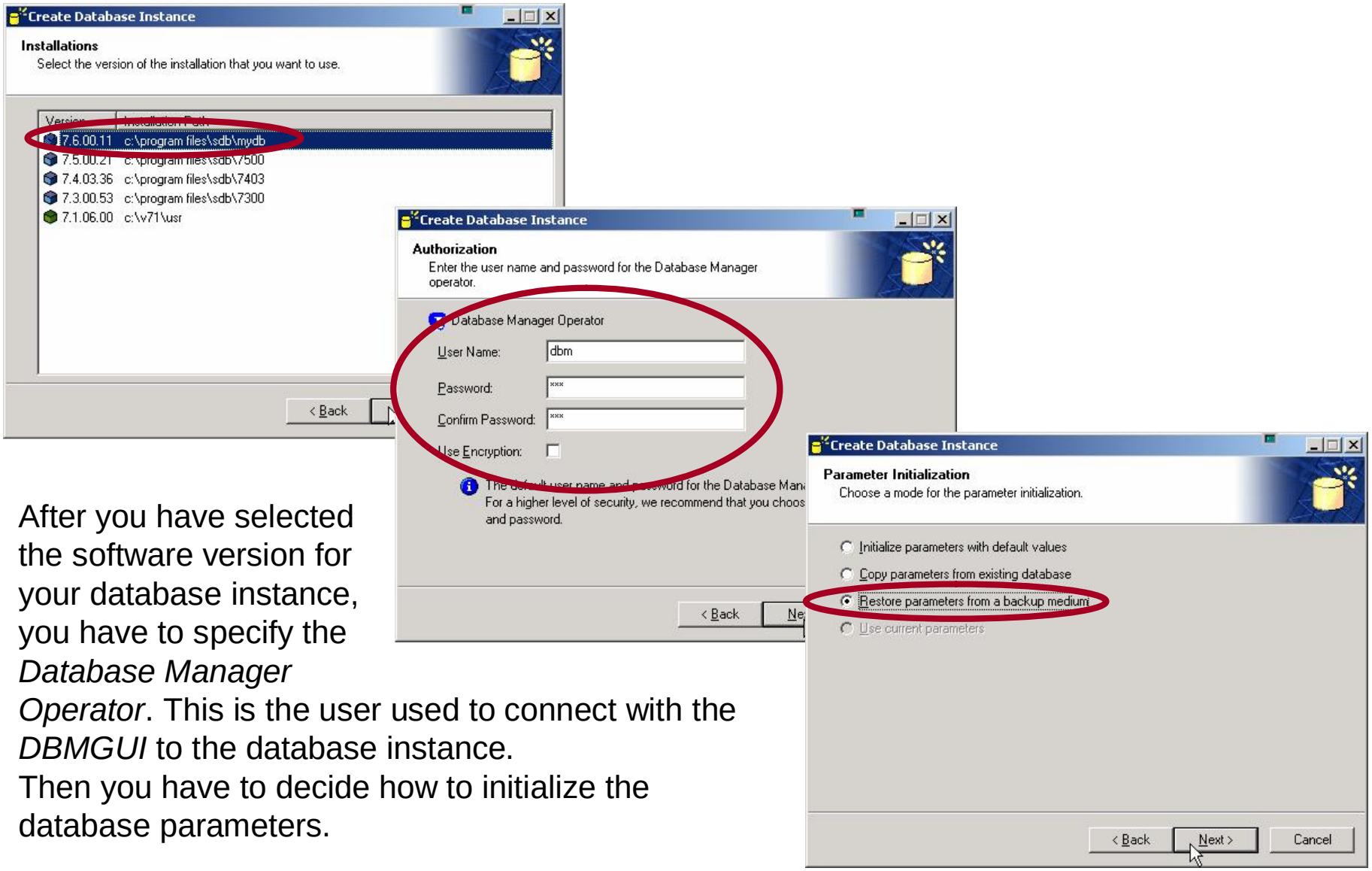
Then you'll get an overview of the selected/entered information and you can start the installation.

Standby Database: Installing Database Instance I



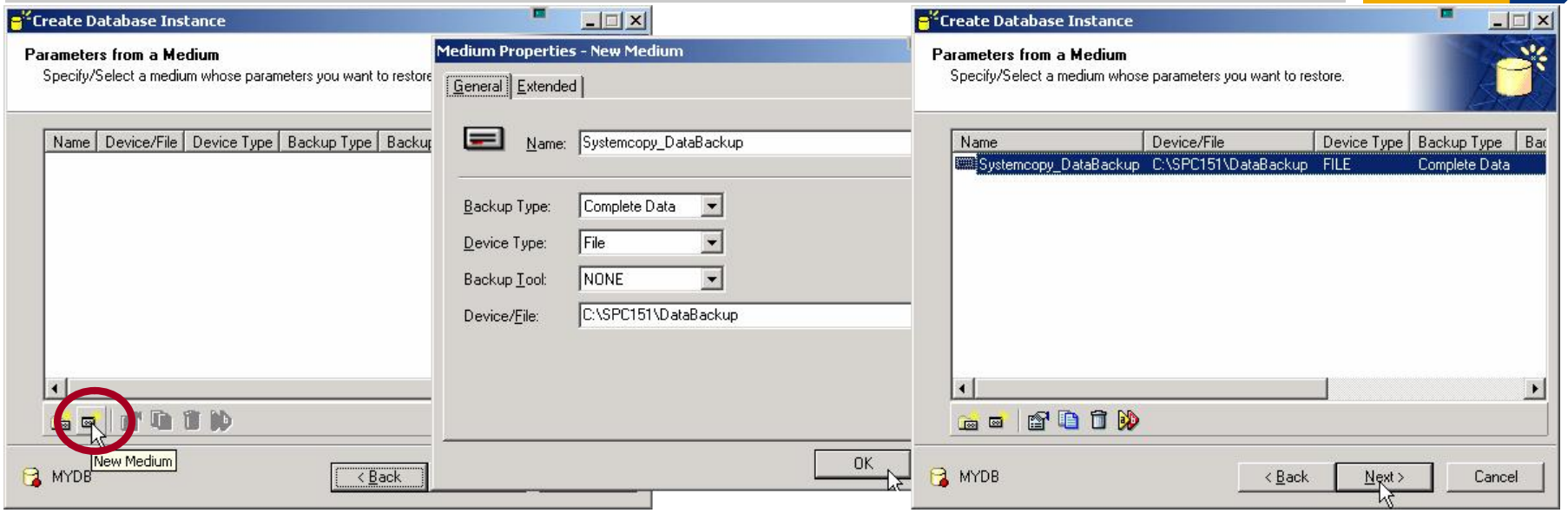
To install the database instance for the shadow database, use *DBMGUI*.
To open the *Installation Wizard*, choose *Create...*
In the *Installation Wizard*, you can select a configuration template so that you have to edit only a few parameter values.
Next you have to specify a database name.

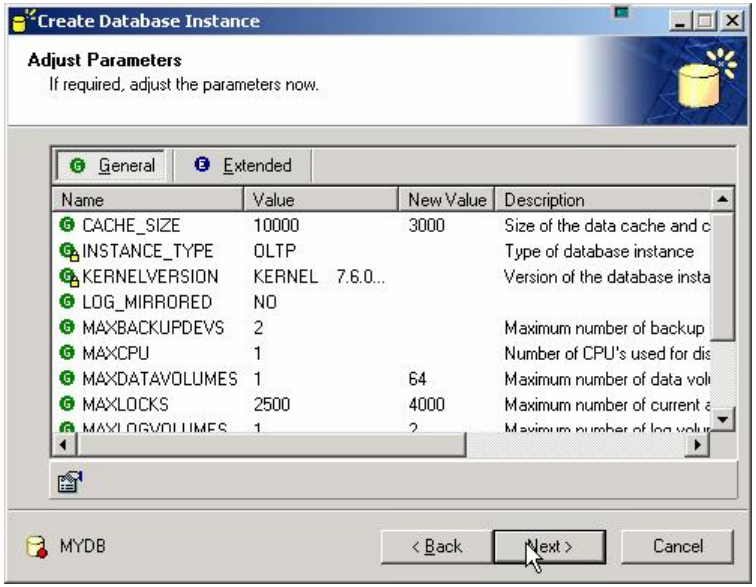
Standby Database: Installing Database Instance II



After you have selected the software version for your database instance, you have to specify the *Database Manager Operator*. This is the user used to connect with the *DBMGUI* to the database instance.
Then you have to decide how to initialize the database parameters.

Standby Database: Installing Database Instance III

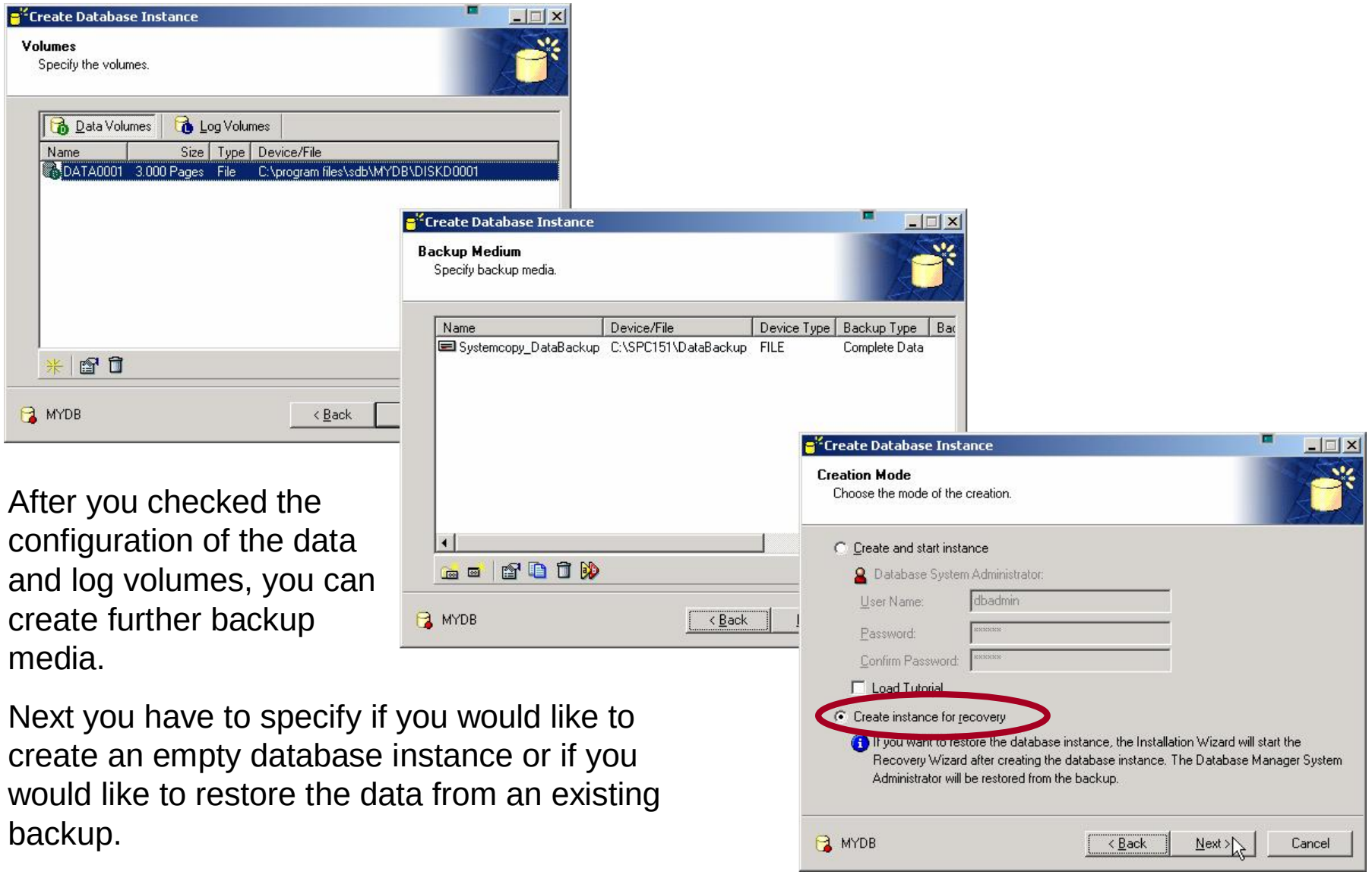




If you would like to get the parameter values from a backup, you have to specify the backup medium for that.

Afterwards you get a list with the parameter values that you can adapt if necessary.

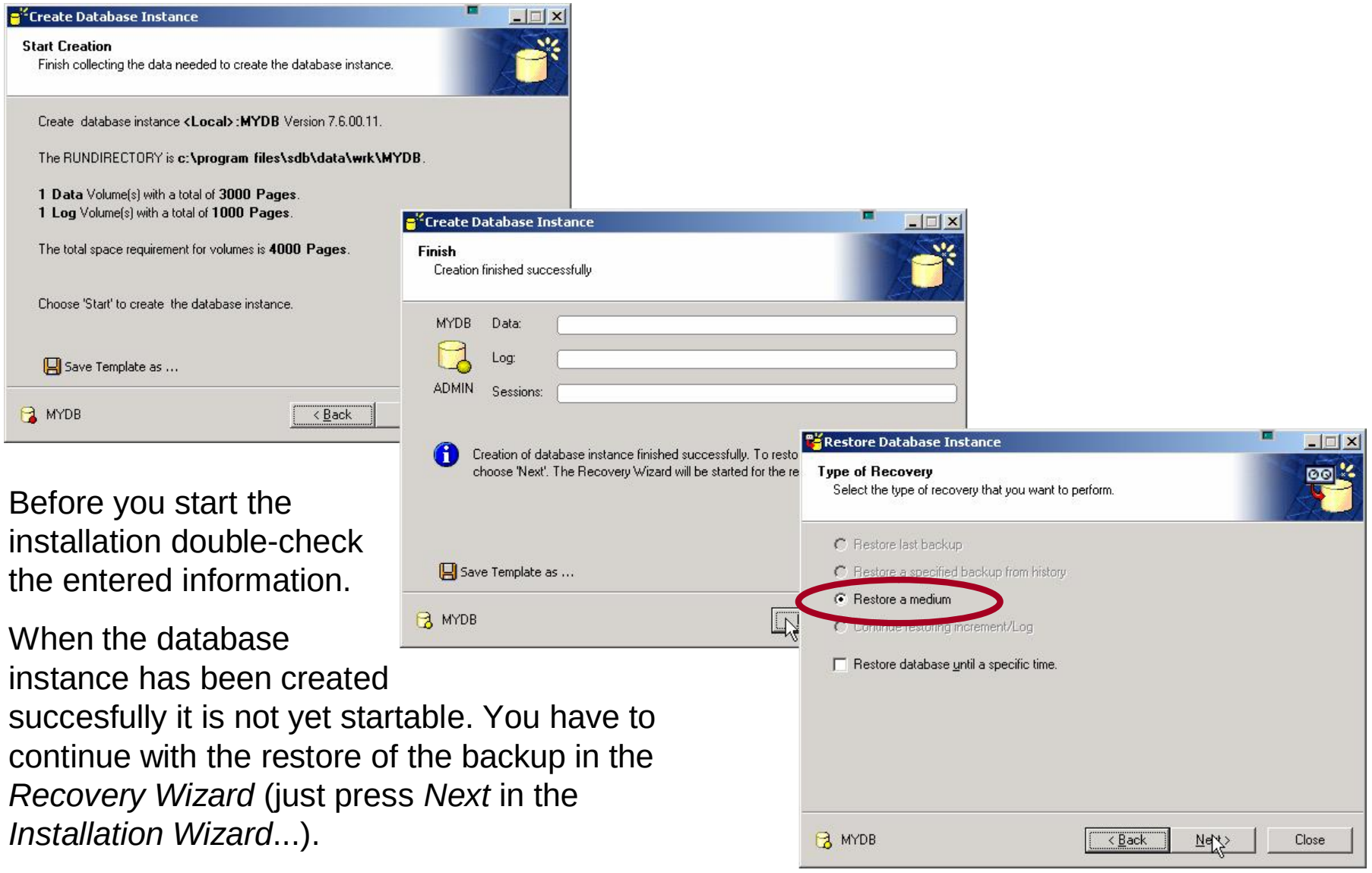
Standby Database: Installing Database Instance IV



After you checked the configuration of the data and log volumes, you can create further backup media.

Next you have to specify if you would like to create an empty database instance or if you would like to restore the data from an existing backup.

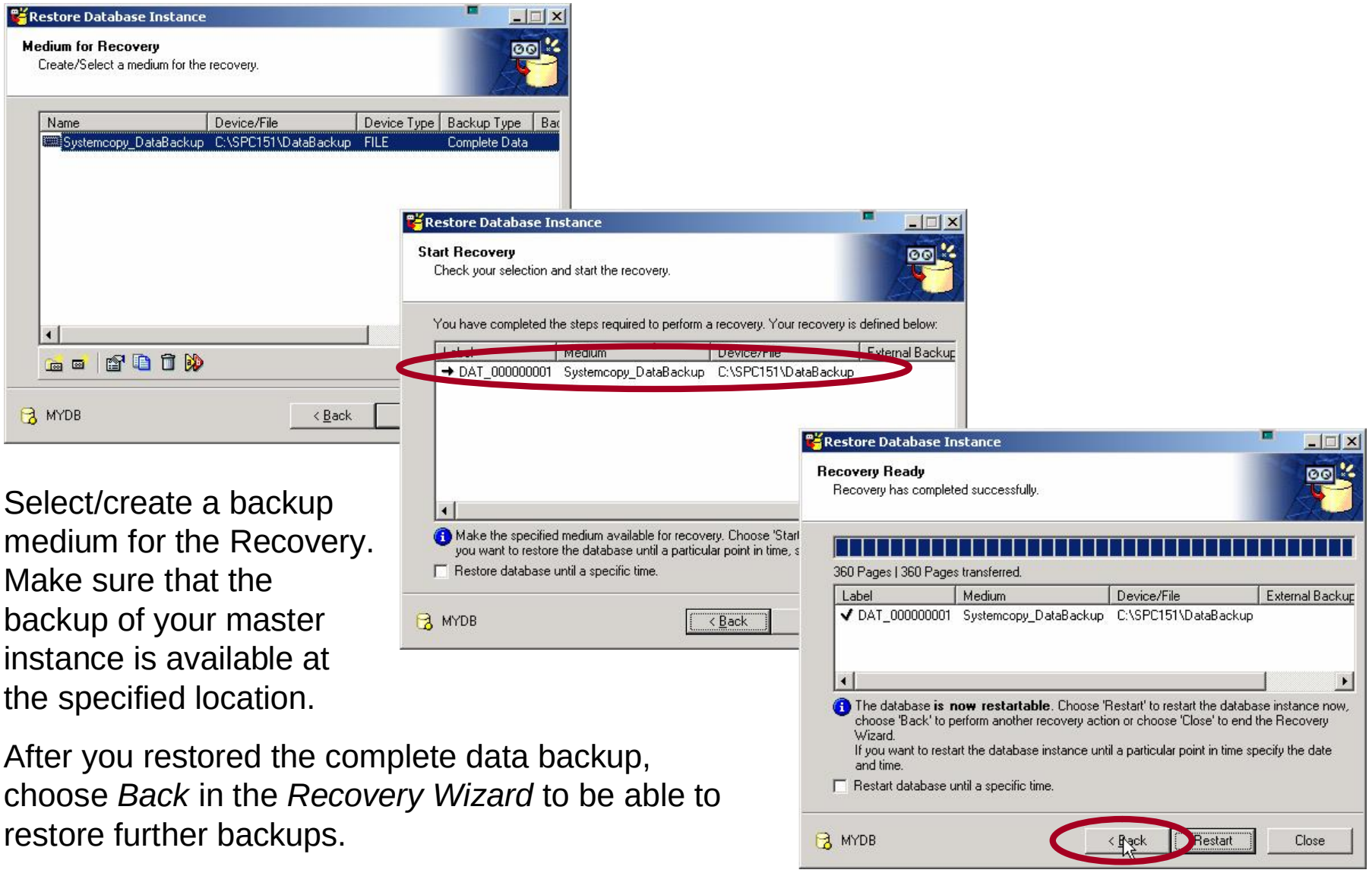
Standby Database: Installing Database Instance V



Before you start the installation double-check the entered information.

When the database instance has been created succesfully it is not yet startable. You have to continue with the restore of the backup in the *Recovery Wizard* (just press *Next* in the *Installation Wizard*...).

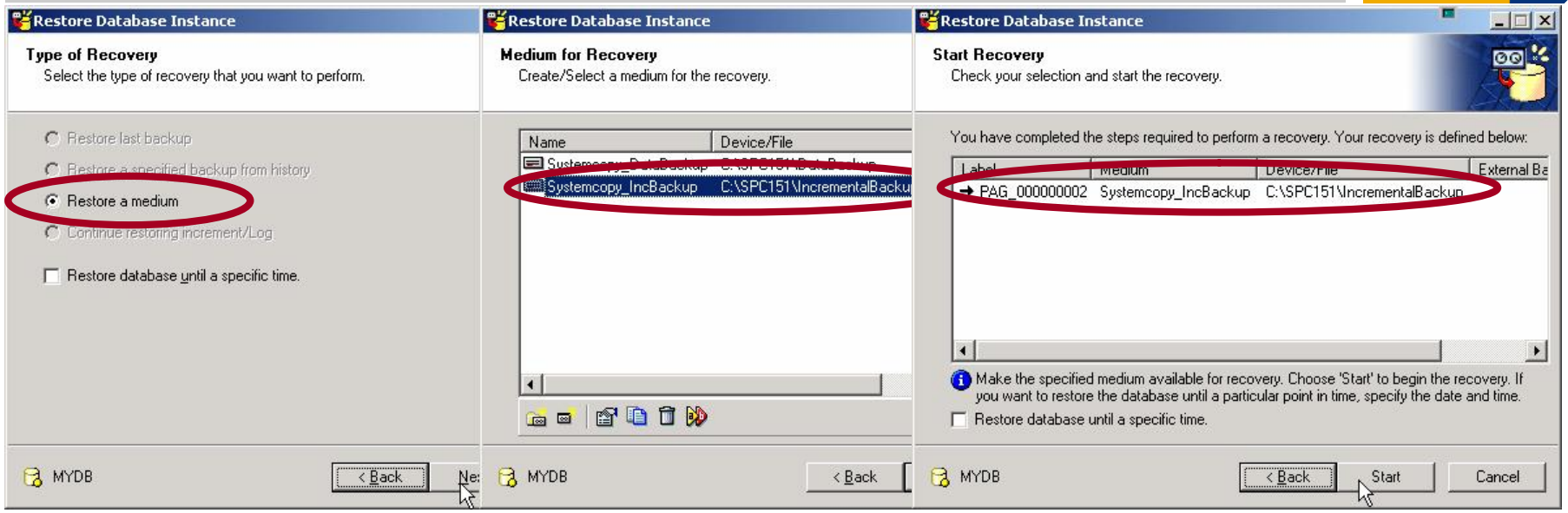
Standby Database: Restoring the Initial Backup

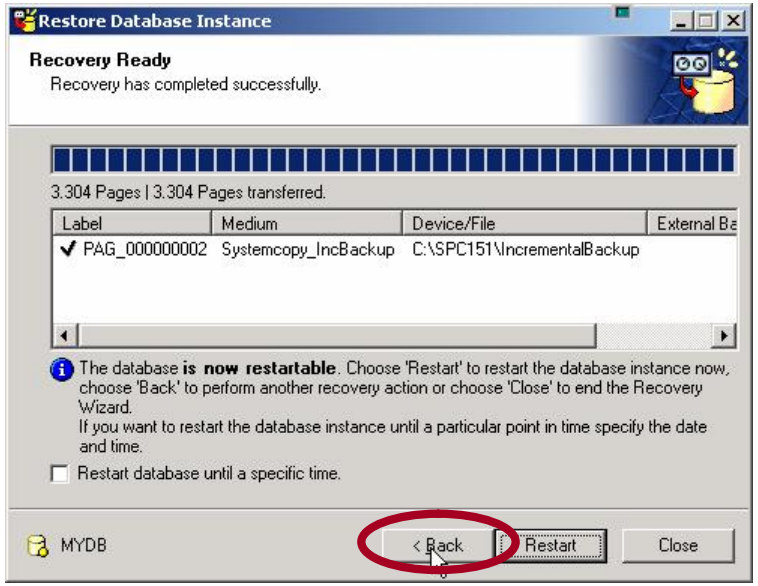


Select/create a backup medium for the Recovery. Make sure that the backup of your master instance is available at the specified location.

After you restored the complete data backup, choose *Back* in the *Recovery Wizard* to be able to restore further backups.

Standby Database: Restoring an Incremental Backup



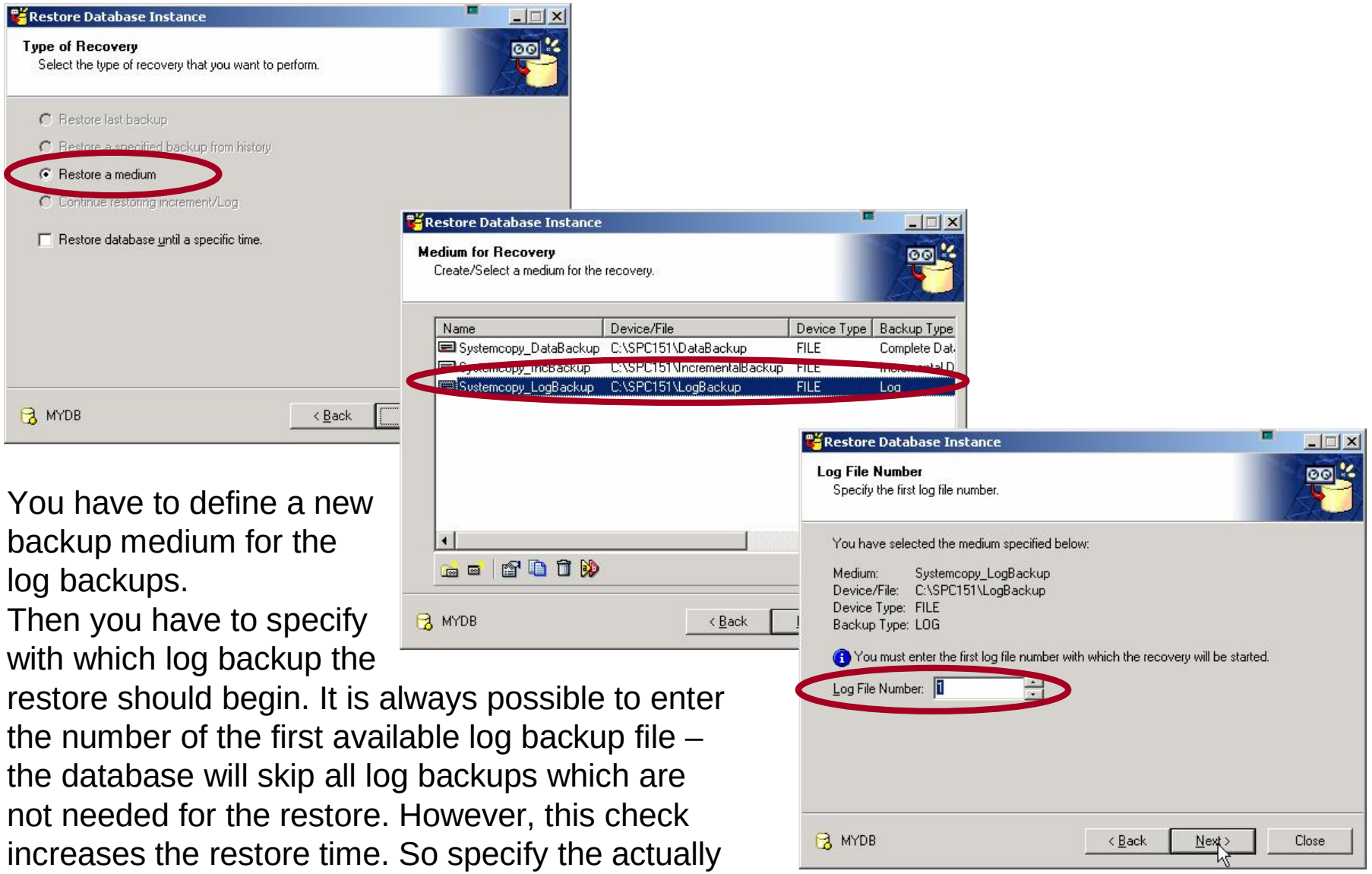


Depending on the available backups you can now continue with the restore of an incremental backup or with the restore of log backups.

You have to define a new backup medium for the incremental backup.

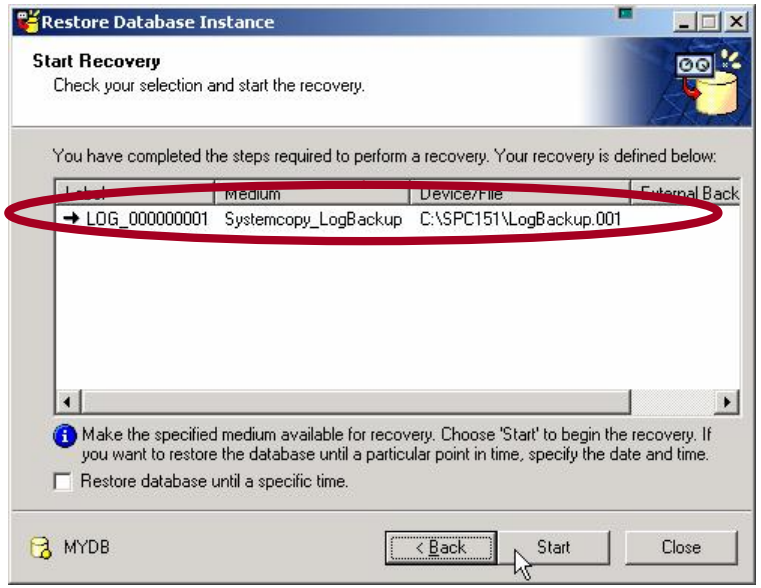
After you restored the incremental backup, choose *Back* to be able to restore further log backups.

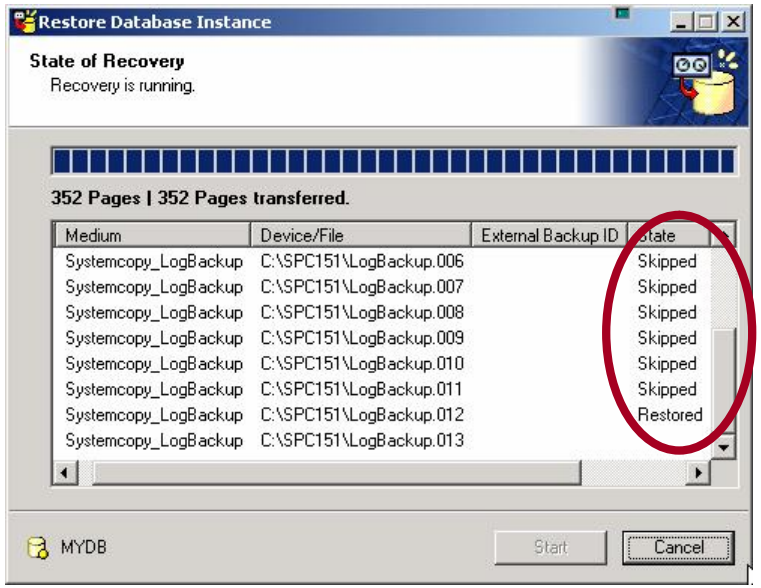
Standby Database: Restoring Log Backups I



You have to define a new backup medium for the log backups.
Then you have to specify with which log backup the restore should begin. It is always possible to enter the number of the first available log backup file – the database will skip all log backups which are not needed for the restore. However, this check increases the restore time. So specify the actually needed log backup number if you know it.

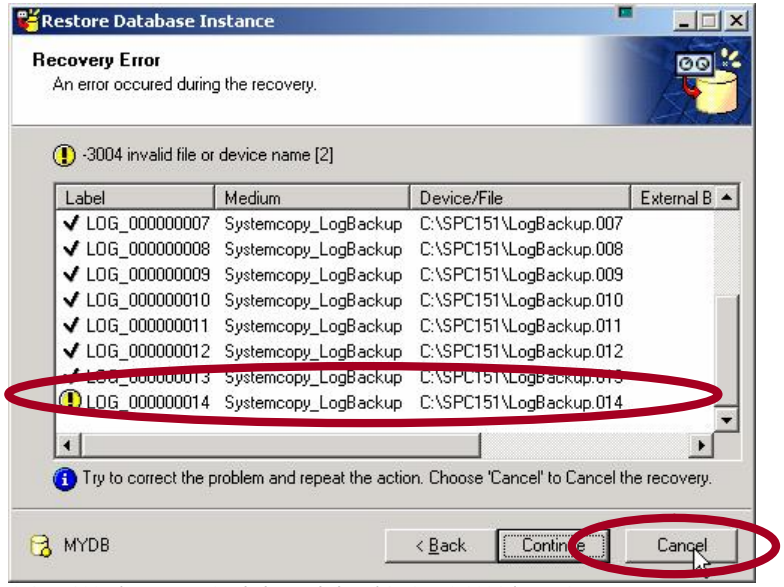
Standby Database: Restoring Log Backups II

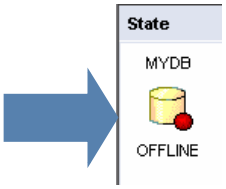




All available log backup files are restored one after another automatically. When the Database Manager tries to restore a log backup which is not available, an error is reported. Then you have to decide

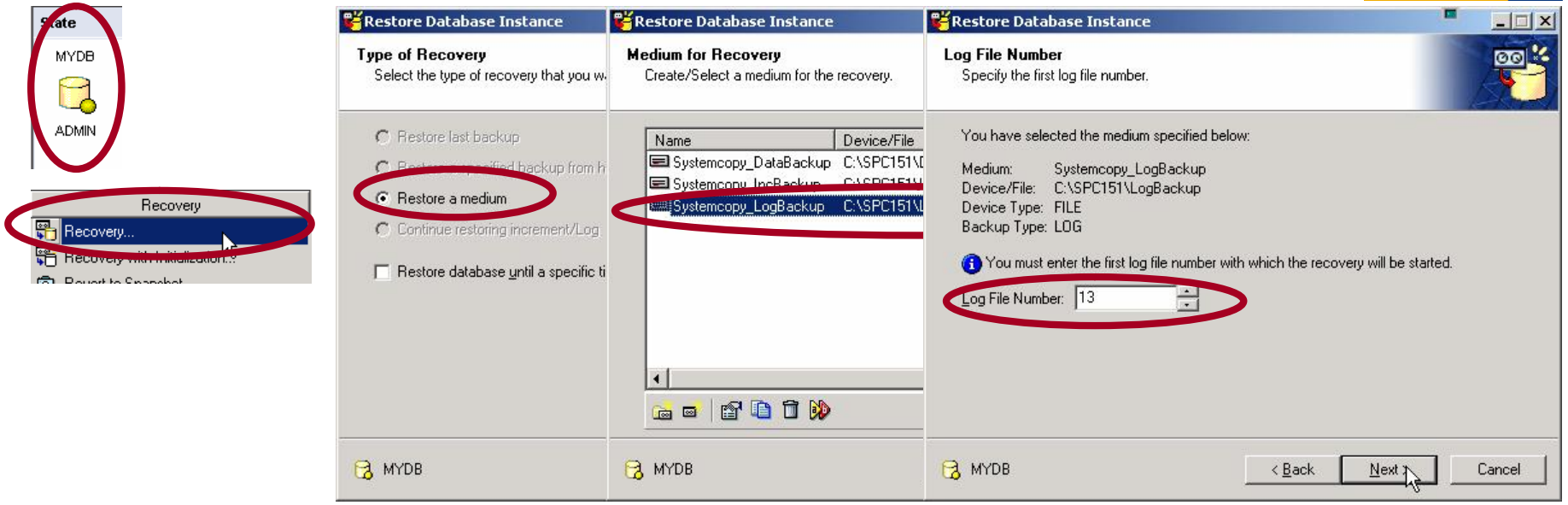
- if you have restored all needed data,
- if you would like to continue the restore later or
- if you can make the needed log backup file available and continue then with the restore.

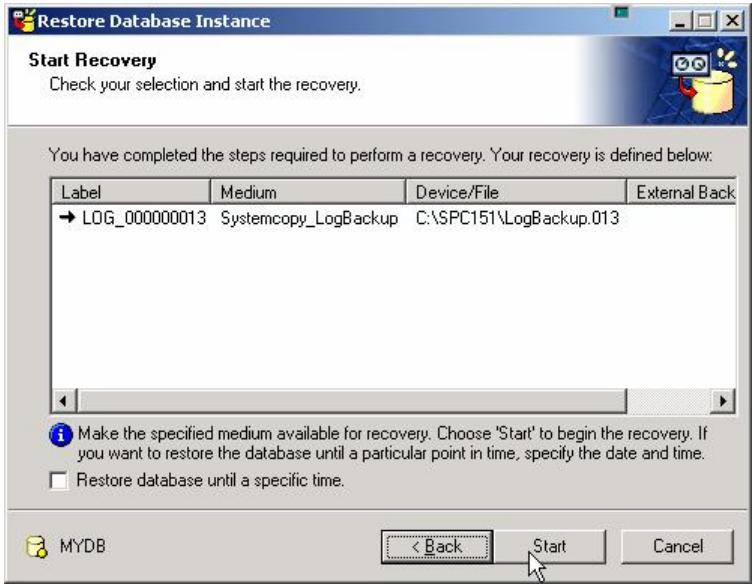




Attention: *Back* can be used to change the media definition or to start the database. *Continue* tries to restore the backup file again. *Cancel* aborts the restore and stops the database instance.

Standby Database: Restoring Log Backups III



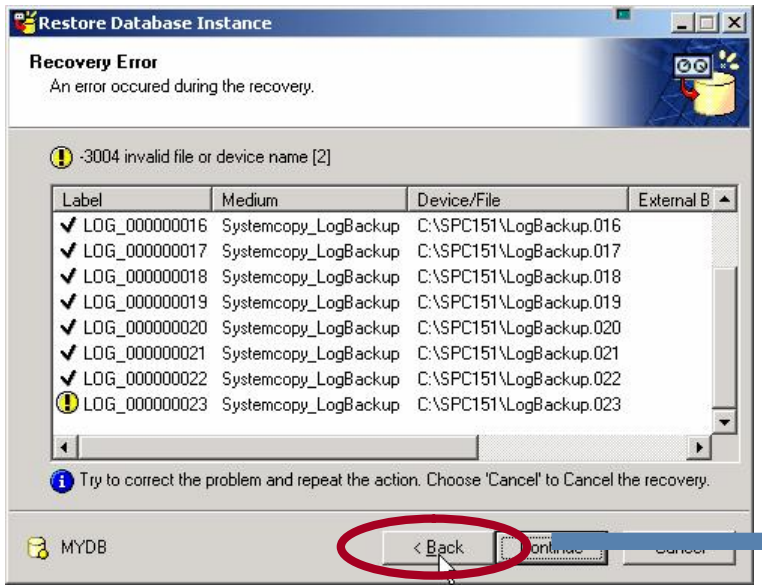


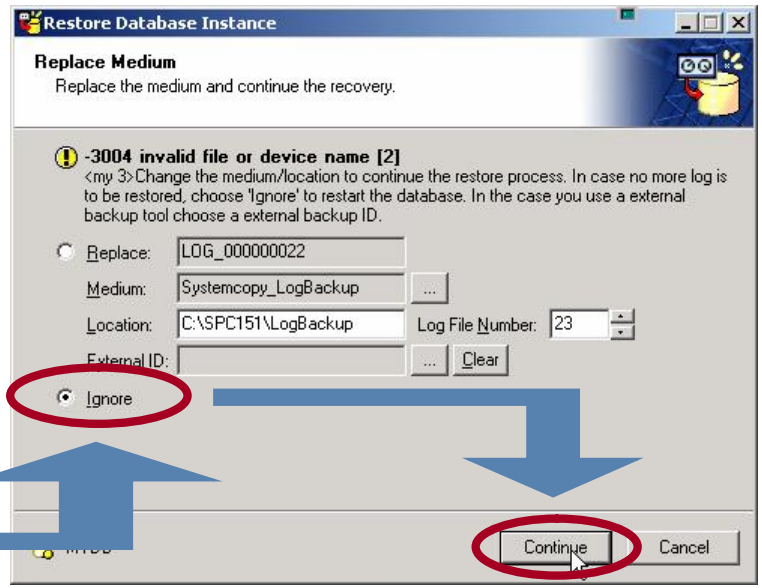
When you would like to continue with the restore, you have to start the database instance into ADMIN mode.

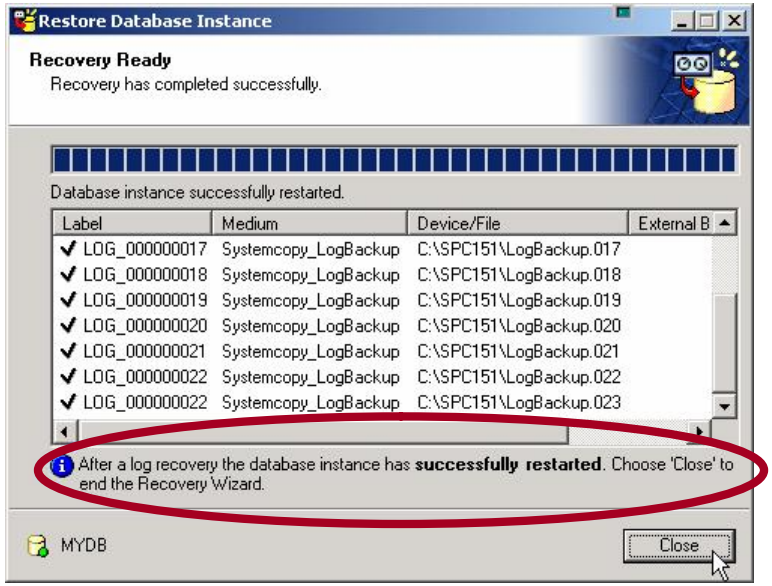
Attention: Do not start the database instance in ONLINE mode if you would like to continue the restore. When the database instance was ONLINE, you'll have to start with the initialization and restore of a complete backup again!

To continue the restore you have to specify the last log backup file which was already restored successfully.

Standby Database: Restoring Log Backups IV

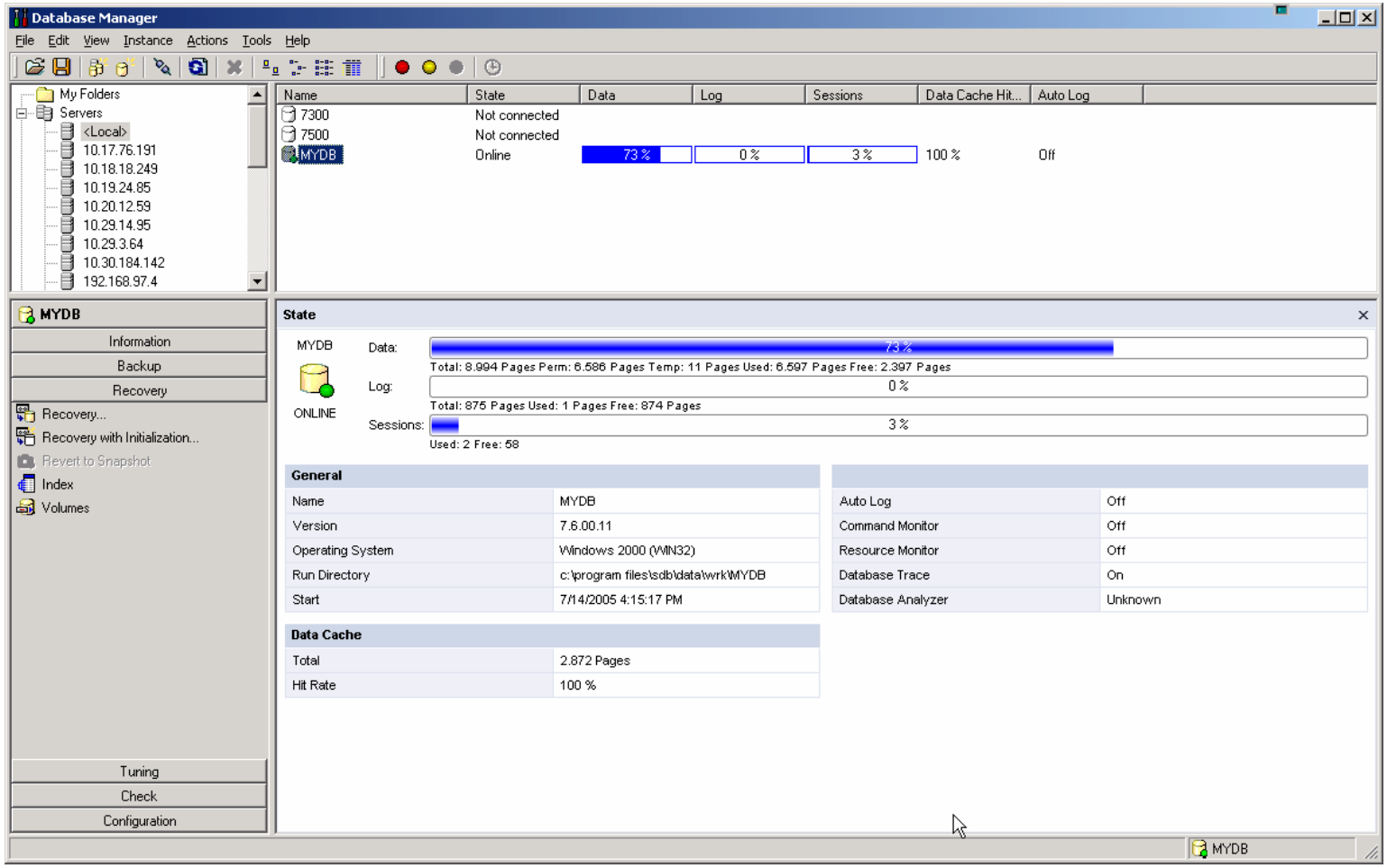






When all log backups have been restored successfully and you would like to start the standby instance, you have to choose *Back*. To start the database select *Ignore* and press *Continue*.

Standby Database: ONLINE



The standby instance is in ONLINE mode now. Please remember to load the system tables (*Configuration → Upgrade System Tables*). Then the database can be used as the production instance and it can be administered and monitored using DBMGUI or transaction DB50.

Chapter

What Has to be Monitored?

Standby Database

Central Monitoring – Transaction DB59

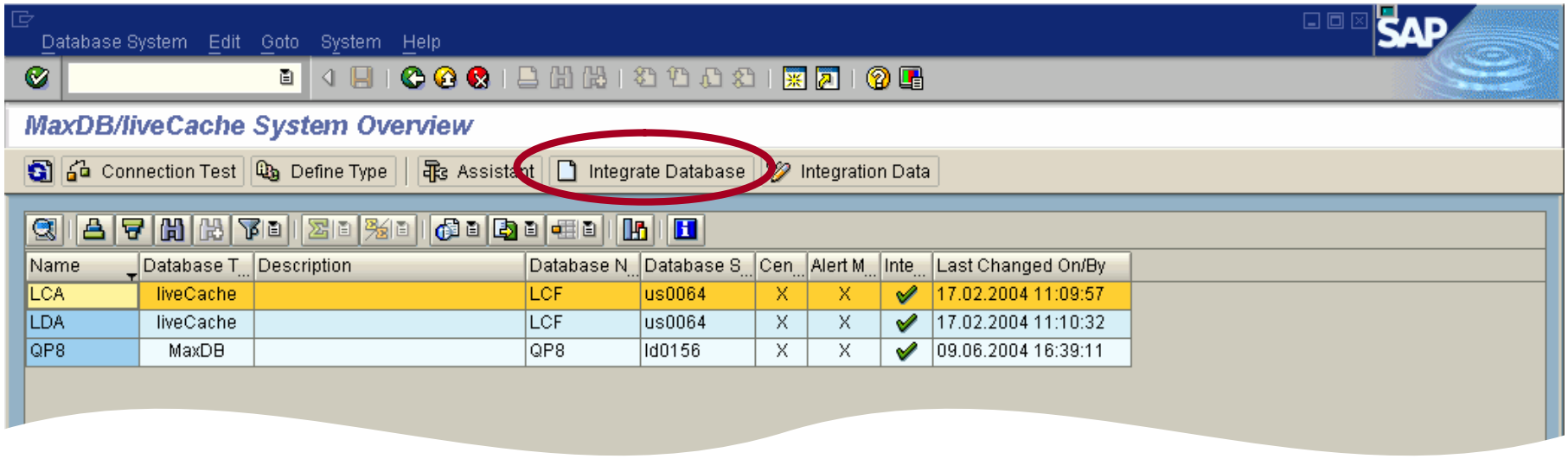
Database Assistant – Transaction DB50

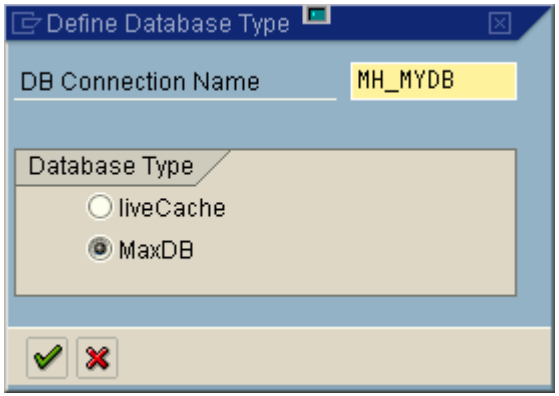
Performance Analysis Tools

Alert Monitor – Transaction RZ20

Snapshot

Central Monitoring – Transaction DB59

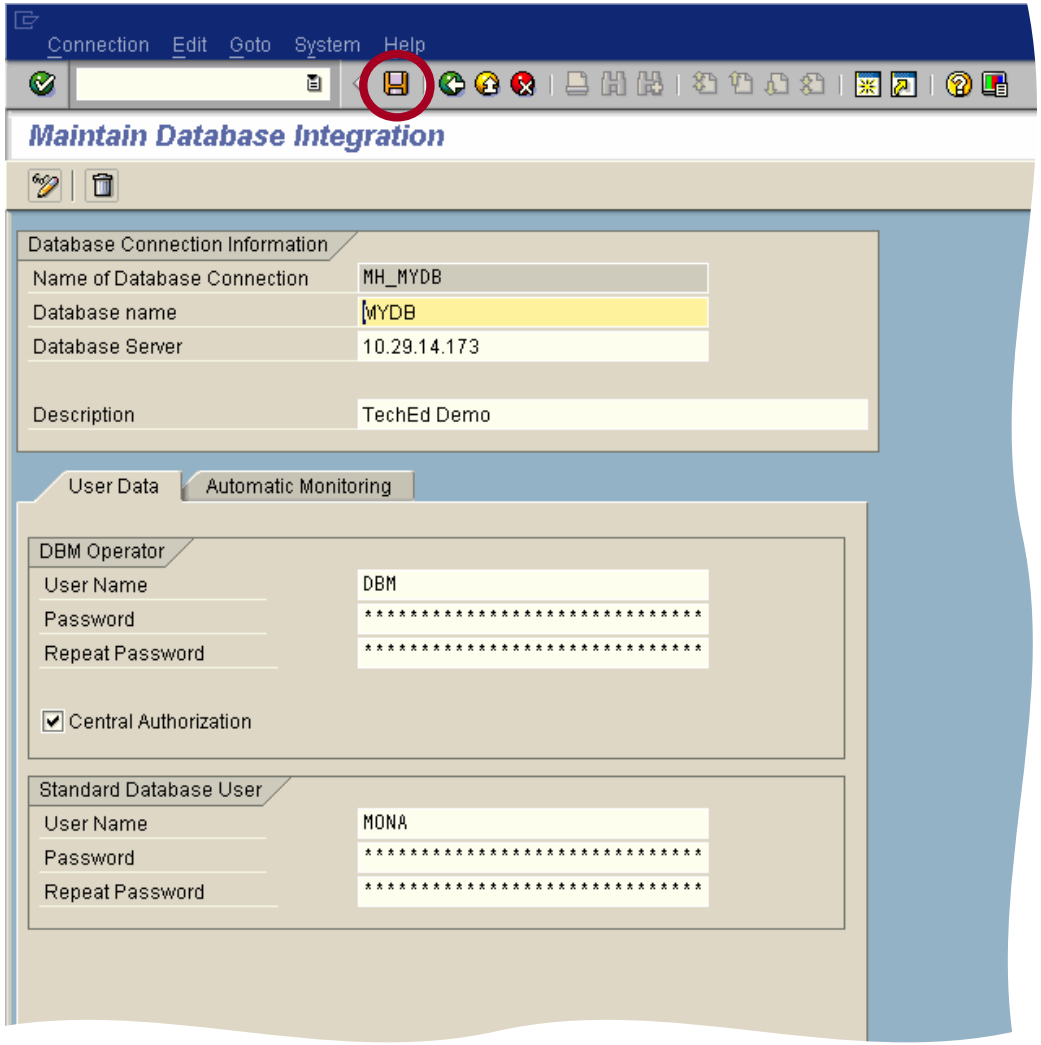




Initially transaction DB59 contains only an entry for the system's own MaxDB instance and - in case of an APO/SCM system - the two liveCache connection identifiers LCA and LDA. As of SCM 5.0 the connection identifier LEA is used as well.

To be able to monitor other MaxDB instances within this system, you have to integrate the corresponding database instance - i.e. you have to enter the user information to connect to this database instance.

Database Integration

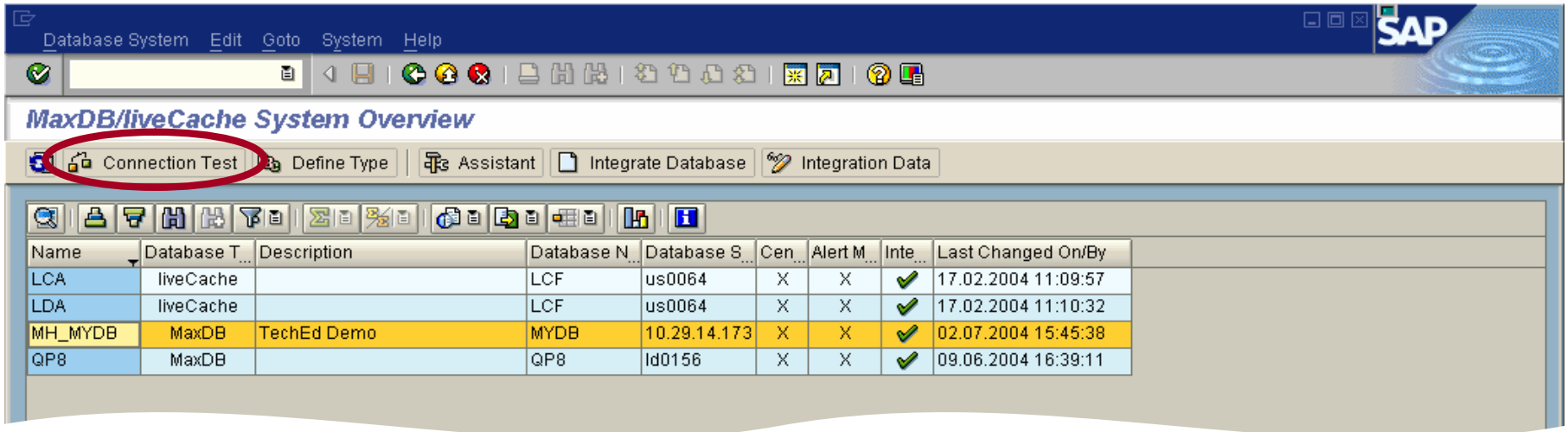


To be able to connect to the database instance you have to enter:

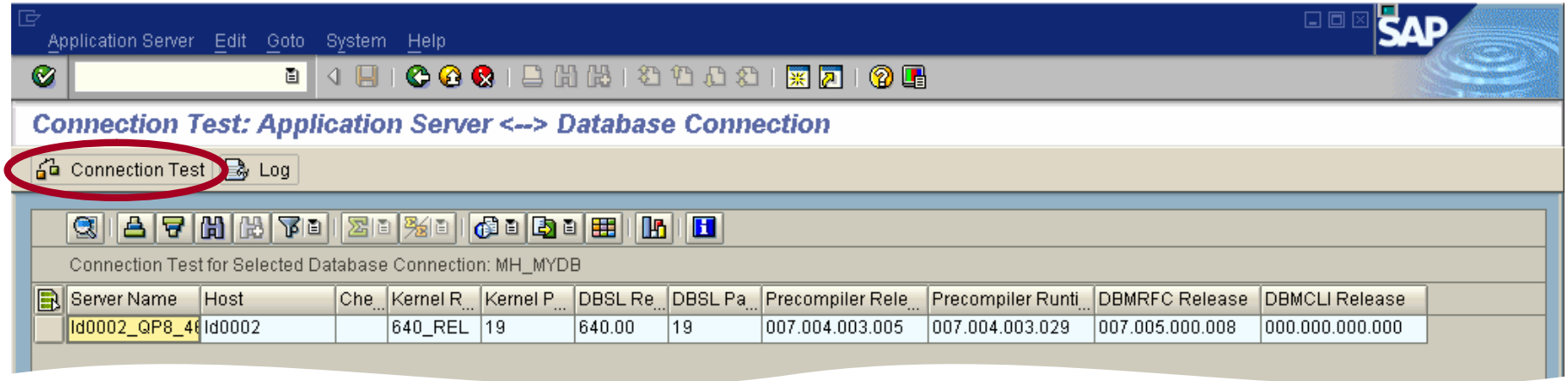
- The name of the database instance.
- The hostname or IP address of the server on which the database instance runs.
- The DBM operator and his password. This is the user used to connect to the Database Manager GUI or CLI. Default: control with password control.
- The Standard Database User, which for SAP applications is SAPR3 or SAP<SID> as the default.

  Make sure that the checkbox for the central authorization is marked, that the connect information is stored in tables DBCON and DBCONUSR of the monitoring system.

Connection Test

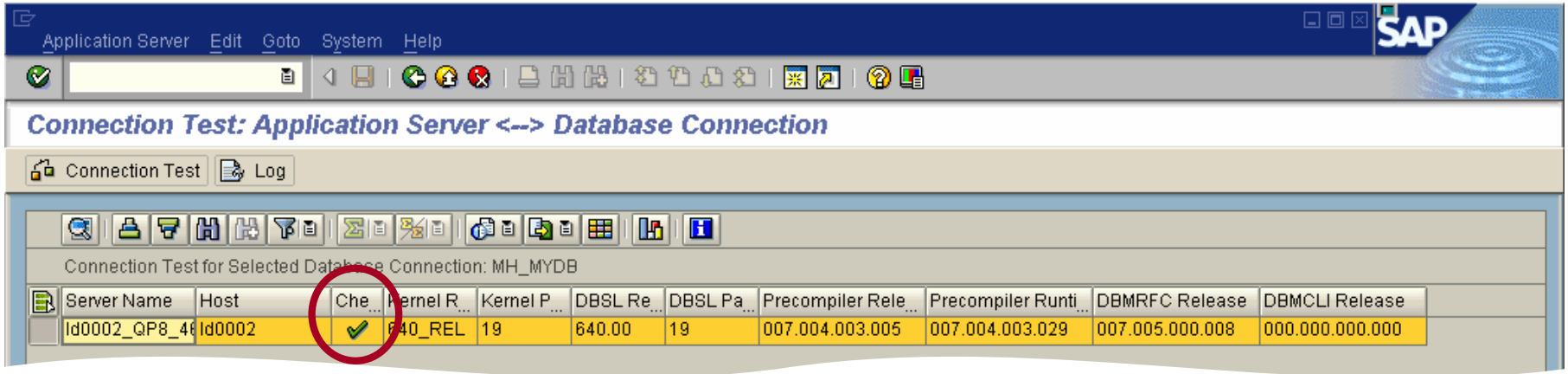


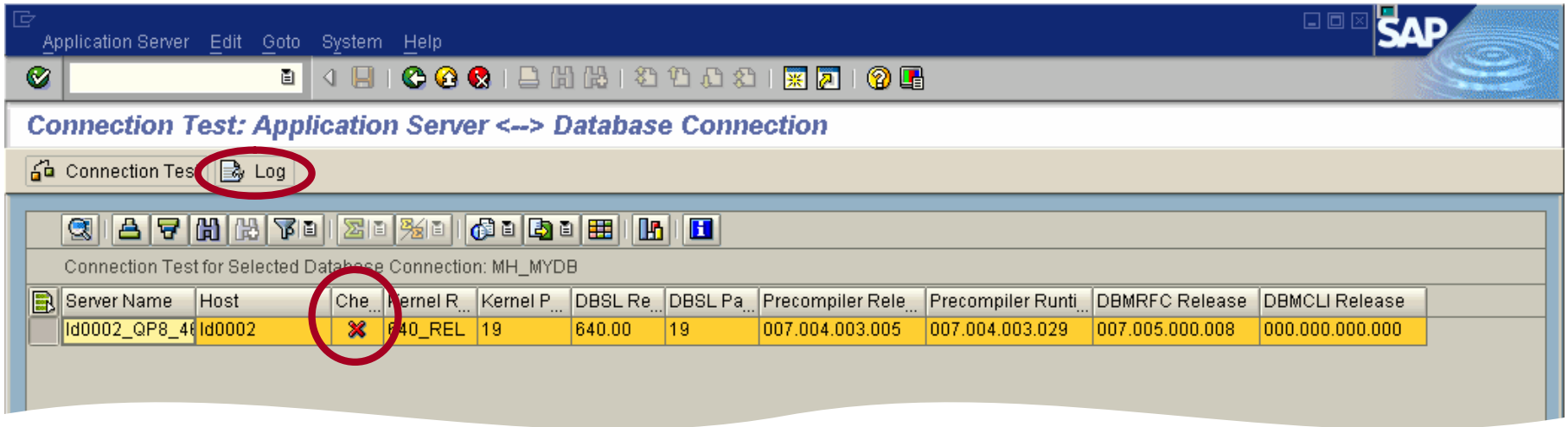
To check, if the entered connect information works, mark the entry of the concerning database instance and choose *Connection Test*.



You'll get a list of all application servers. Select one and choose *Connection Test* to check one after another if the connection works from all application servers.

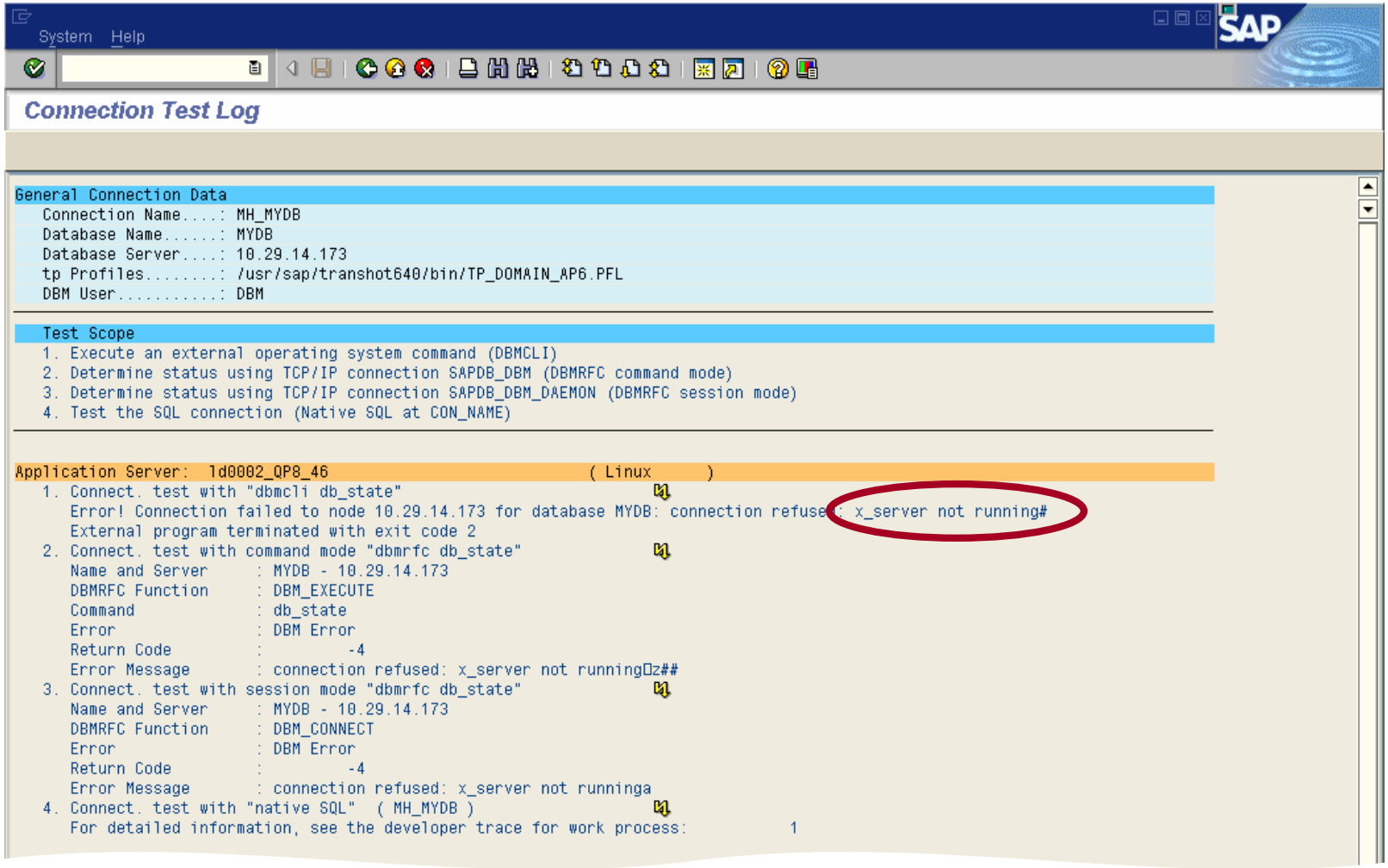
Connection Test





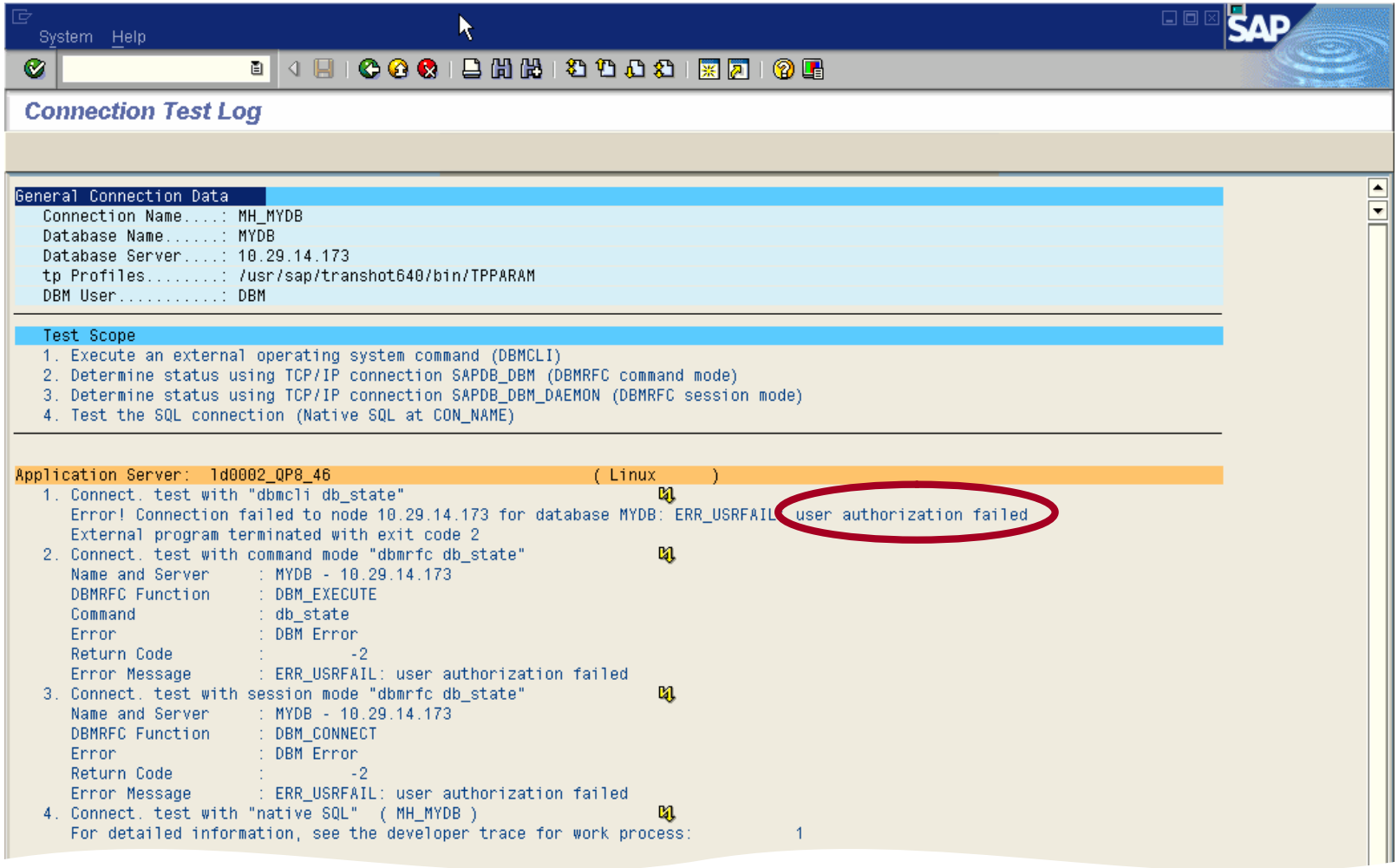
The connection test first checks the DBMCLI connection, then the command and session mode of DBMRFC and afterwards the SQL connection.
If everything is OK, a green check mark appears for this application server, otherwise a red cross is shown. In this case you have to check the log file.

X-Server Not Running



One possible problem is that the X-Server is not running on the database server. The X-Server is the TCP/IP listener of MaxDB which handles remote connections to the database instance.

Wrong DBM Operator Password

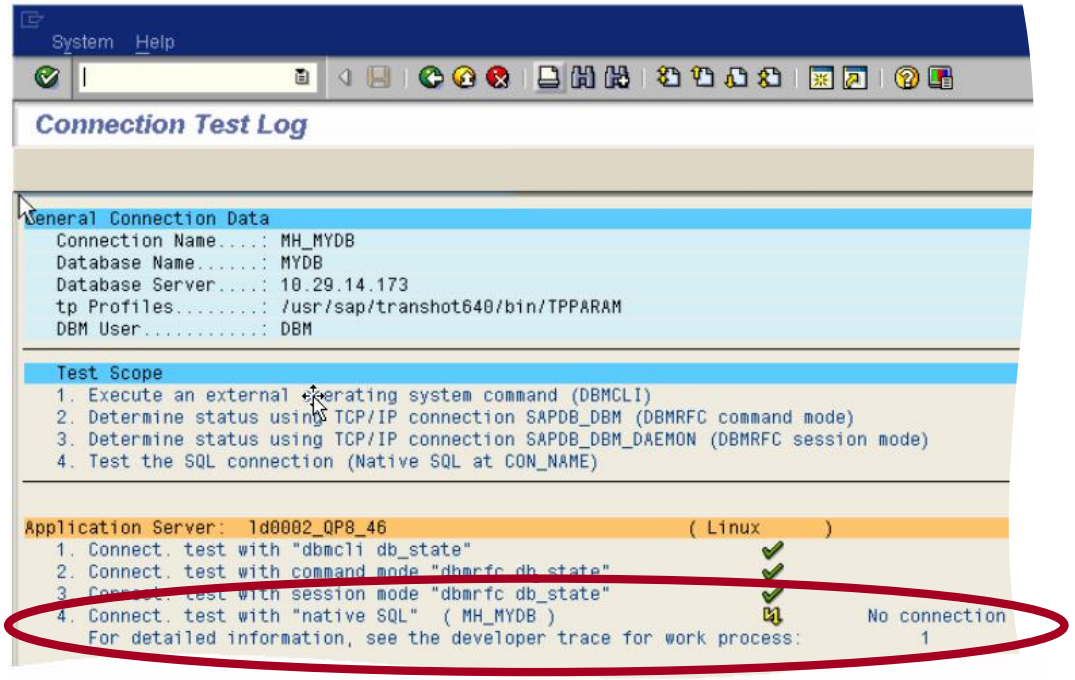


The connection test also fails if the DBM operator user and/or password has not been entered correctly.

Wrong Standard Database User Password

If the DBM Operator and his password have been entered correctly, the DBMCLI and DBMRFC connection work.

Then the native SQL test may fail because the standard database user has not been entered correctly.



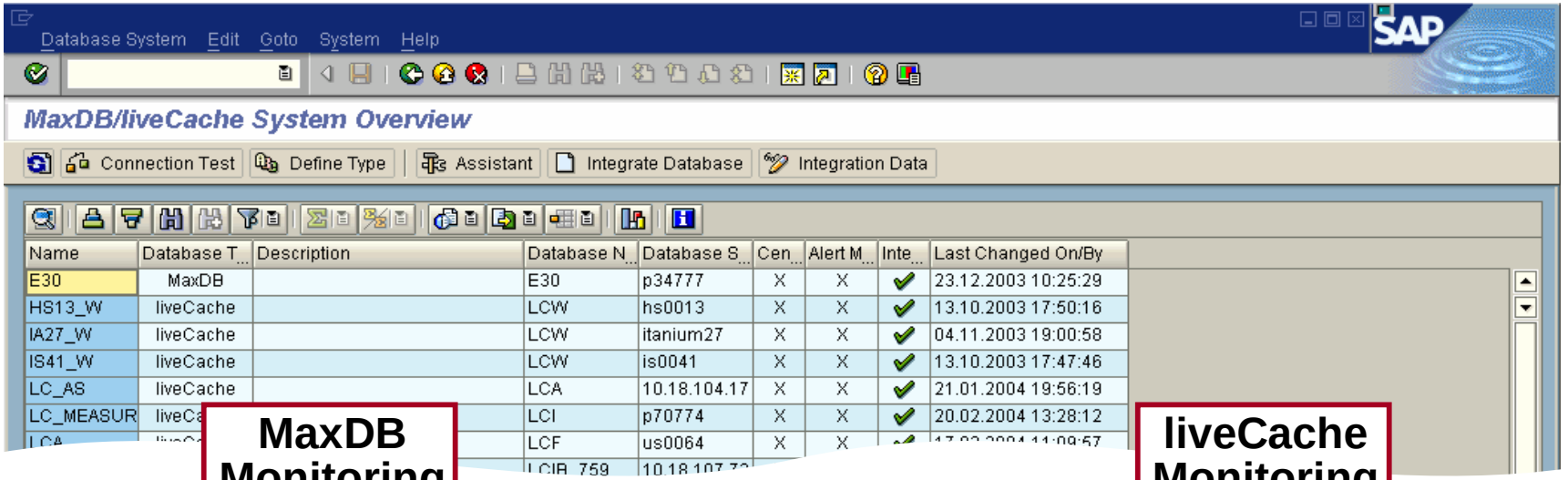

```
A  ABAP/4 Program SAPLSADB                                    .
A  Source LSADBU34                                  Line 14.
A  Error Code DBIF_DSQL2_SQL_ERROR.
A  Module  $Id: //bas/640_REL/src/krn/runt/abexsql.c#5 $ SAP.
A  Function DsqlErrorHandler Line 1986.
A  An SQL error occurred when executing Native SQL..
A
E
E Wed Jul  7 16:34:59 2004
E  Replication is disabled
B
B Wed Jul  7 16:35:01 2004
B  Connect to MH_MYDB as MONA with 10.29.14.173-MYDB
C  INFO : SQLOPT (set by environment) =
C  INFO : SQLOPT= -I 0 -t 0 -F SAPDB.23612.pct
C  Precompiler Runtime : C-PreComp 7.4.3    Build 029-121-050-967
C  Precompiler runtime is SAP DB 7.4.3.029
C  Try to connect as MONA/<pwd>@10.29.14.173-MYDB on connection 1 ...
C
C Wed Jul  7 16:35:02 2004
C  *** ERROR => CONNECT failed : sqlcode=-4008 (Unknown user name/password combination)
[dbsdbutl.c   325]
B  ***LOG BY2=> sql error -4008      performing CON        [dbds#3 @ 1044] [dbds    1044 ]
B  ***LOG BY0=> Unknown user name/password combination [dbds#3 @ 1044] [dbds    1044 ]
S
```

In this case you can find more information in the corresponding dev_w# file.

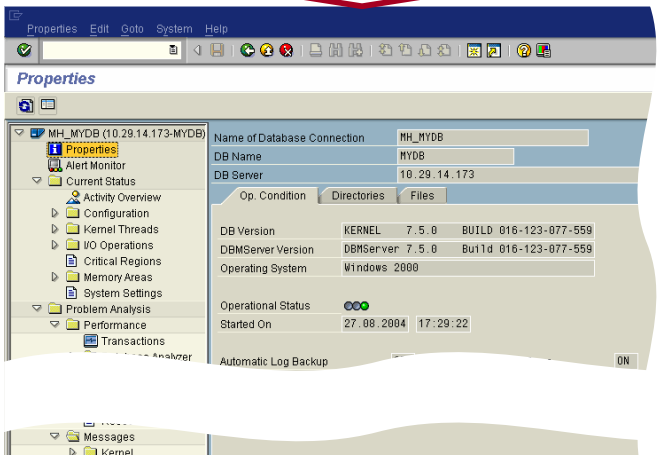
Error -4008 (Unknown user name/ password combination) indicates that the user data for the standard database user is not correct.

Remote Monitoring

Transaction DB59:
Central entry point to monitor MaxDB & liveCache instances

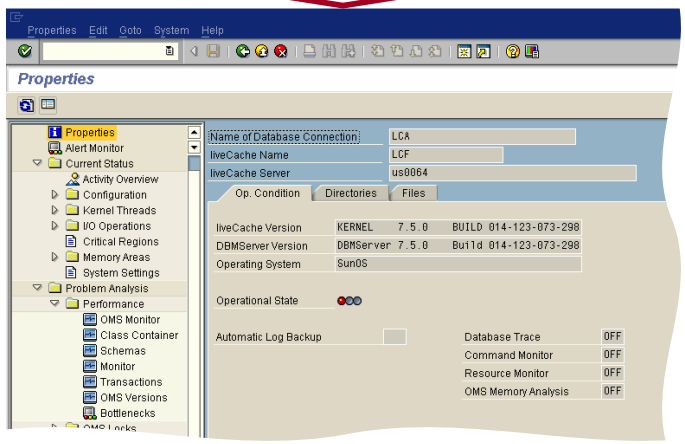


MaxDB Monitoring DB50



Monitoring of remote instances is possible. Even stand-alone databases like small test databases can be monitored in the SAP system.

liveCache Monitoring LC10



What Has to be Monitored?

Standby Database

Central Monitoring – Transaction DB59

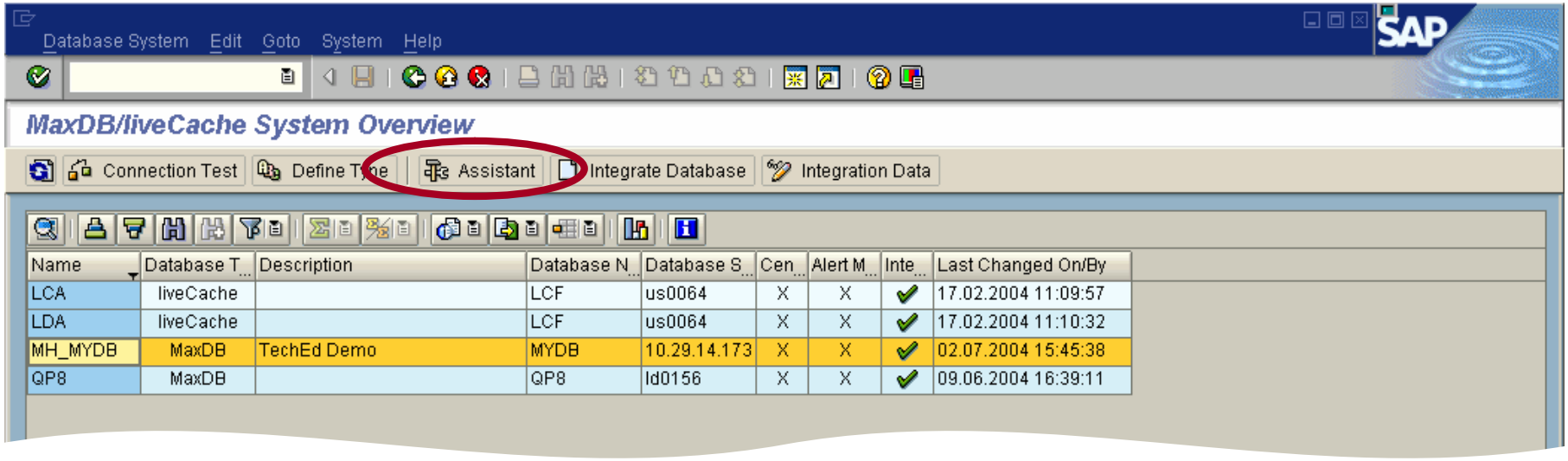
Database Assistant – Transaction DB50

Performance Analysis Tools

Alert Monitor – Transaction RZ20

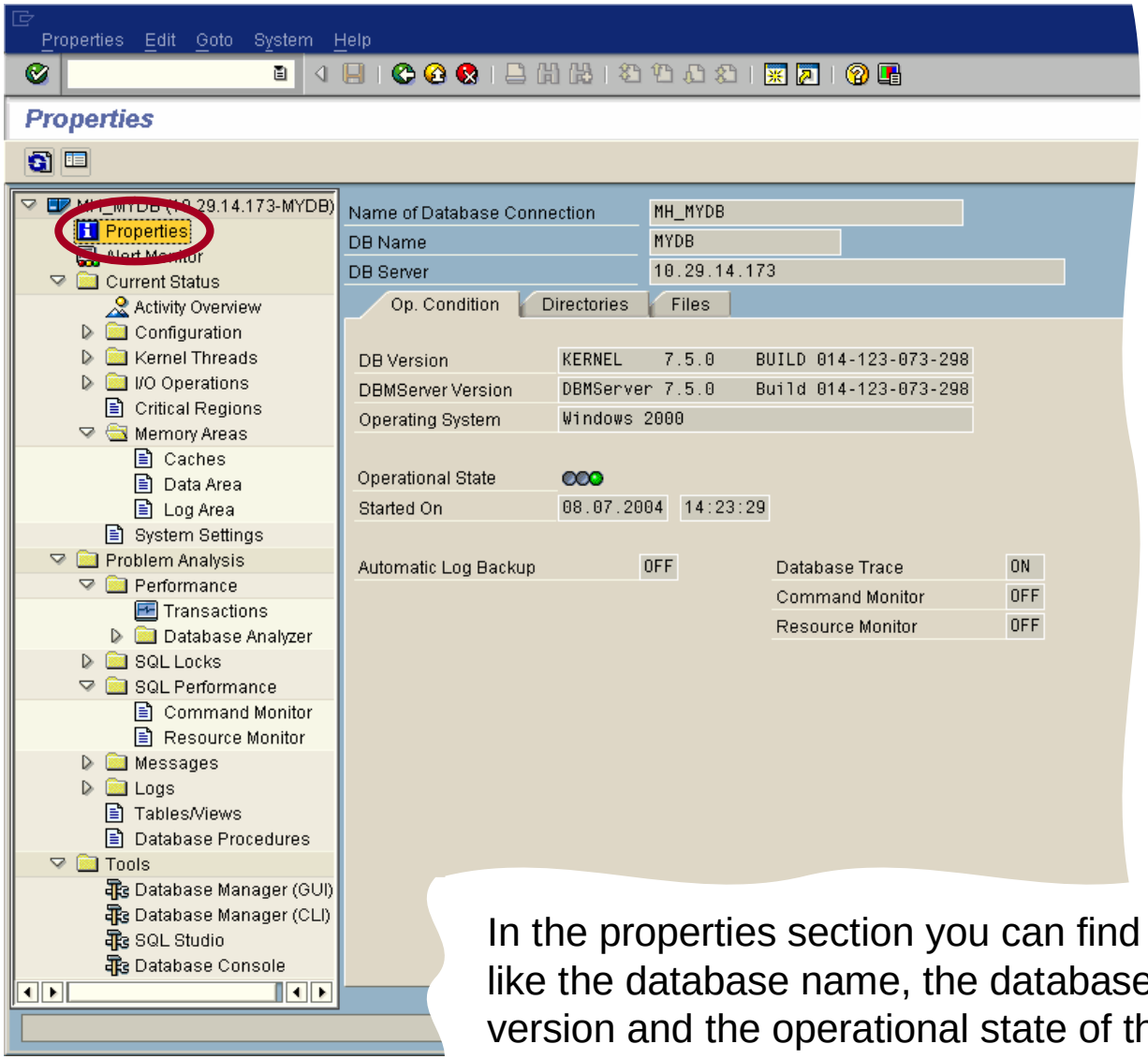
Snapshot

Starting Transaction DB50



After you entered the connect information for your database instance you can start the database assistant. Select the newly created entry in the list of databases and choose *Assistant*.

Transaction DB50 – Properties



This is the *Properties* section of transaction DB50N.

Transaction DB50 can only connect to the system's database instance.

Transaction DB50N is started from transaction DB59 as it is able to connect to different database instances.

DB50 and DB50N are nearly identical. In this presentation always DB50N is used, although the slides say DB50.

In the properties section you can find some general information like the database name, the database server, the database version and the operational state of the database instance.

Operational States

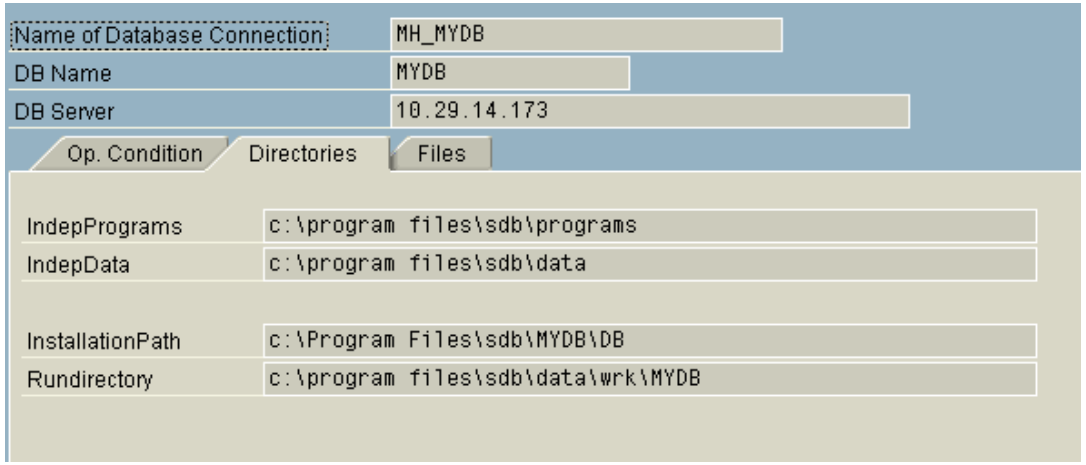
There are three possible operational states of MaxDB and liveCache:

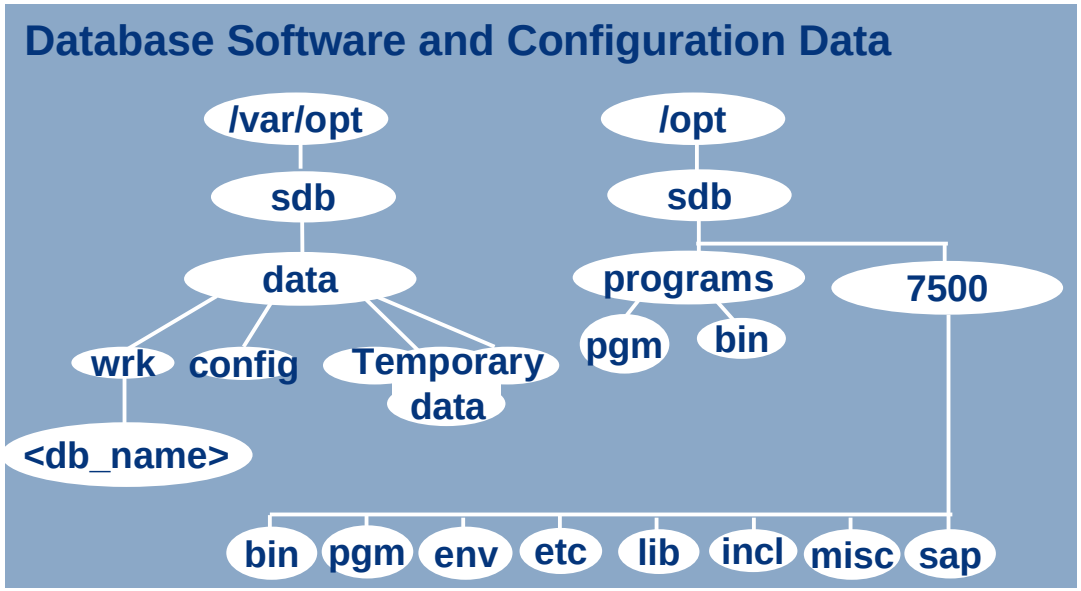
OFFLINE:
MaxDB kernel processes and caches do not exist. No user can use the database.

ADMIN:
The MaxDB kernel is active (processes are started, caches are initialized). Users cannot connect to the database. Only the DBM operator can connect and perform administrative tasks.

ONLINE:
The MaxDB kernel is active and ready to work. Users can connect to the database.

MaxDB And liveCache Directory Structure





The IndepPrograms directory contains programs and libraries shared by the MaxDB instances and MaxDB applications.
These programs are downwards compatible.

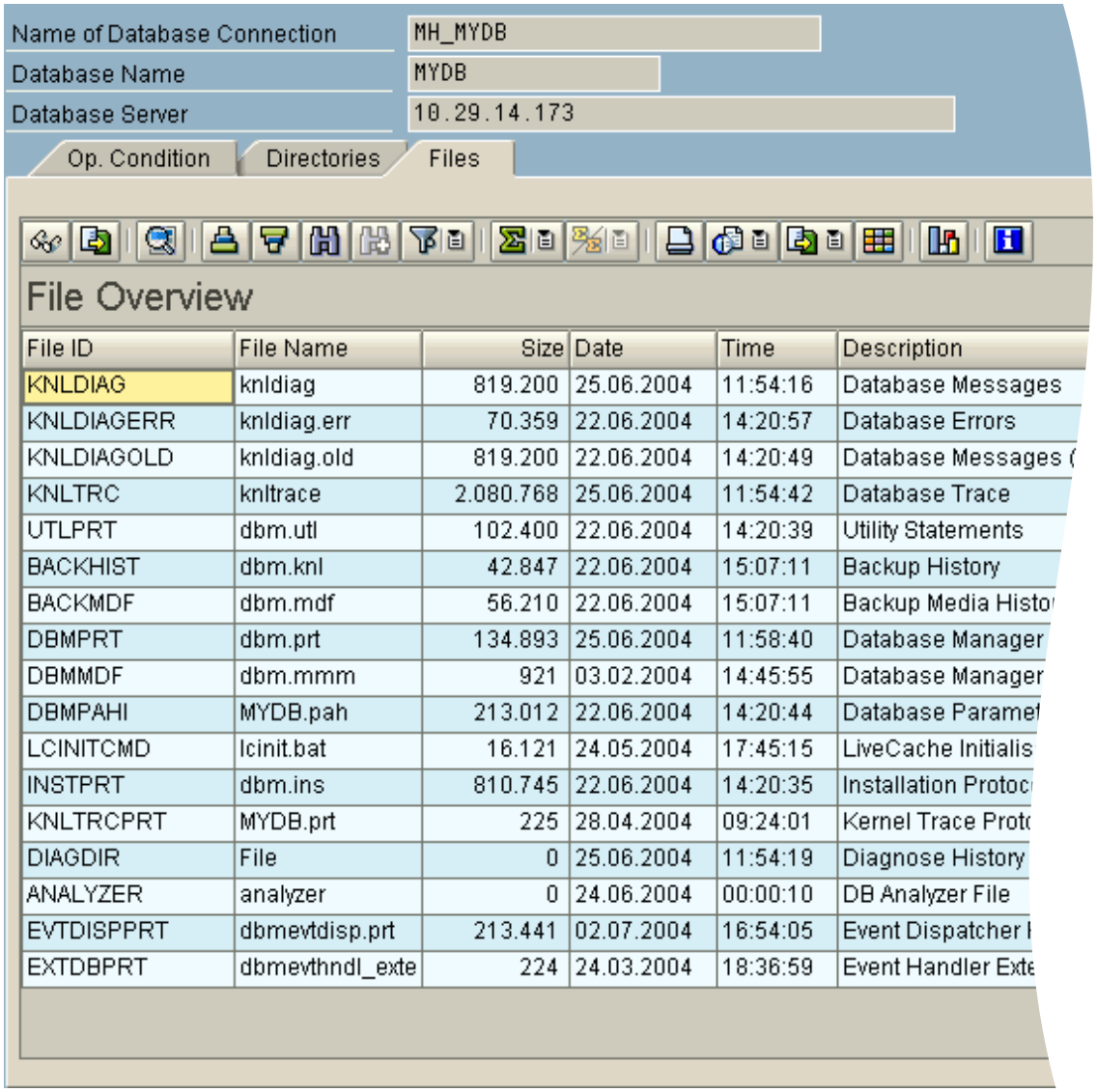
The IndepData directory contains the configuration data and rundirectories of MaxDB instances.

The location of these directories is specified during the first installation of MaxDB software. They exist only once on the server.

The InstallationPath contains the server software that depends on the database version (e.g. kernel). Several dependent directories can exist alongside each other.

The rundirectory contains the status files of a MaxDB instance.

MaxDB Status And Log files



File ID	File Name	Size	Date	Time	Description
KNLDIAG	knldiag	819.200	25.06.2004	11:54:16	Database Messages
KNLDIAGERR	knldiag.err	70.359	22.06.2004	14:20:57	Database Errors
KNLDIAGOLD	knldiag.old	819.200	22.06.2004	14:20:49	Database Messages (
KNLTRC	knltrace	2.080.768	25.06.2004	11:54:42	Database Trace
UTLPRT	dbm.utl	102.400	22.06.2004	14:20:39	Utility Statements
BACKHIST	dbm.knl	42.847	22.06.2004	15:07:11	Backup History
BACKMDF	dbm.mdf	56.210	22.06.2004	15:07:11	Backup Media Histo
DBMPRT	dbm.prt	134.893	25.06.2004	11:58:40	Database Manager
DBMMDF	dbm.mmm	921	03.02.2004	14:45:55	Database Manager
DBMPAHI	MYDB.pah	213.012	22.06.2004	14:20:44	Database Paramet
LCINITCMD	lcinit.bat	16.121	24.05.2004	17:45:15	LiveCache Initialis
INSTPRT	dbm.ins	810.745	22.06.2004	14:20:35	Installation Protoc
KNLTRCPRT	MYDB.prt	225	28.04.2004	09:24:01	Kernel Trace Prot
DIAGDIR	File	0	25.06.2004	11:54:19	Diagnose History
ANALYZER	analyzer	0	24.06.2004	00:00:10	DB Analyzer File
EVTDISPPRT	dbmevtdisp.prt	213.441	02.07.2004	16:54:05	Event Dispatcher I
EXTDBPRT	dbmevthndl_exte	224	24.03.2004	18:36:59	Event Handler Exte

Most important log files:

KNLDIAG - contains status and error messages of the database kernel

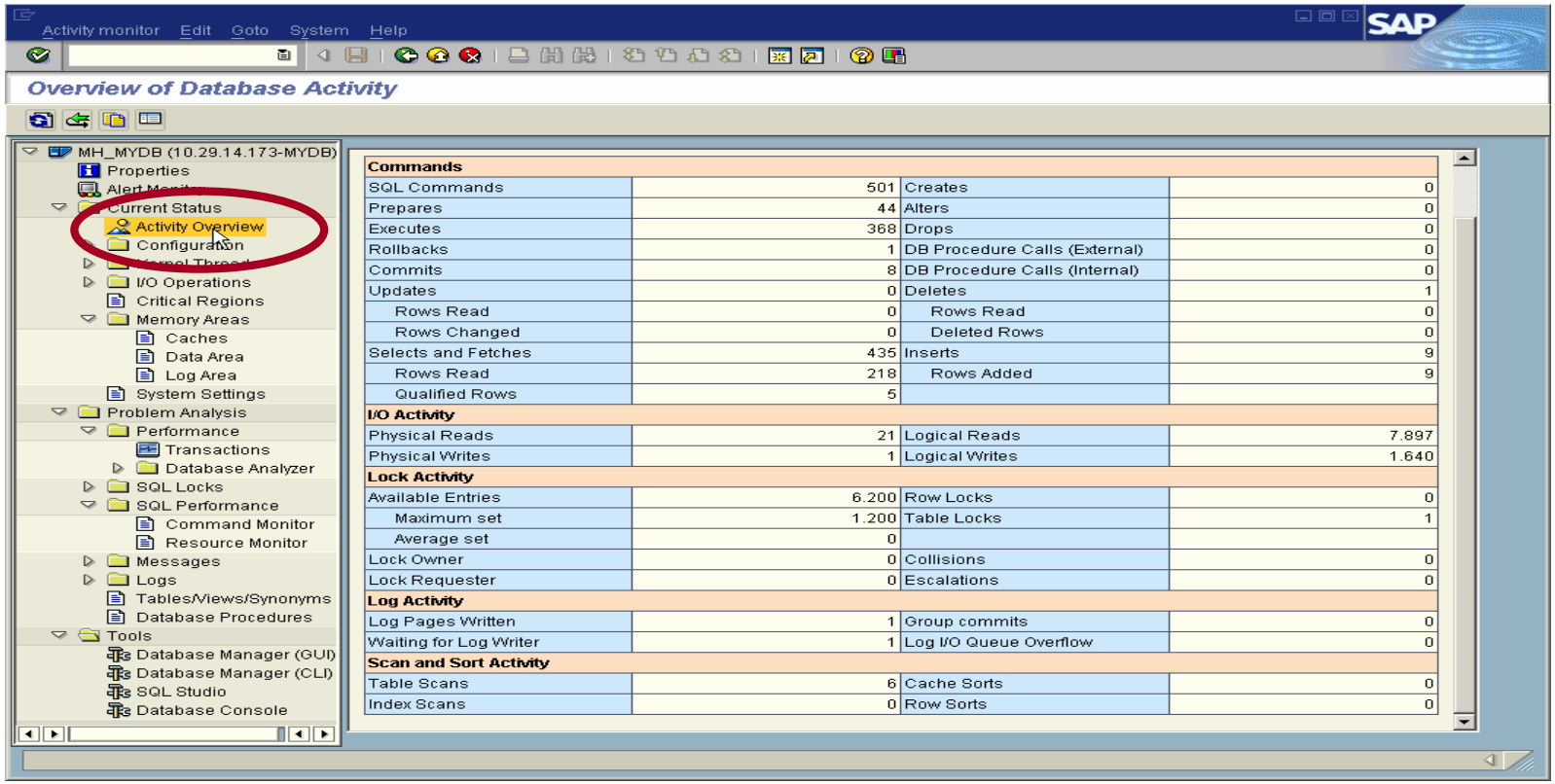
KNLDIAGERR - contains all error messages since database installation

UTLPRT - contains administrative commands sent to the database kernel (e.g. SHUTDOWN, BACKUP, CHECK DATA) including their return code(s)

BACKHIST - contains all backup and recovery actions

DBMPRT - contains all (administrative) commands sent to the dbmserver

Database Activity

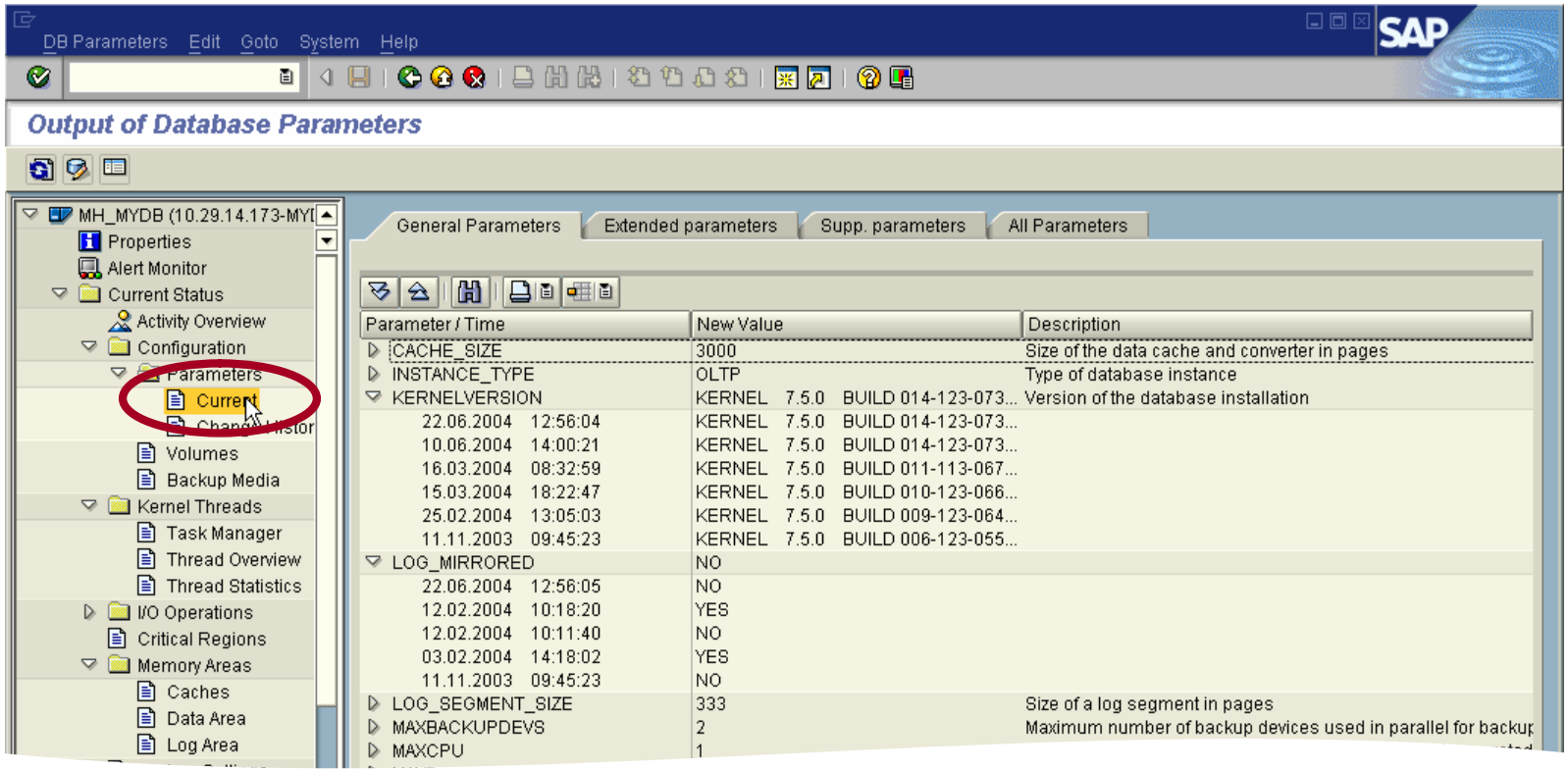


Commands			
SQL Commands	501	Creates	0
Prepares	44	Alters	0
Executes	368	Drops	0
Rollbacks	1	DB Procedure Calls (External)	0
Commits	8	DB Procedure Calls (Internal)	0
Updates	0	Deletes	1
Rows Read	0	Rows Read	0
Rows Changed	0	Deleted Rows	0
Selects and Fetches	435	Inserts	9
Rows Read	218	Rows Added	9
Qualified Rows	5		
I/O Activity			
Physical Reads	21	Logical Reads	7.897
Physical Writes	1	Logical Writes	1.640
Lock Activity			
Available Entries	6.200	Row Locks	0
Maximum set	1.200	Table Locks	1
Average set	0		
Lock Owner	0	Collisions	0
Lock Requester	0	Escalations	0
Log Activity			
Log Pages Written	1	Group commits	0
Waiting for Log Writer	1	Log I/O Queue Overflow	0
Scan and Sort Activity			
Table Scans	6	Cache Sorts	0
Index Scans	0	Row Sorts	0

The *Activity Overview* gives an overview of the database activity since it was started. E.g. you can see:

- the number of SQL statements executed,
- the number of lock collisions,
- the number of lock escalations

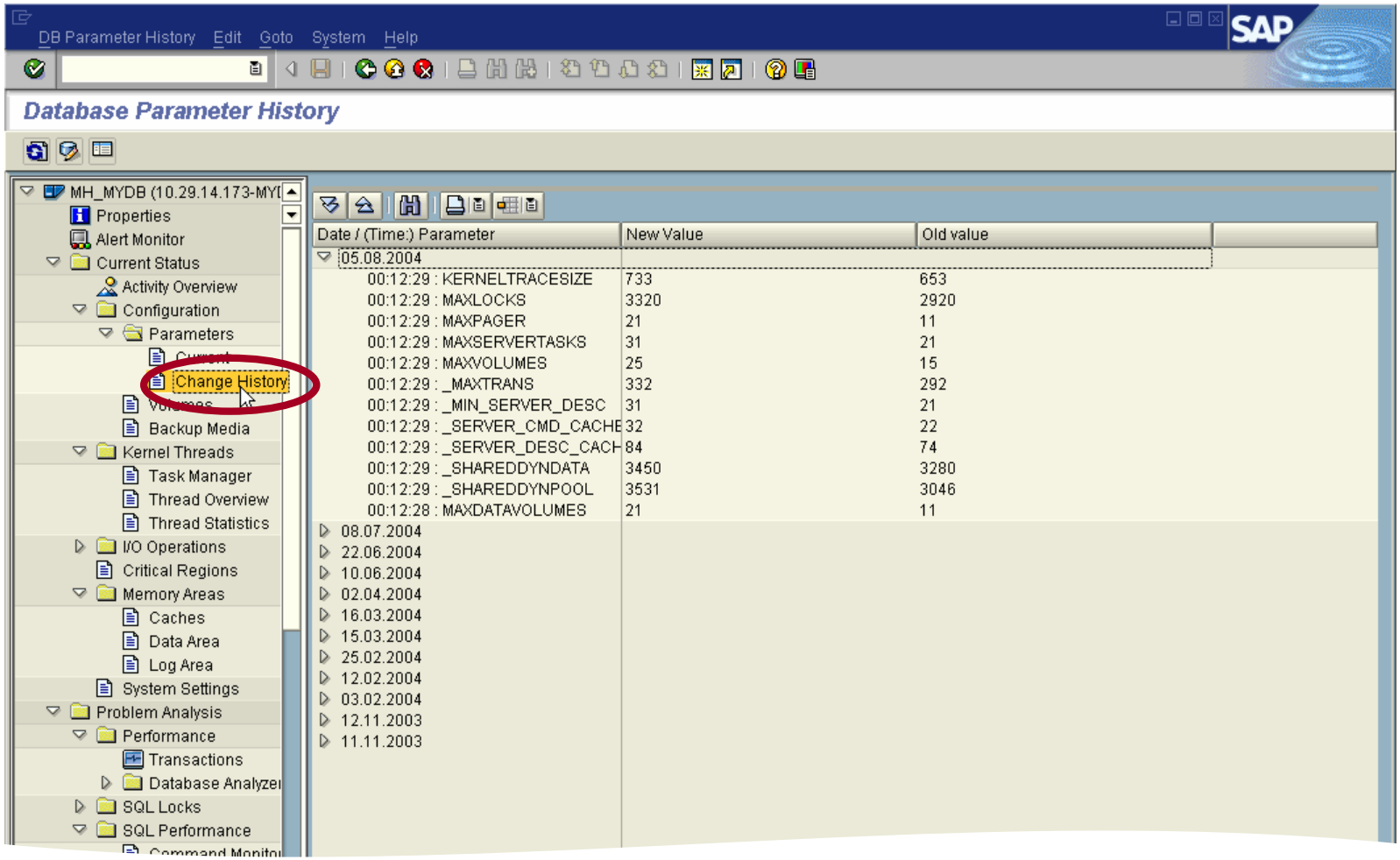
Parameter Settings



You can check the current parameter settings and the change history of each parameter. The most important parameters are listed in the *General Parameters* section. All other parameters should only be changed if requested by the MaxDB support.

Parameters can be changed using the Database Manager GUI. Whenever a parameter has been changed, the database has to be restarted that the new parameter settings take effect (see note #814704 for online changeable parameters).

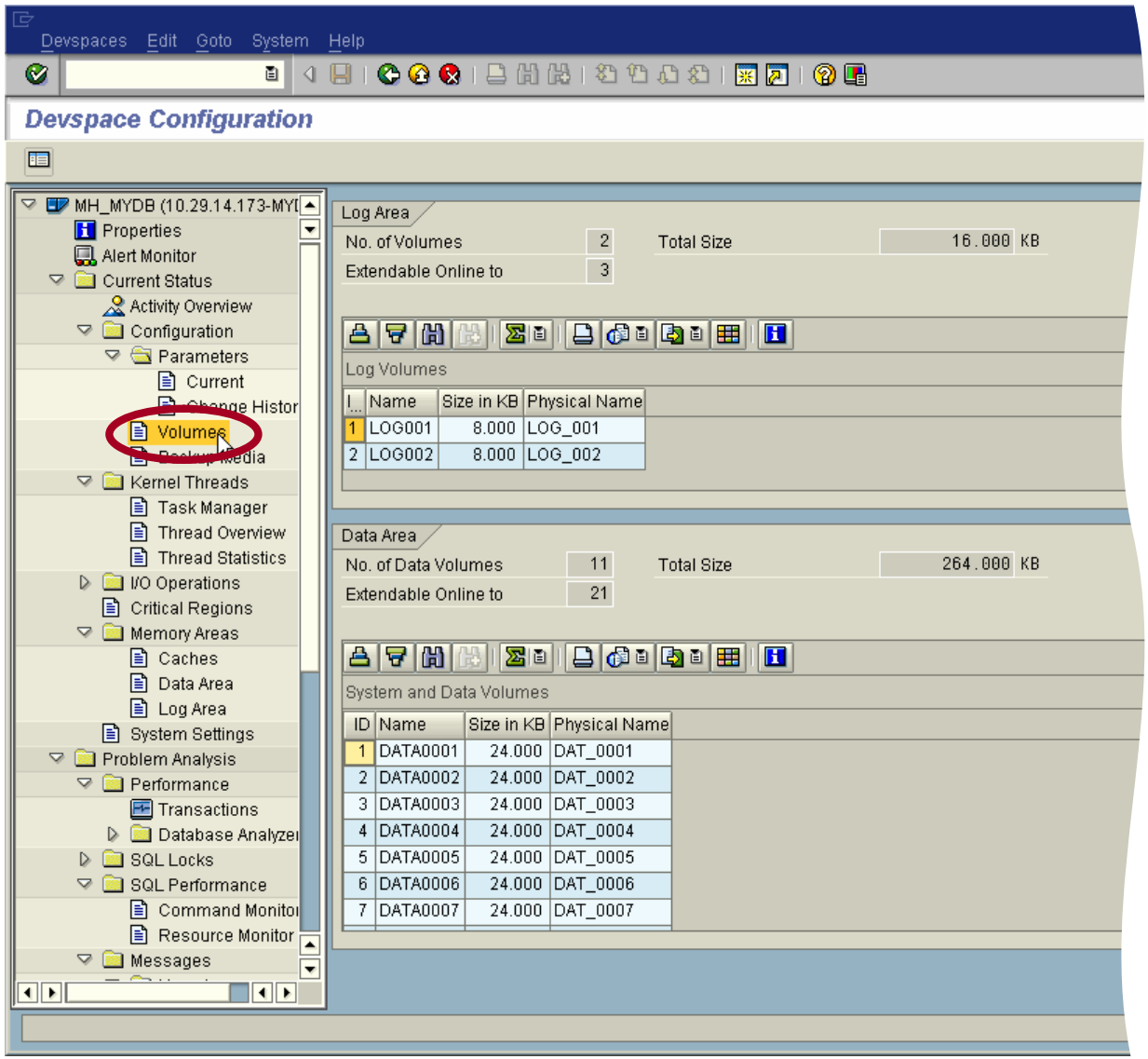
Parameter Change History



It is also possible to get an overview of the parameter changes by date. You can see

- which parameters have been changed on which day
- the old as well as the new parameter values.

Volume Configuration



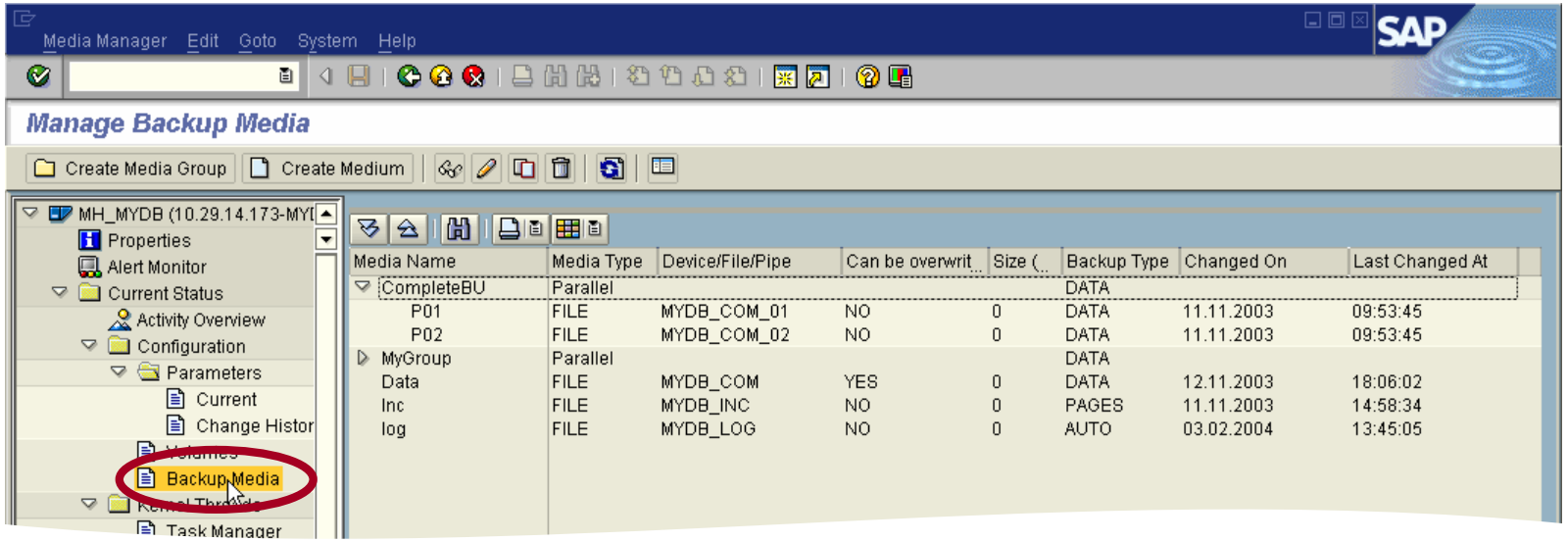
The *Volumes* section shows all configured log and data volumes including their locations and sizes.

You can also see, how many data volumes could be added while the database is ONLINE.

This is limited by the database parameter MAXDATAVOLUMES.

New volumes can be added using the Database Manager GUI.

Backup Media

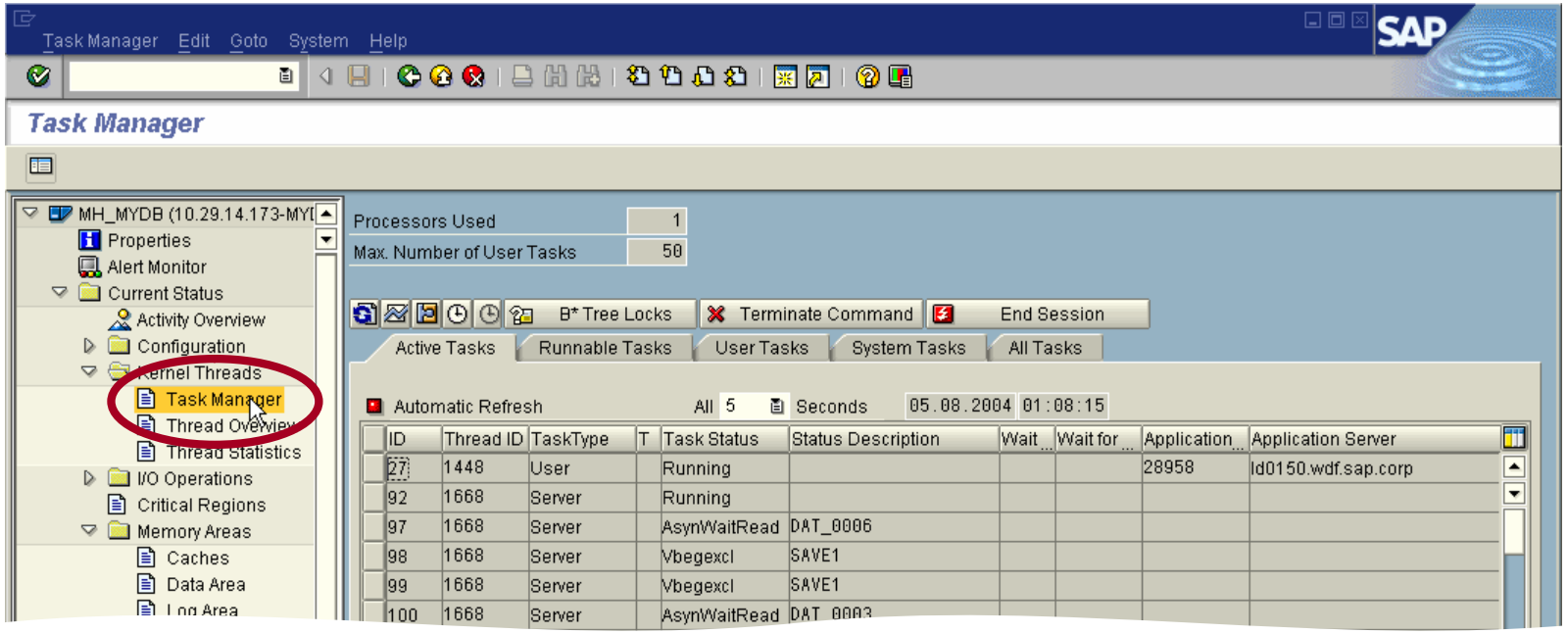


The *Backup Media* section provides an overview of all defined backup media. It is also possible to define new backup media.
MaxDB supports parallel backups to several files/tapes/pipes.
Therefore a media group has to be defined which consists of several single backup media. The following external backup tools can be used to create backups:

- **Tivoli Storage Manager**
- **Networker**
- **Tools which support the Interface BackInt for Oracle**

The backup media can also be defined with Database Manager GUI. Backups are created using this tool as well or with transaction DB13/DB13C or the DBACOCKPIT.

Task Manager



The *Task Manager* shows the status of all currently active database tasks. In a running system, possible states are:

Running – task is in kernel code of MaxDB and uses CPU

Runnable, Vsleep – task is in kernel code of MaxDB and waiting for a free slot in its thread (UKT)

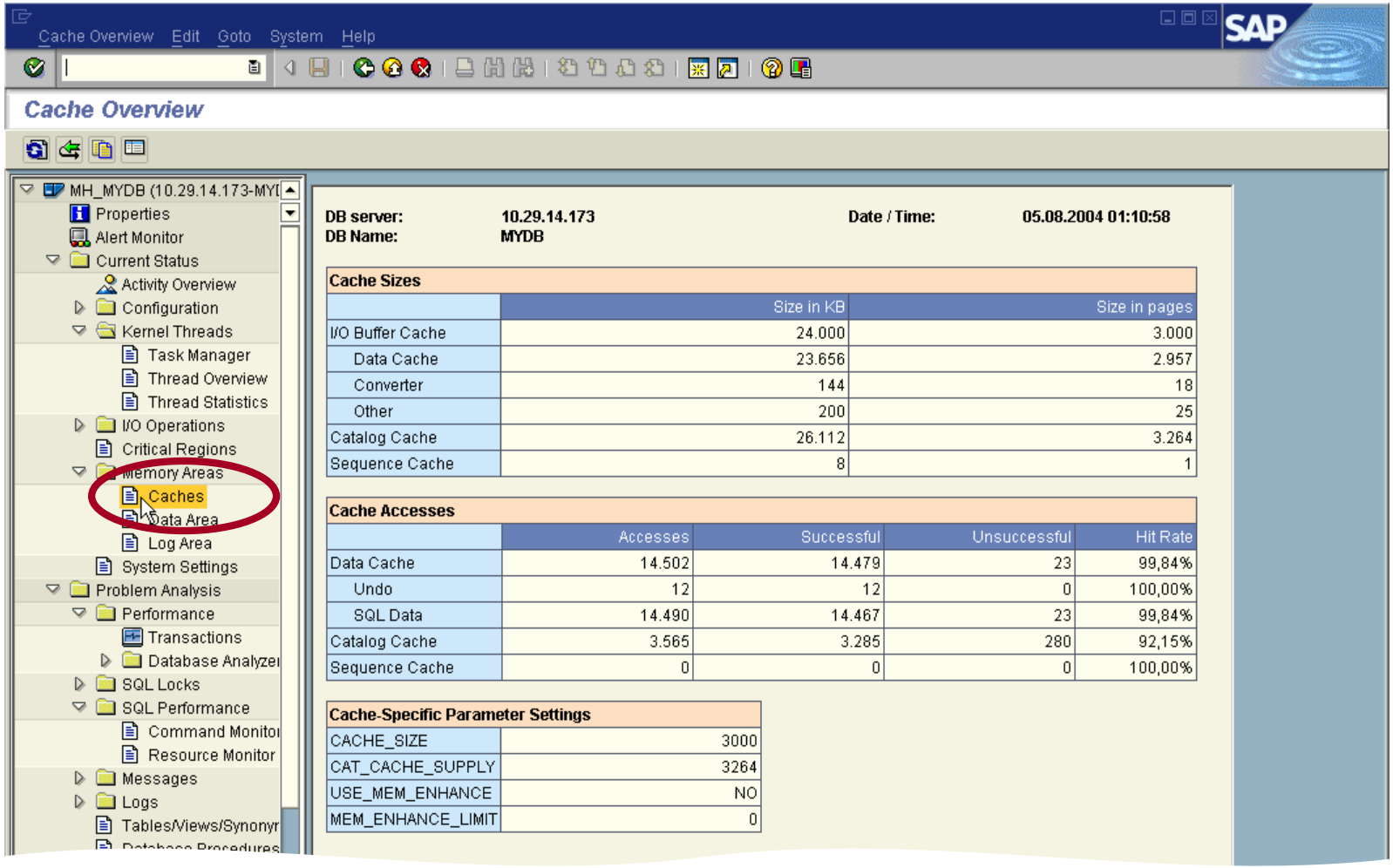
LogIOWait – task waits for completion of its log request by the archive log writer

IOWait (R) or IOWait (W) – task waits for data I/O completion (read or write)

Vbegexcl or Vsuspend – task waits to acquire an internal lock in MaxDB

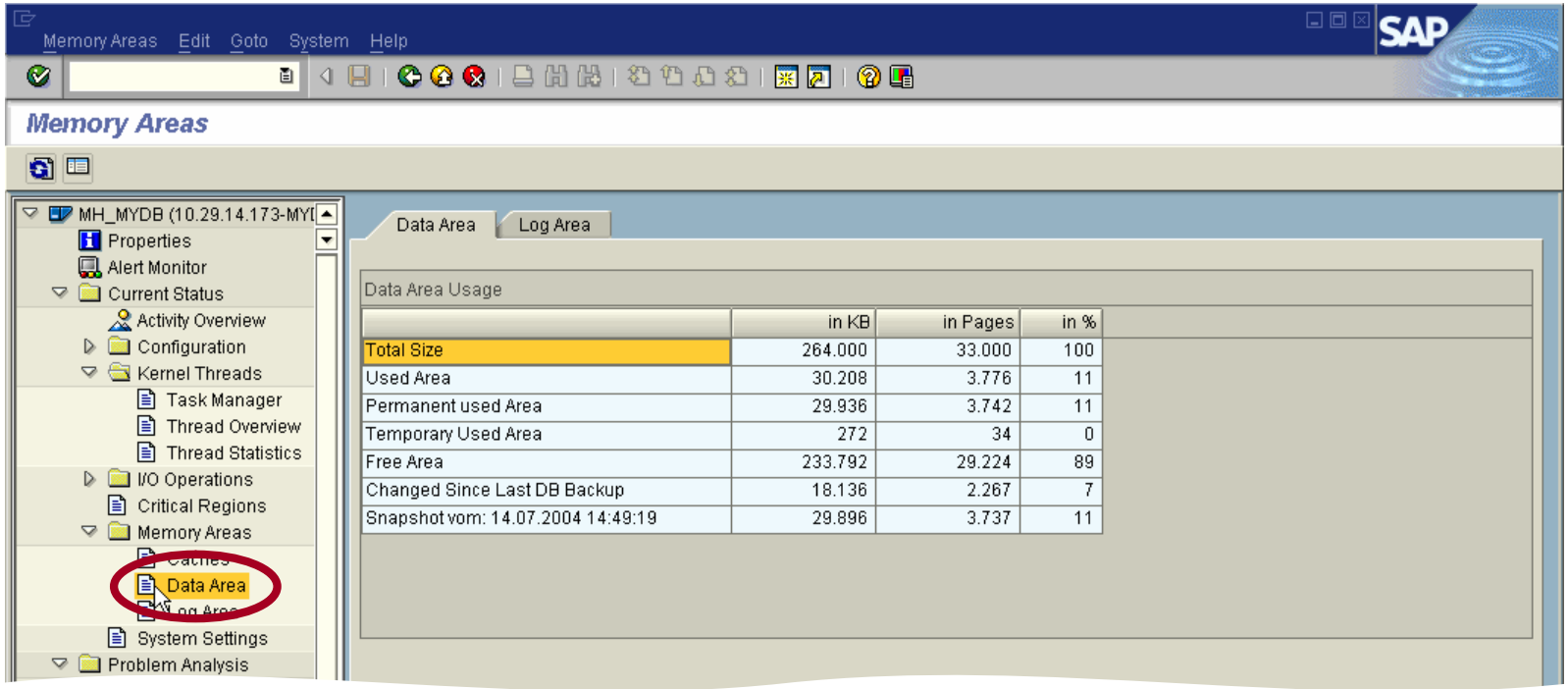
Vwait – task waits for an SQL lock held by another application process to be released (locks are released after a COMMIT or ROLLBACK)

Memory Areas: Caches



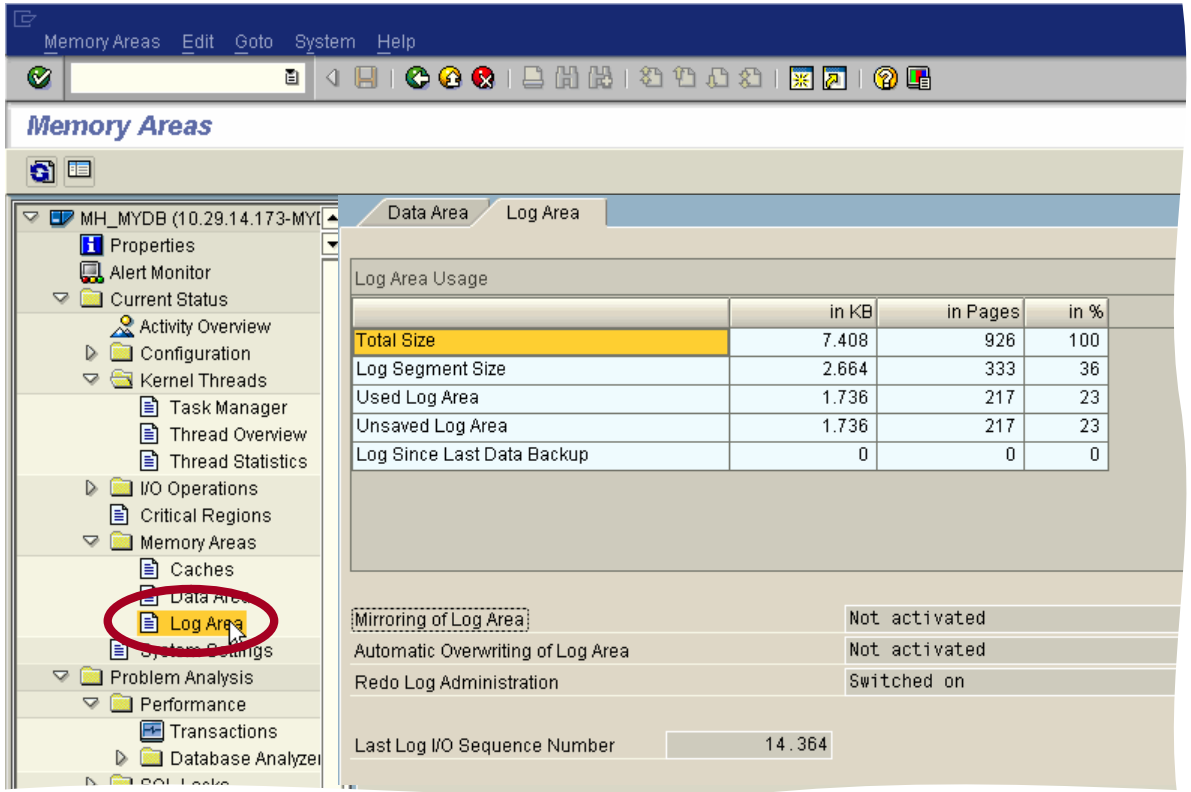
The *Caches* area shows the configured sizes of the different memory areas and the hit rates of these caches. The Data Cache hit rate should always be >= 98%.

Memory Areas: Data Area



The data area can consist of several data volumes.
The *Total Size* shows the sum of the sizes of all data volumes.
You can see the filling level of the data area as well as the proportion of temporary data.
The data is automatically distributed to all volumes equally. You don't have to define table spaces.
Empty data pages are reused by the database automatically. No reorganization is necessary.

Memory Areas: Log Area



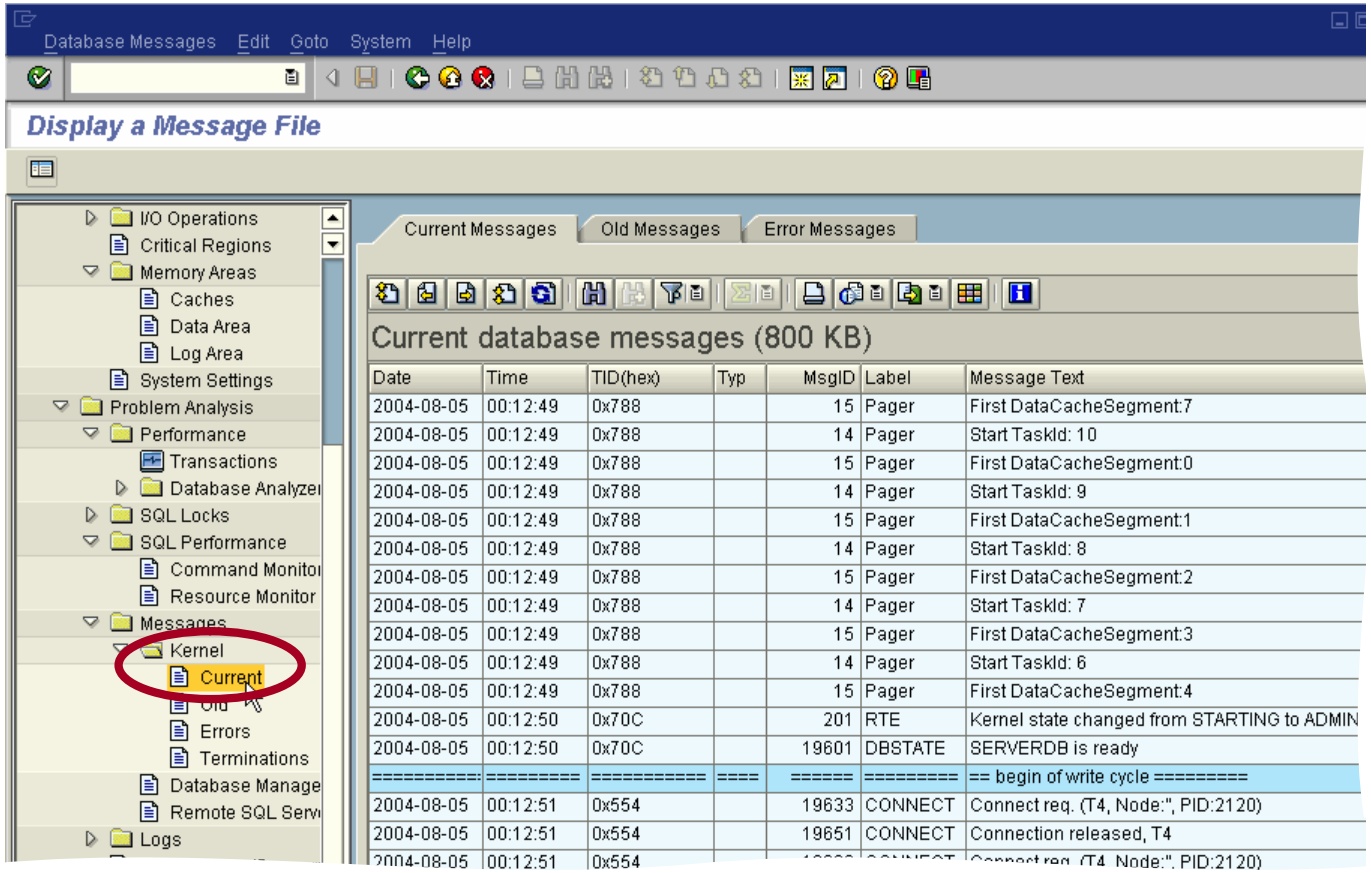
The log area can consist of several log volumes - however they are used as one single log area.

The *Log Segment Size* determines how large the log backups are created by the autosave log.

Whenever *Log Segment Size* log pages are written, they are copied to the log backup file and the pages in the log volume can be overwritten.

For test or demo systems it is possible to activate an overwrite mode for the log volumes - then you don't have to take log backups before the information on the log volume can be overwritten.
It is also possible to switch off the writing of log information.
Both of these possibilities are not recommended for productive systems as you won't be able to restore the database to the latest state.

Database Messages



File knldiag has a fixed size.

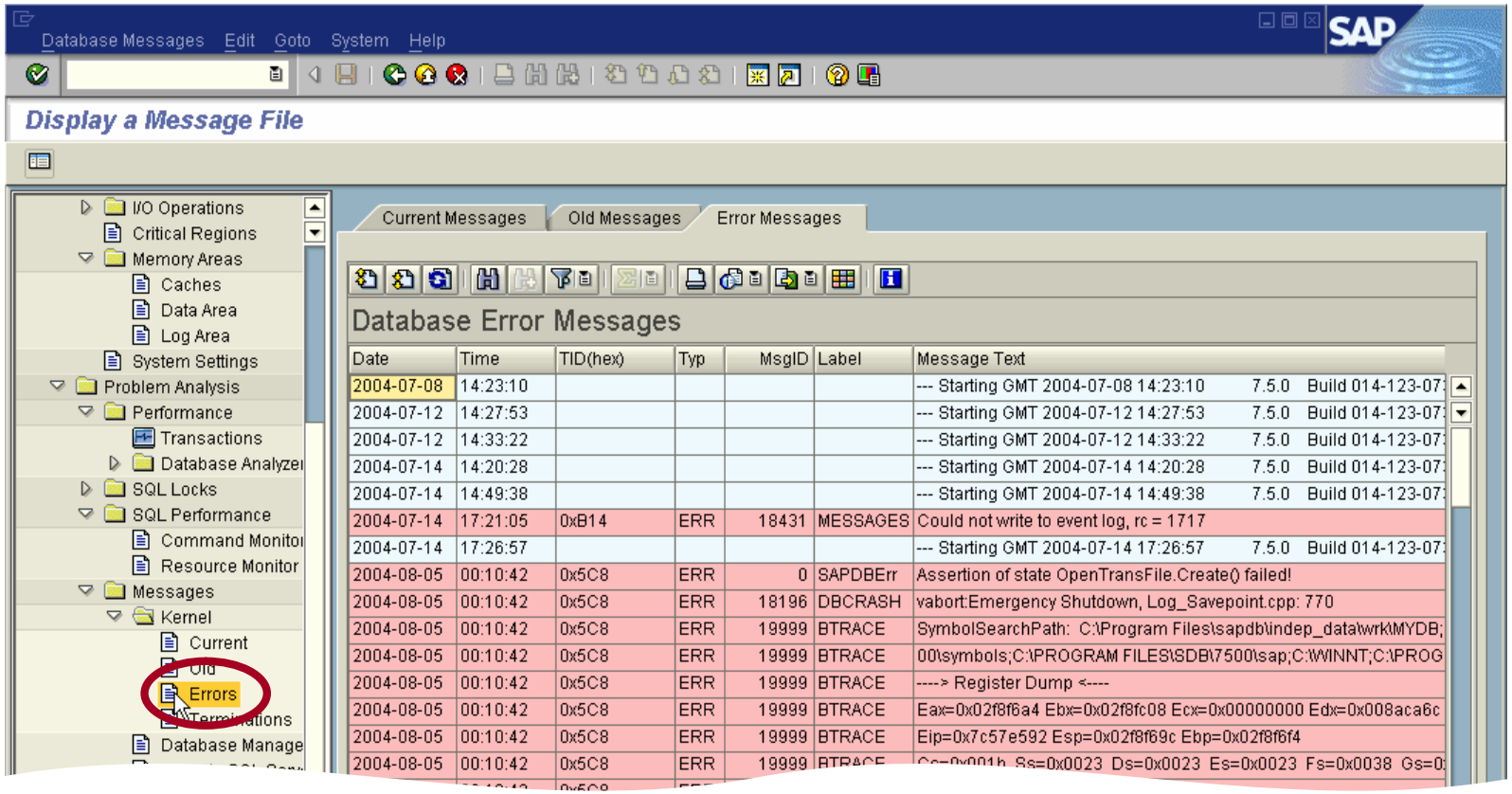
It is initialized when the database is started.

The last version of this file is then copied to knldiag.old.

Knldiag consists of two parts:
the first part contains information about the database start and is not overwritten.

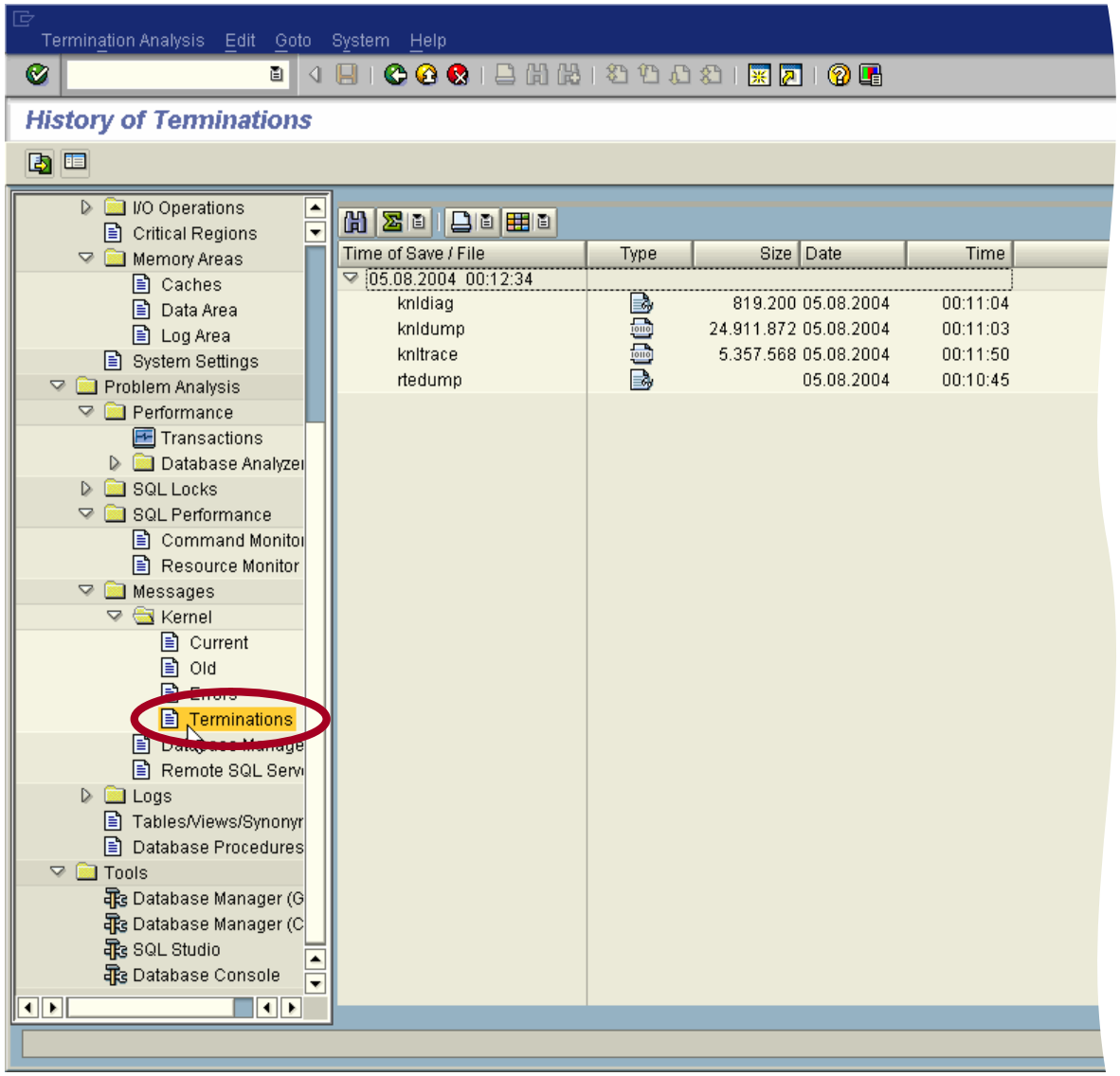
In the second part information is logged during the runtime of the database. This part is overwritten cyclically. The current write position is marked with
`'--- current write position --- '`
In case of problems with the database you should always check file knldiag for error messages.

Database Error Messages



File knldiag.err contains message '`--- Starting`…' whenever the database has been started from state OFFLINE to state ADMIN.
All other messages are error messages - e.g. information about a crash, including a back trace which can be used by the developers to find the cause of the crash.

Database Terminations



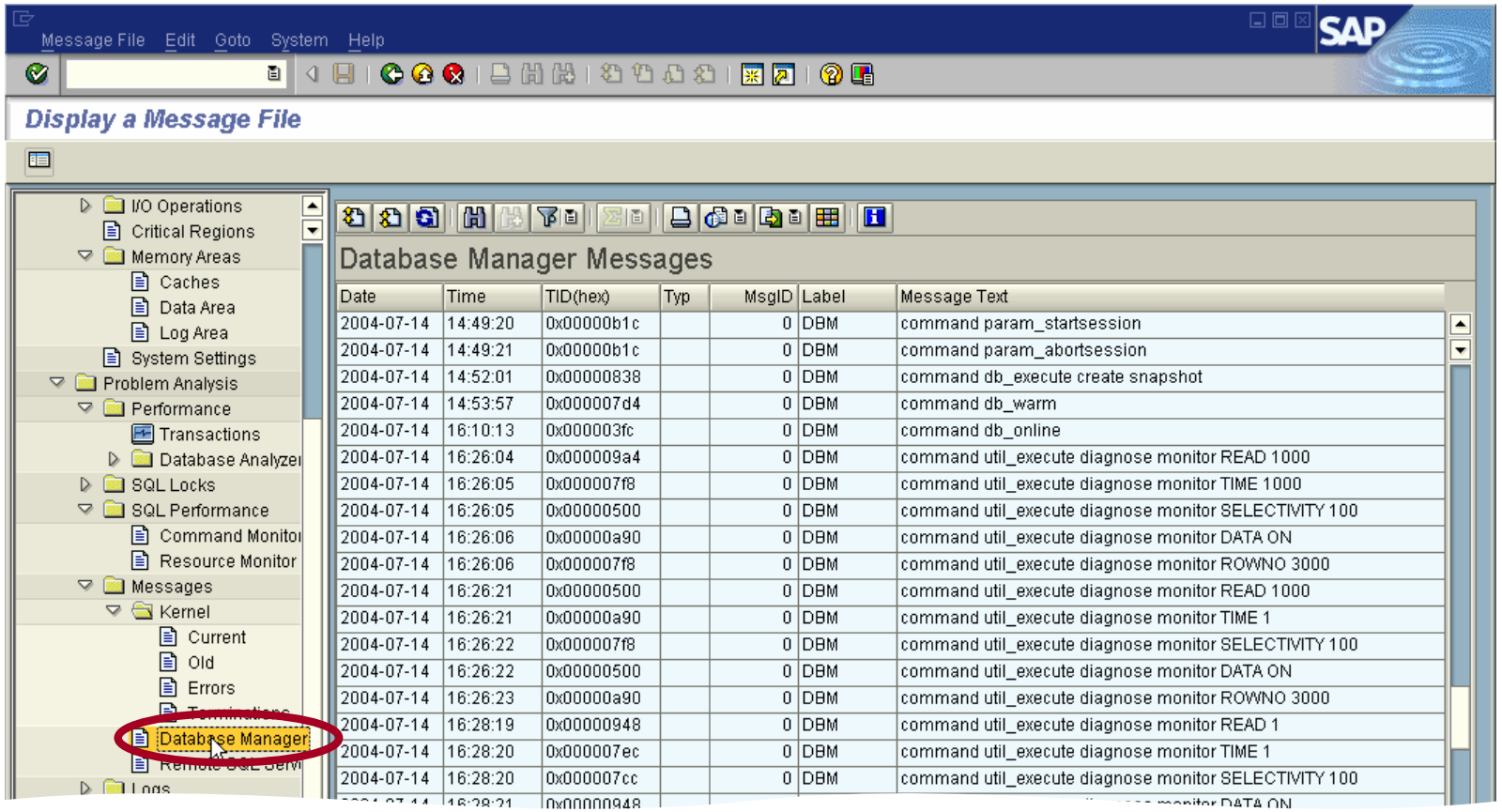
If the database is not stopped correctly the most important log files are saved in the DIAGHISTORY folder during the next start of the database instance.

This ensures that they are not overwritten and can still be analyzed to determine the cause of the crash.

These files can be seen in the *Terminations* section.

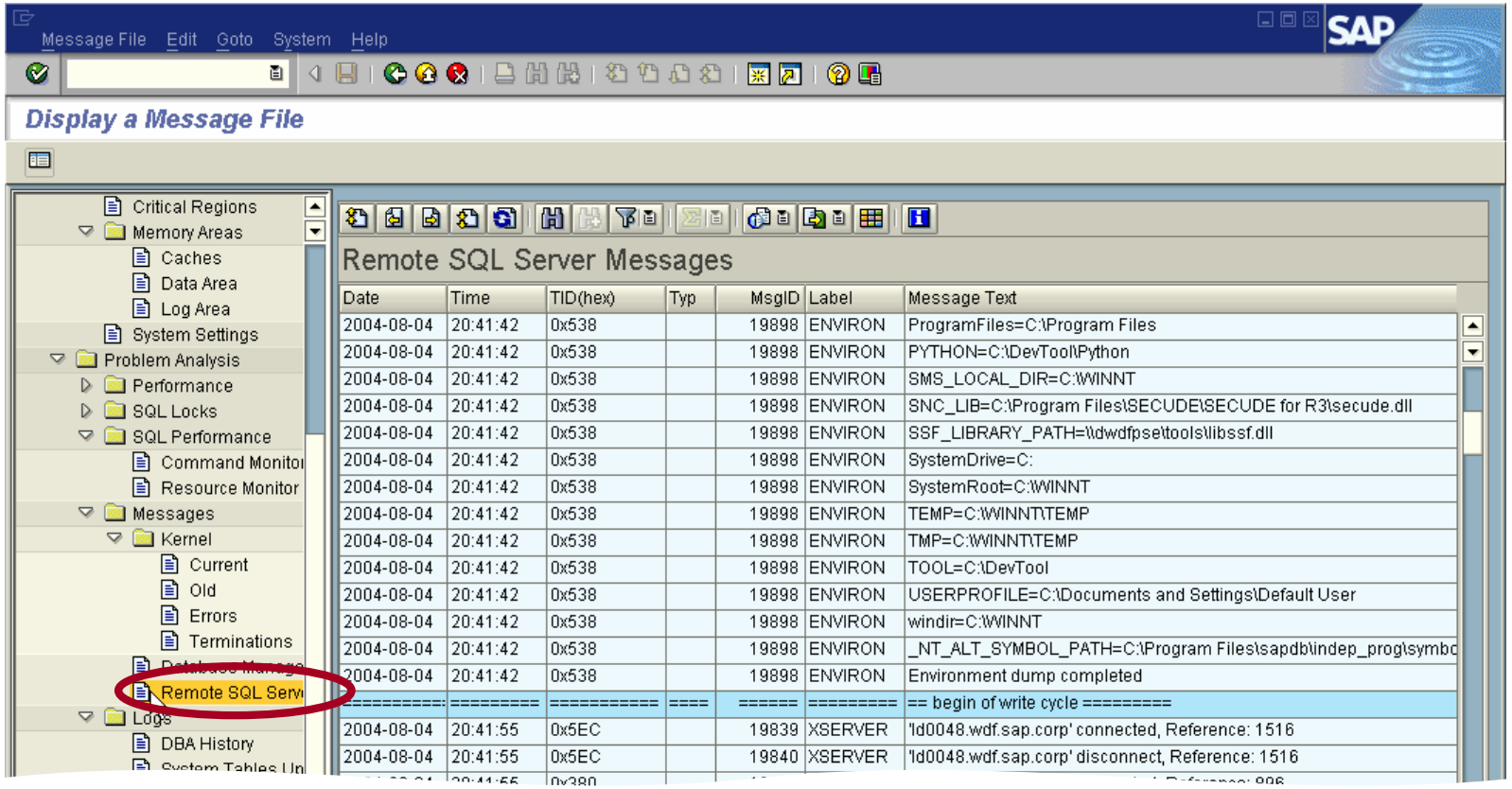
As a default only two sets of log files are held in the diaghistory.

Database Manager Messages



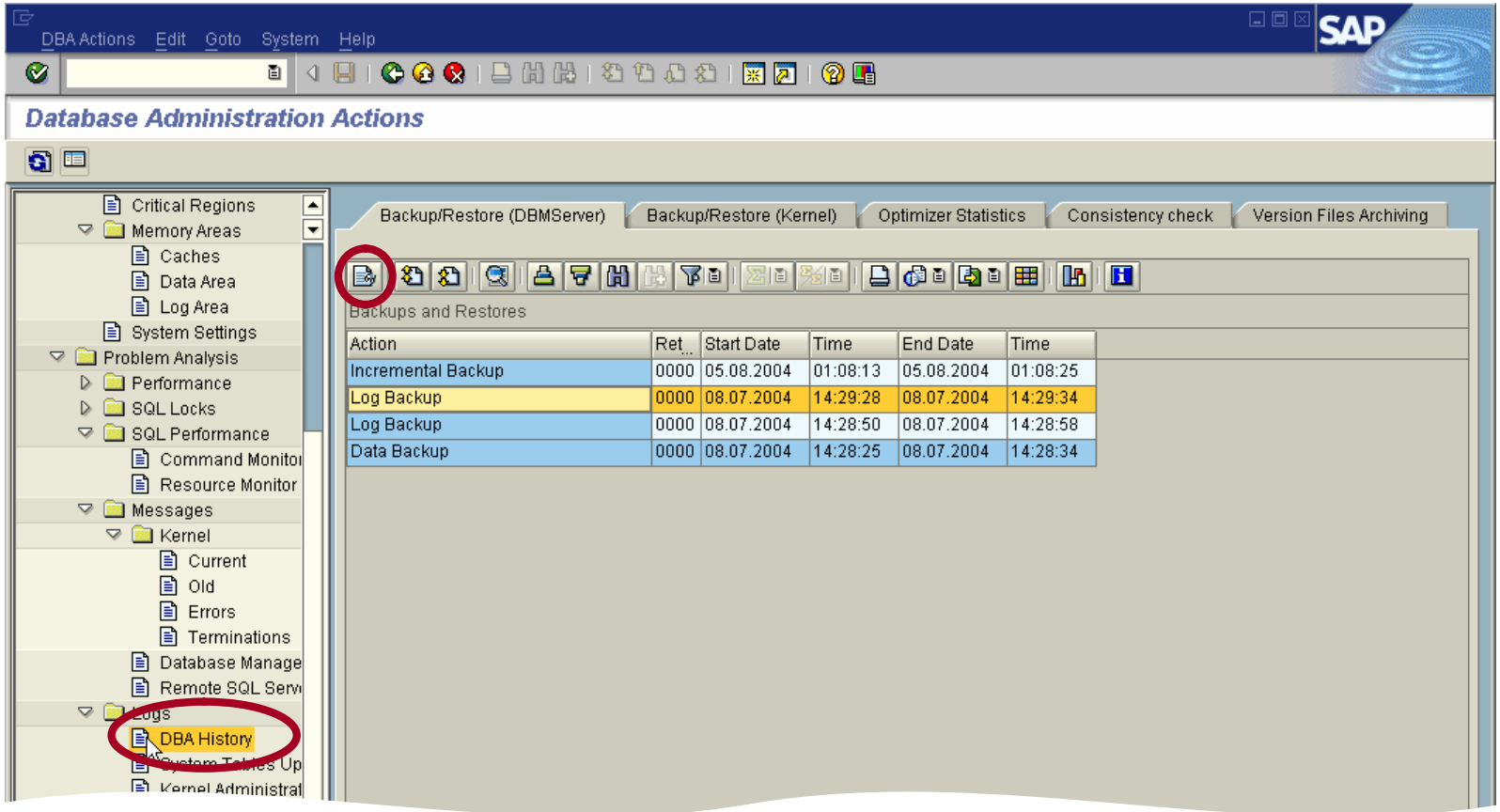
File dbm.prt contains statements sent to the dbmserver.
Whenever an administrative command has been executed using Database Manager GUI or CLI this is logged in this file, including the error code (if an error occurred).
E.g. you can see, when a start or stop command has been executed.

Remote SQL Server Messages



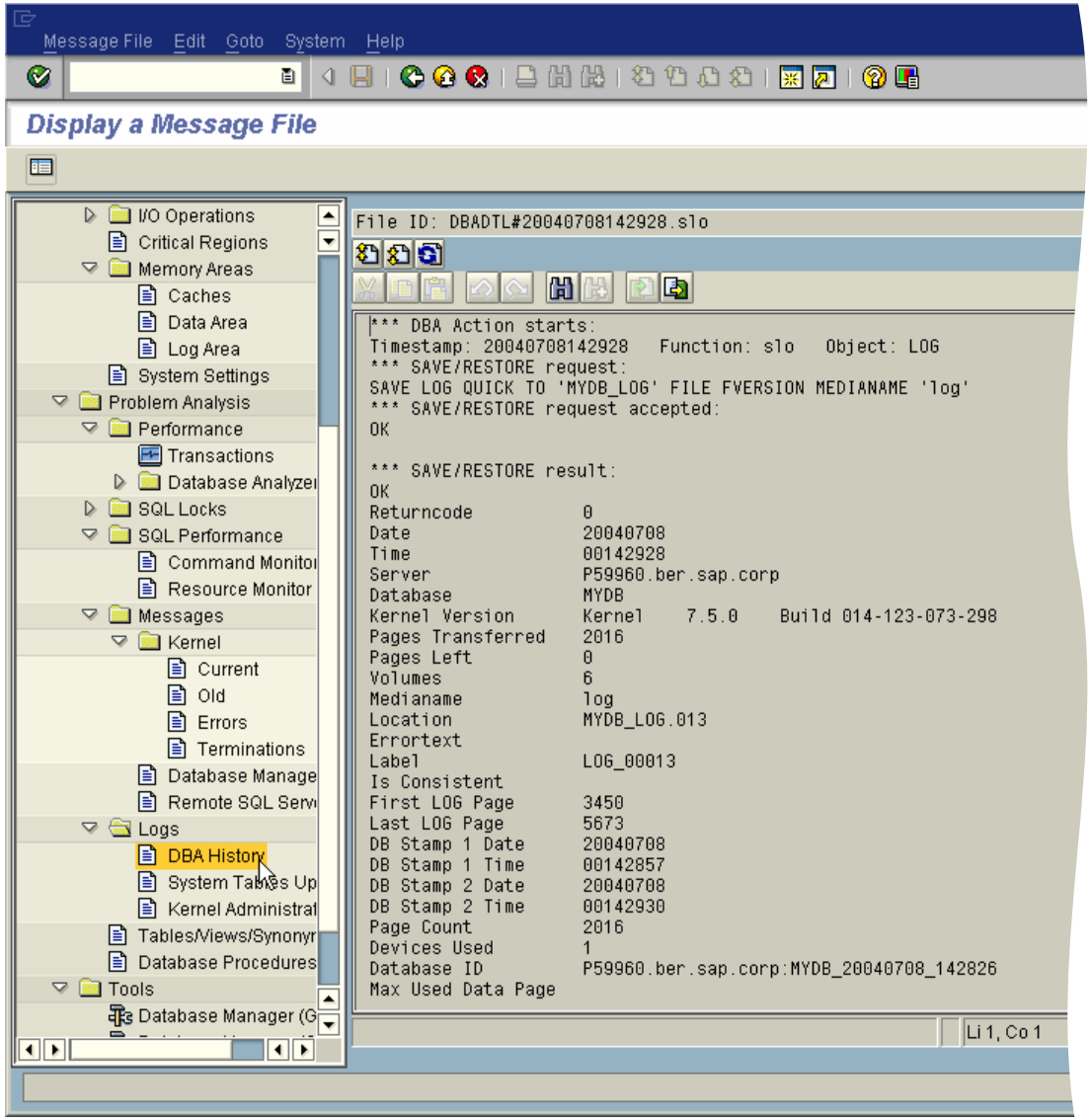
File xserver.prt has a fixed size. The first part of this file contains startup information of the X-Server, including an environment dump. There you can see the relevant environment variable settings of the user who started the X-Server. This part is not overwritten.
The second part contains runtime information like e.g. connect errors.

DBA History: Backup/Restore (DBMServer)



The DBA History contains information about administrative tasks.
E.g. you can see information about executed backup and recovery actions or consistency checks.
You can display a detailed log file for each of these actions.

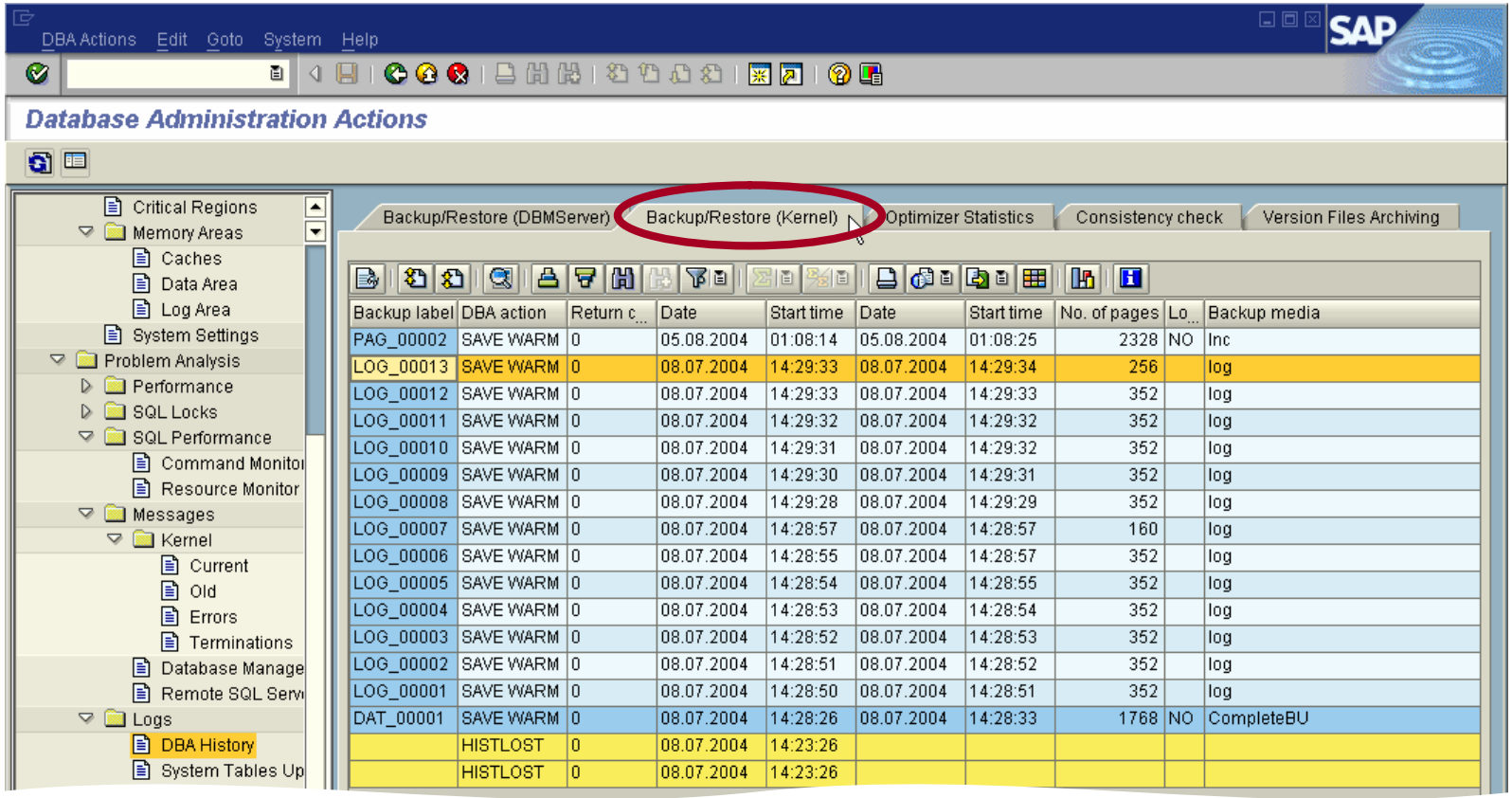
DBA History: Backup/Restore Log File



The log file of a backup contains the backup command and its return code and detailed information like:

- the creation date of the backup
- the number of pages transferred
- the backup label
- the location of the backup file

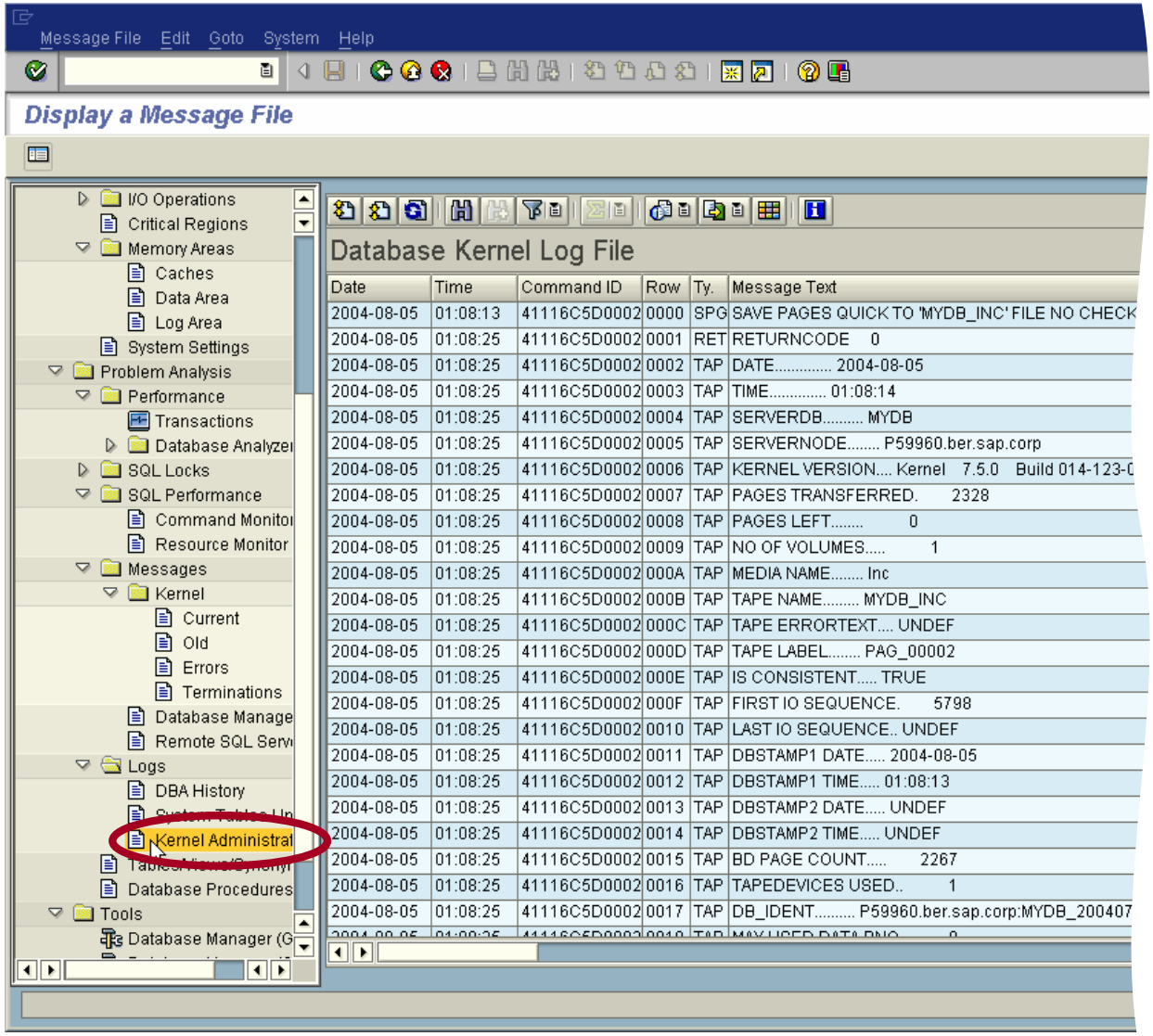
DBA History: Backup/Restore (Kernel)



Backup label	DBA action	Return c...	Date	Start time	Date	Start time	No. of pages	Lo...	Backup media
PAG_00002	SAVE WARM	0	05.08.2004	01:08:14	05.08.2004	01:08:25	2328	NO	Inc
LOG_00013	SAVE WARM	0	08.07.2004	14:29:33	08.07.2004	14:29:34	256		log
LOG_00012	SAVE WARM	0	08.07.2004	14:29:33	08.07.2004	14:29:33	352		log
LOG_00011	SAVE WARM	0	08.07.2004	14:29:32	08.07.2004	14:29:32	352		log
LOG_00010	SAVE WARM	0	08.07.2004	14:29:31	08.07.2004	14:29:32	352		log
LOG_00009	SAVE WARM	0	08.07.2004	14:29:30	08.07.2004	14:29:31	352		log
LOG_00008	SAVE WARM	0	08.07.2004	14:29:28	08.07.2004	14:29:29	352		log
LOG_00007	SAVE WARM	0	08.07.2004	14:28:57	08.07.2004	14:28:57	160		log
LOG_00006	SAVE WARM	0	08.07.2004	14:28:55	08.07.2004	14:28:57	352		log
LOG_00005	SAVE WARM	0	08.07.2004	14:28:54	08.07.2004	14:28:55	352		log
LOG_00004	SAVE WARM	0	08.07.2004	14:28:53	08.07.2004	14:28:54	352		log
LOG_00003	SAVE WARM	0	08.07.2004	14:28:52	08.07.2004	14:28:53	352		log
LOG_00002	SAVE WARM	0	08.07.2004	14:28:51	08.07.2004	14:28:52	352		log
LOG_00001	SAVE WARM	0	08.07.2004	14:28:50	08.07.2004	14:28:51	352		log
DAT_00001	SAVE WARM	0	08.07.2004	14:28:26	08.07.2004	14:28:33	1768	NO	CompleteBU
	HISTLOST	0	08.07.2004	14:23:26					
	HISTLOST	0	08.07.2004	14:23:26					

This is the backup history from the point of view of the database kernel.
Each log backup action might create several log backup files - each of the size of one log segment.
The HISTLOST entries are created whenever the log volumes are initialized - e.g. during the installation. Then you have to create a complete backup again to start a new backup history.

Kernel Administration Log File



File dbm.utl contains information about all administrative tasks sent to the database kernel.

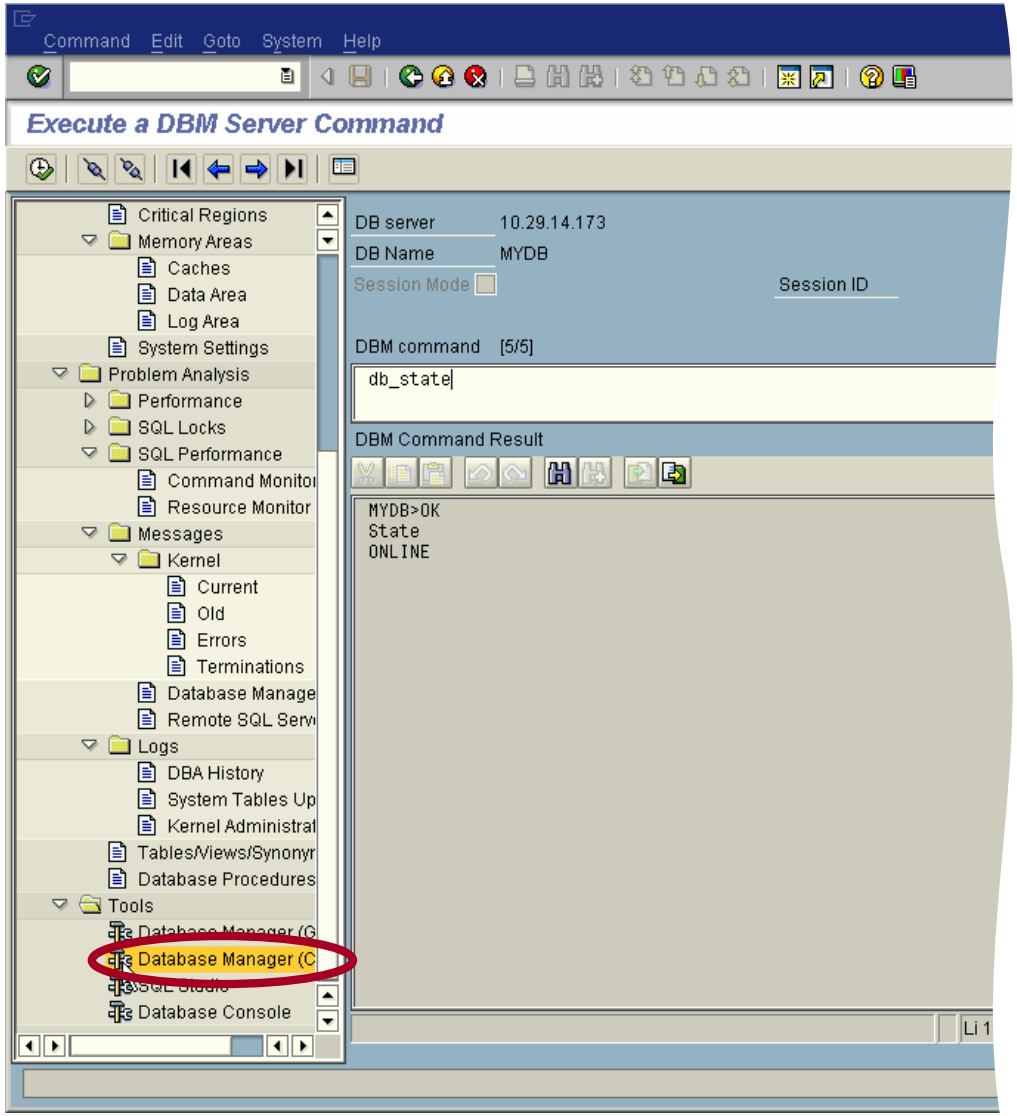
This log file is written by the database kernel itself.

It contains information about

- backups,
- consistency checks and
- starts/stops of the database instance

including the return codes of these commands.

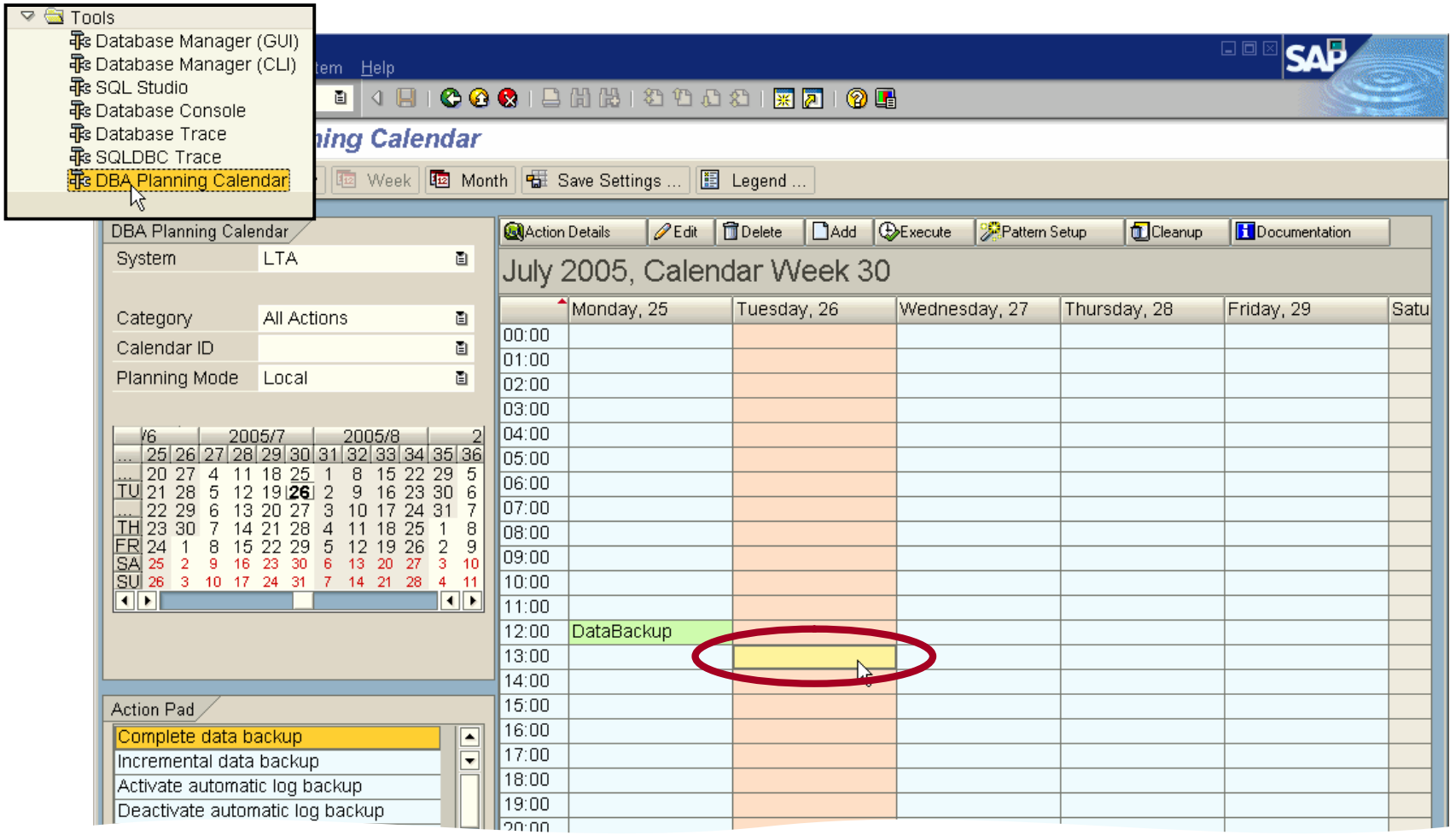
Database Manager (CLI)



Using the Database Manager CLI dbmserver commands can be executed.
Some of the most important commands are:

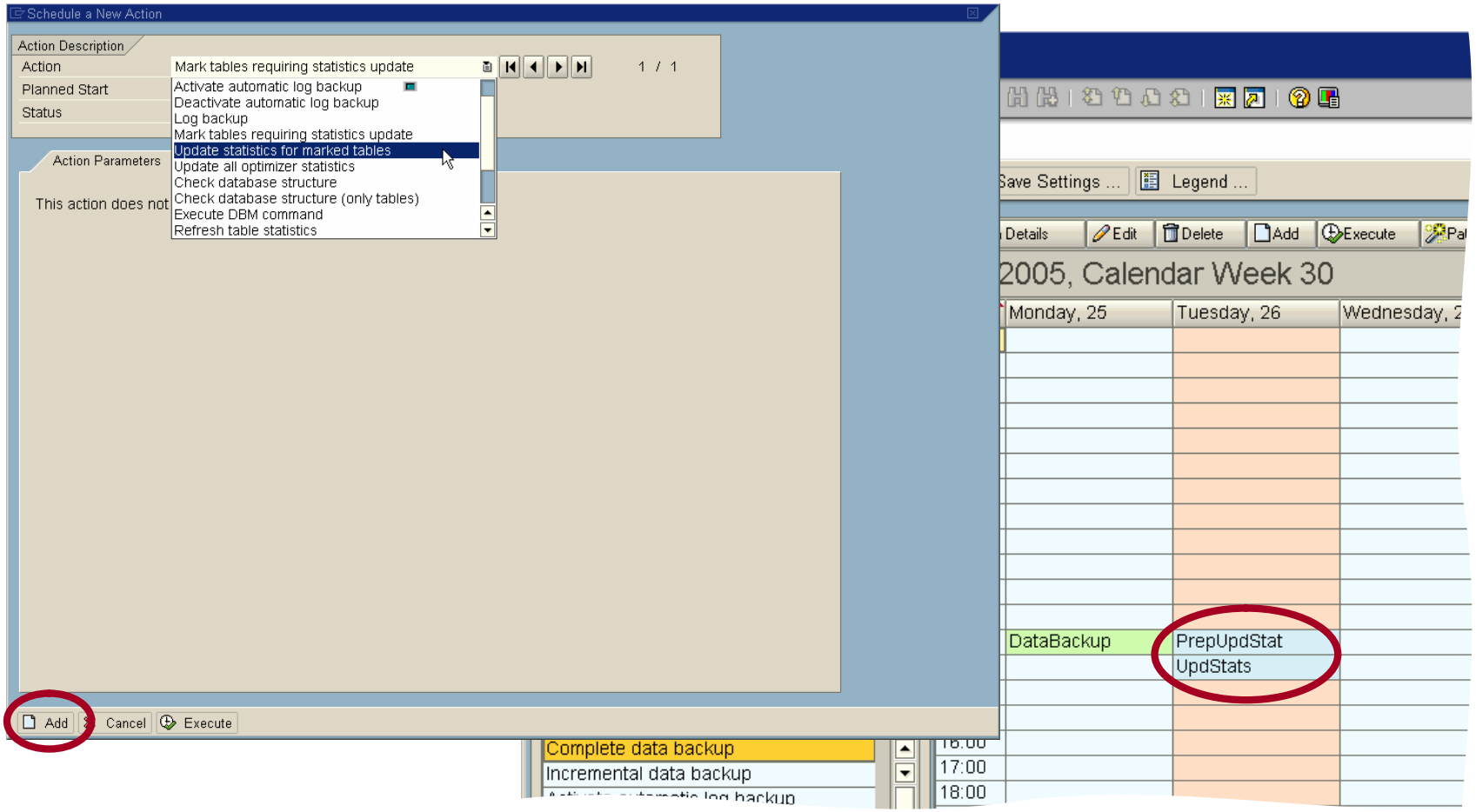
- db_state – determines the database state
- dbm_version – determines the version of the dbmserver
- db_offline – stops the database instance – should not be executed for the systems own database instance!
- db_online - starts the database instance

DBA Planning Calendar



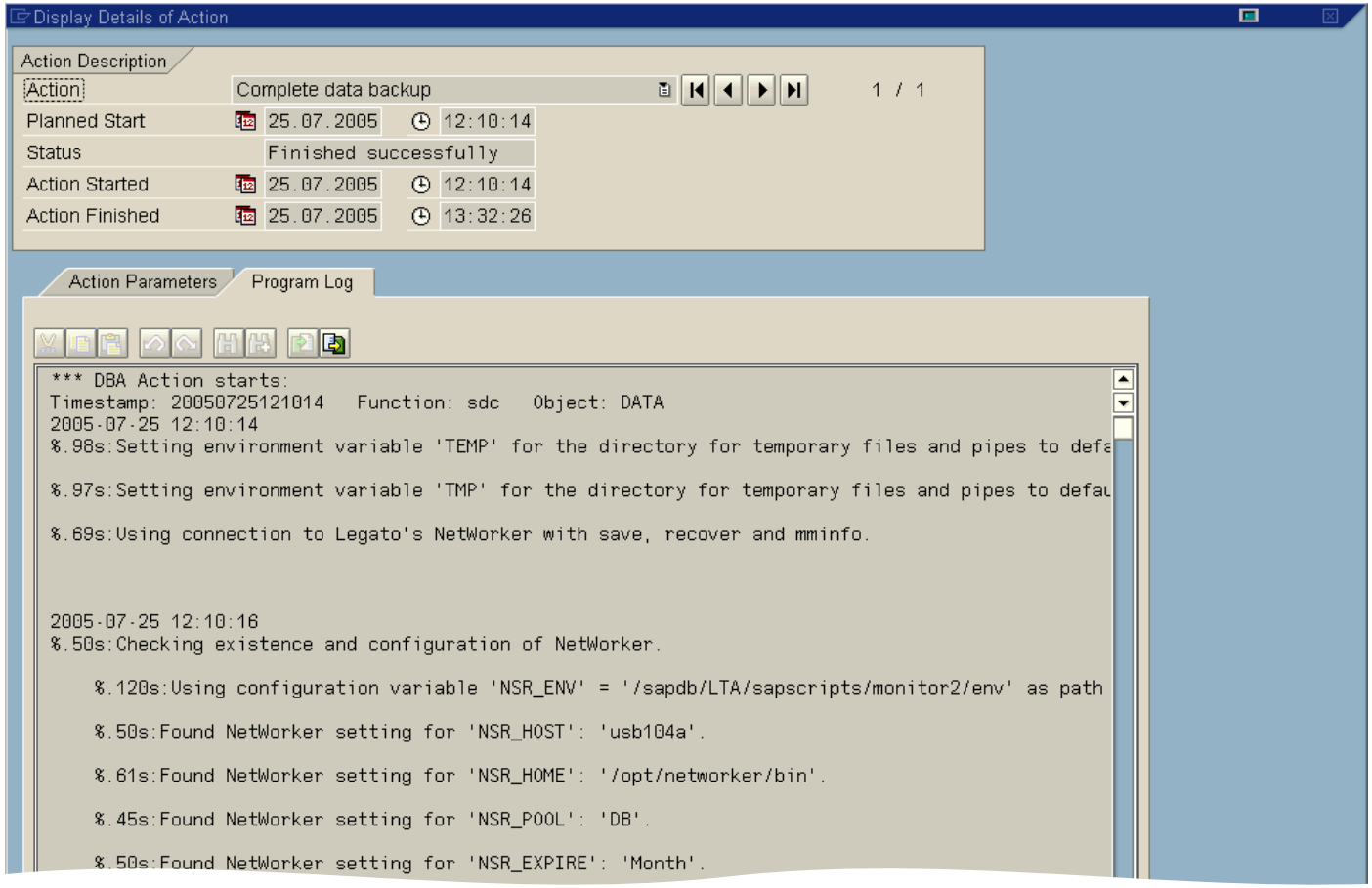
The DBA Planning Calendar allows to schedule important database tasks like backups or consistency checks (like transaction DB13/DB13C in earlier SAP releases). Double-click a line in the calendar view or in the Action Pad to schedule a task.

Scheduling Tasks



For some tasks parameters are necessary – e.g the backup medium for a backup. These parameters can be specified in this window. Furthermore you can specify if the task should be executed immediately or if it should be executed at a specific date/time and if it should be executed in a certain interval (daily, weekly, every few hours, …)

DBA Planning Calendar



If you perform a double-click on a finished action you can have a look at the log files of this task.

Chapter

What Has to be Monitored?

Standby Database

Central Monitoring – Transaction DB59

Database Assistant – Transaction DB50

Performance Analysis Tools

Alert Monitor – Transaction RZ20

Snapshot

Performance Analysis

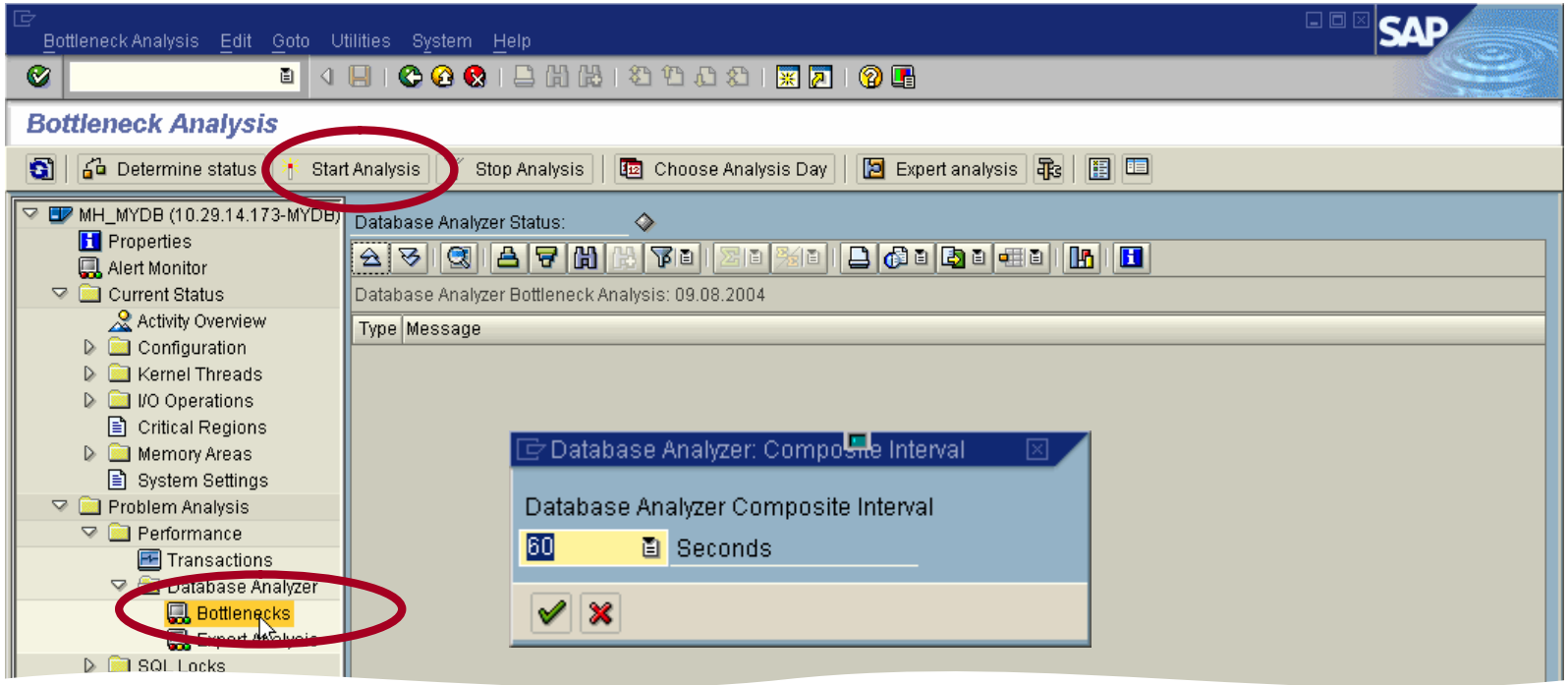
If you have performance problems you can start the MaxDB performance analysis tools

- Database Analyzer
- Command Monitor
- Resource Monitor

In case you have problems with special transactions you can run these afterwards and analyze the collected data concerning these transactions.

In case of general performance problems the tools should run for a while and you should check the output of these tools regularly for any problems.

Starting the Database Analyzer

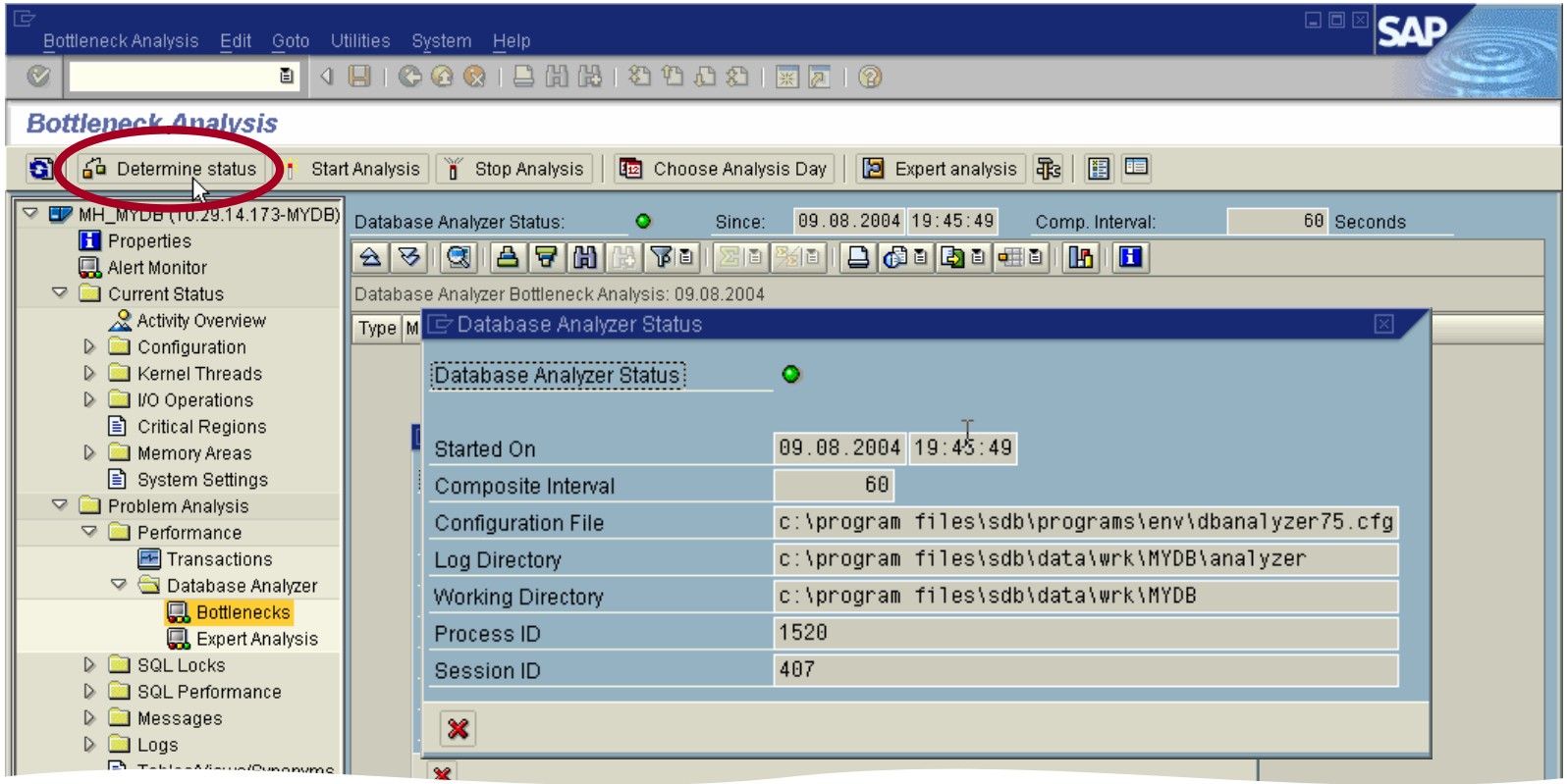


The database analyzer is a rule-based expert system for performance analysis.
It collects statistical and monitoring data as well as system messages. It detects and reports e.g.

- Low cache hit ratio
- High I/O load
- Low hit ratio of DML commands (SELECT, UPDATE, DELETE)

For a short time analysis the interval should be set to 60 - 120 seconds.
For long time monitoring it should be set to 900 seconds.

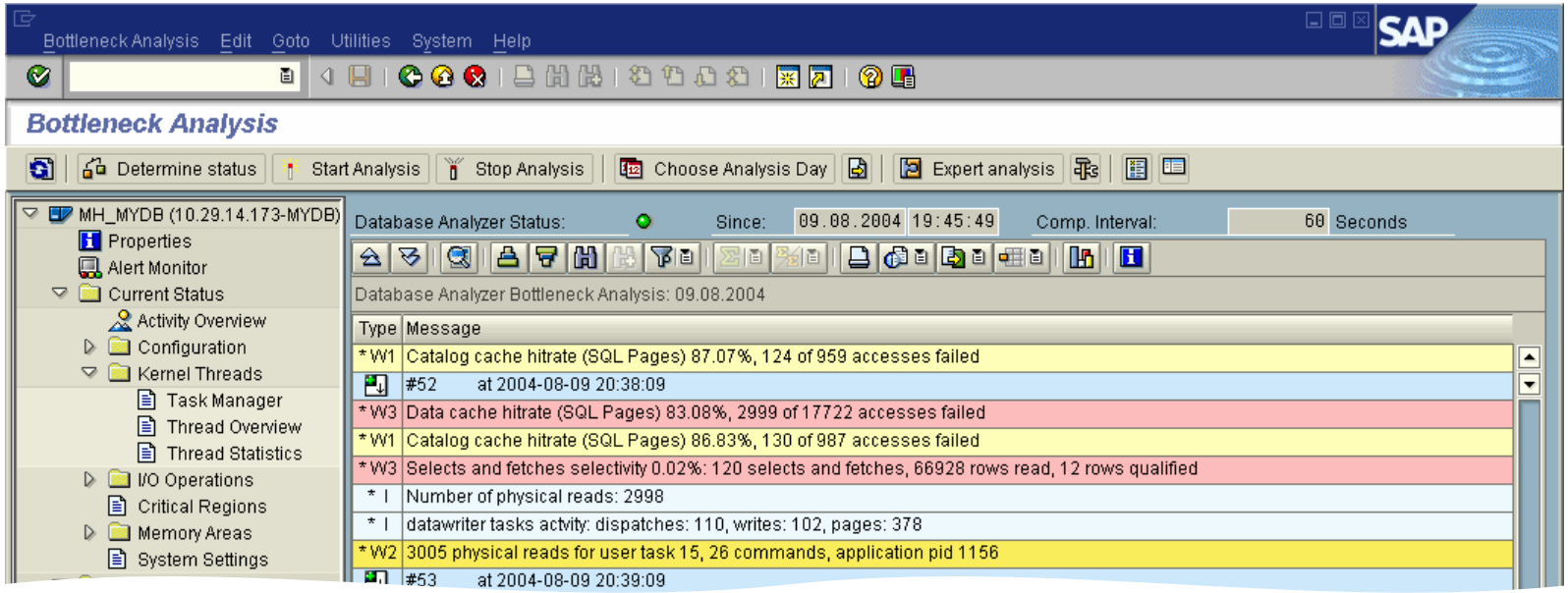
Database Analyzer Status



Database Analyzer uses a set of sophisticated rules to classify the current state of MaxDB by analyzing several MaxDB parameters.
These rules and the source of collected data are delivered in file dbanalyzer.cfg.
In the status window you can see which configuration file is used and where the log files are stored.

The log files of each day are stored in a separate directory where you can analyze them later on.

Database Analyzer Output



The Database Analyzer rates the information and bottlenecks:

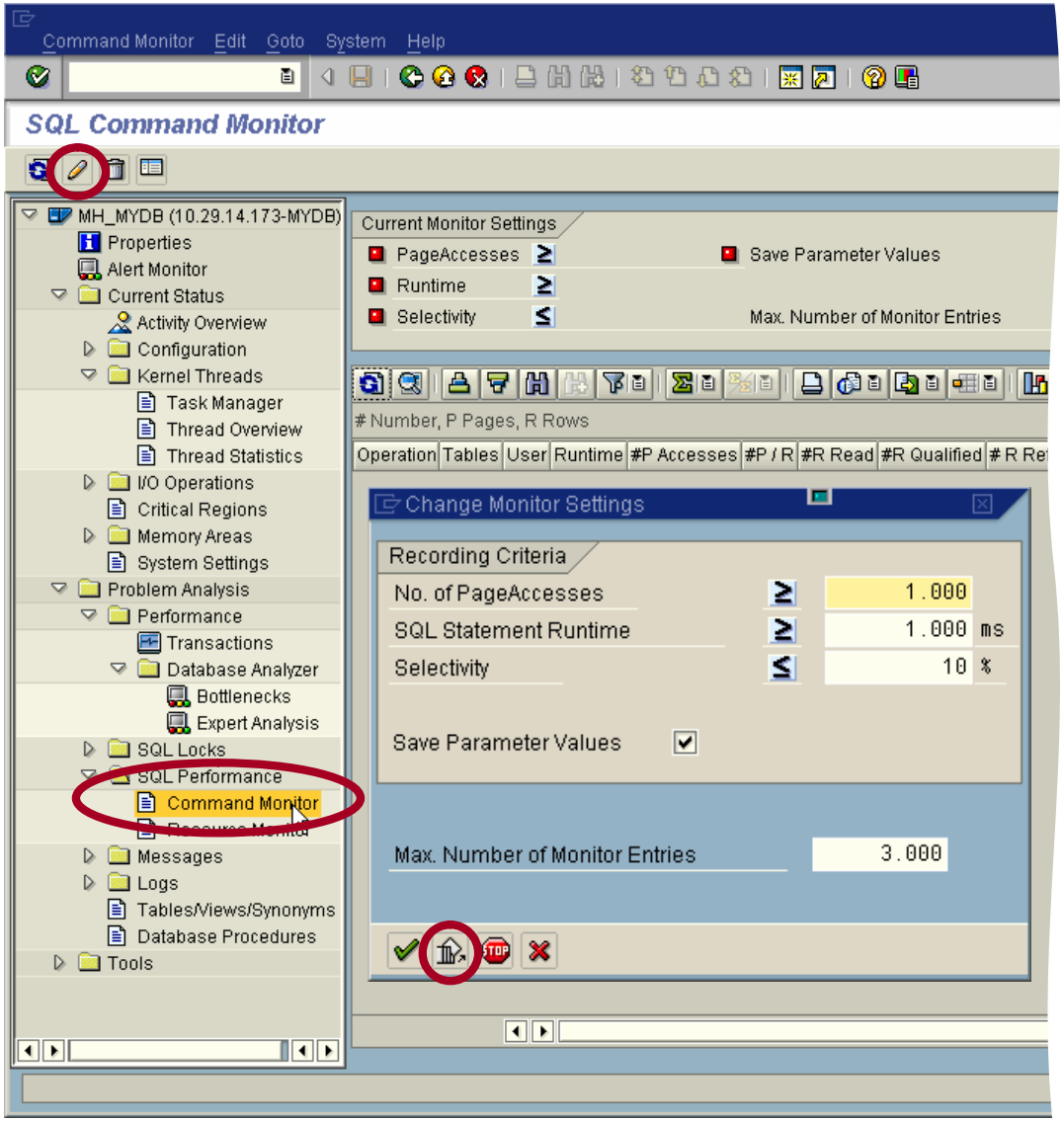
I: General information, such as the number of executed commands

W1 to W3: Bottleneck warnings with low, medium, and high priority

An example for a warning might be
W3 Selects and fetches selectivity 0,02% -> rows read 66928, 12 rows qualified

That means that the access strategies to data in SQL tables is bad because a high number of table rows have to be read internally to find a small number of rows that meet the qualification in the WHERE clause.

Starting The Command Monitor



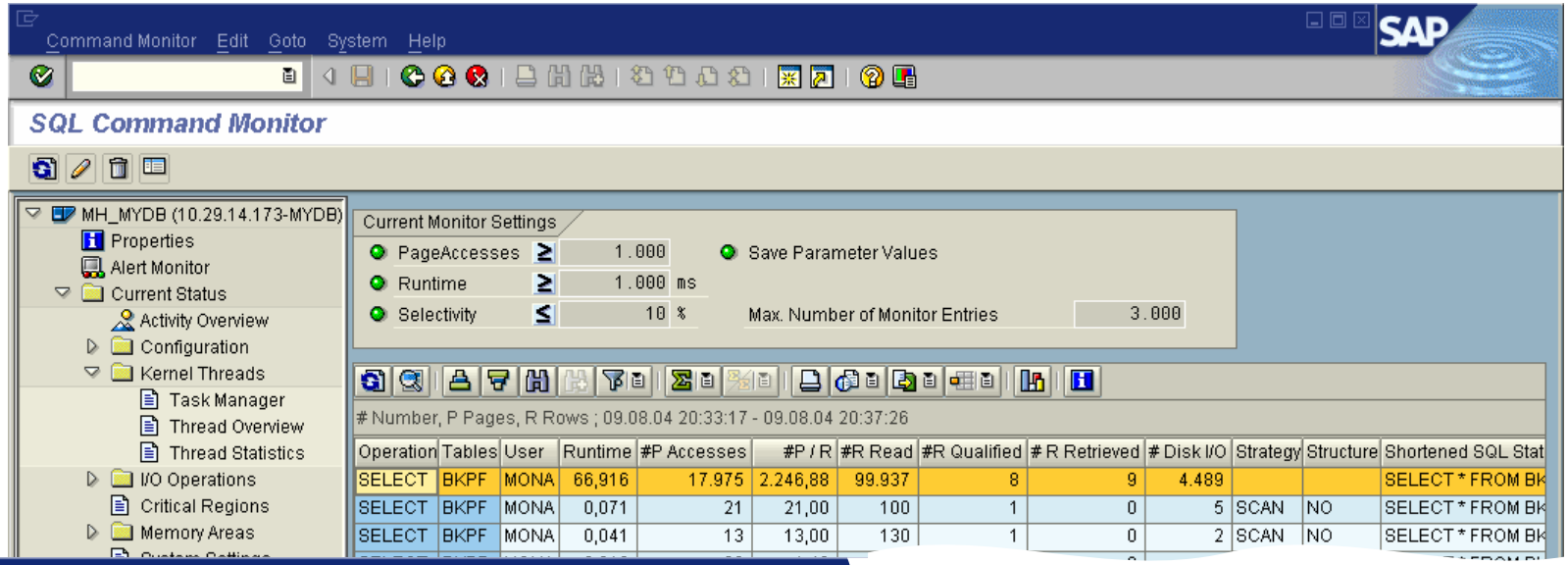
To identify long running statements or statements with a bad selectivity, the command monitor can be used. It collects specific data about SQL statements whose resource consumption violates configurable thresholds like runtime, page accesses or selectivity.

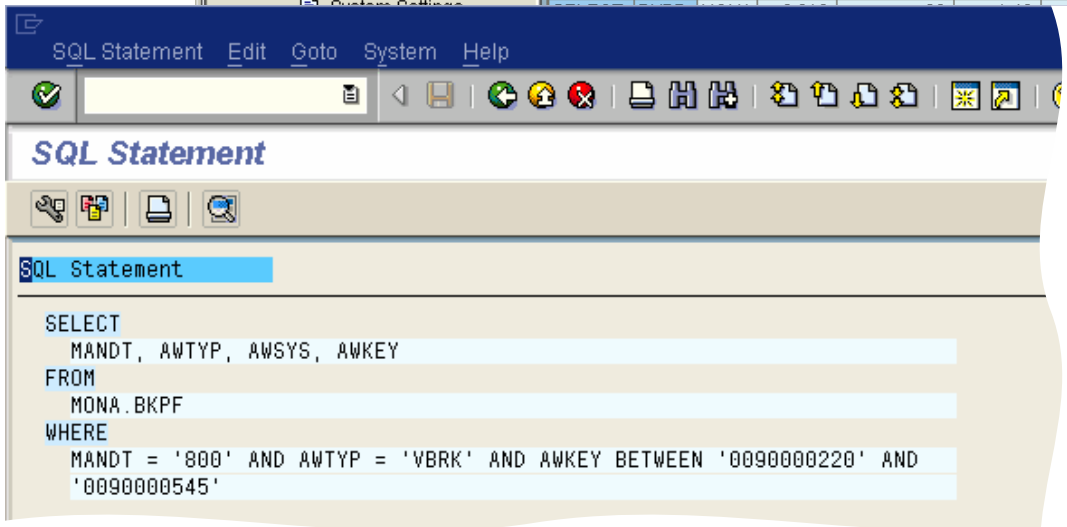
This monitor is mainly used to catch statements with high individual runtime.

The command monitor also collects the exact user input data used during statement execution.
This is essential to create the correct execution plan used for statement execution.

The command monitor keeps only a specified number of statements - old statements are overwritten when this number is reached.

Command Monitor Output



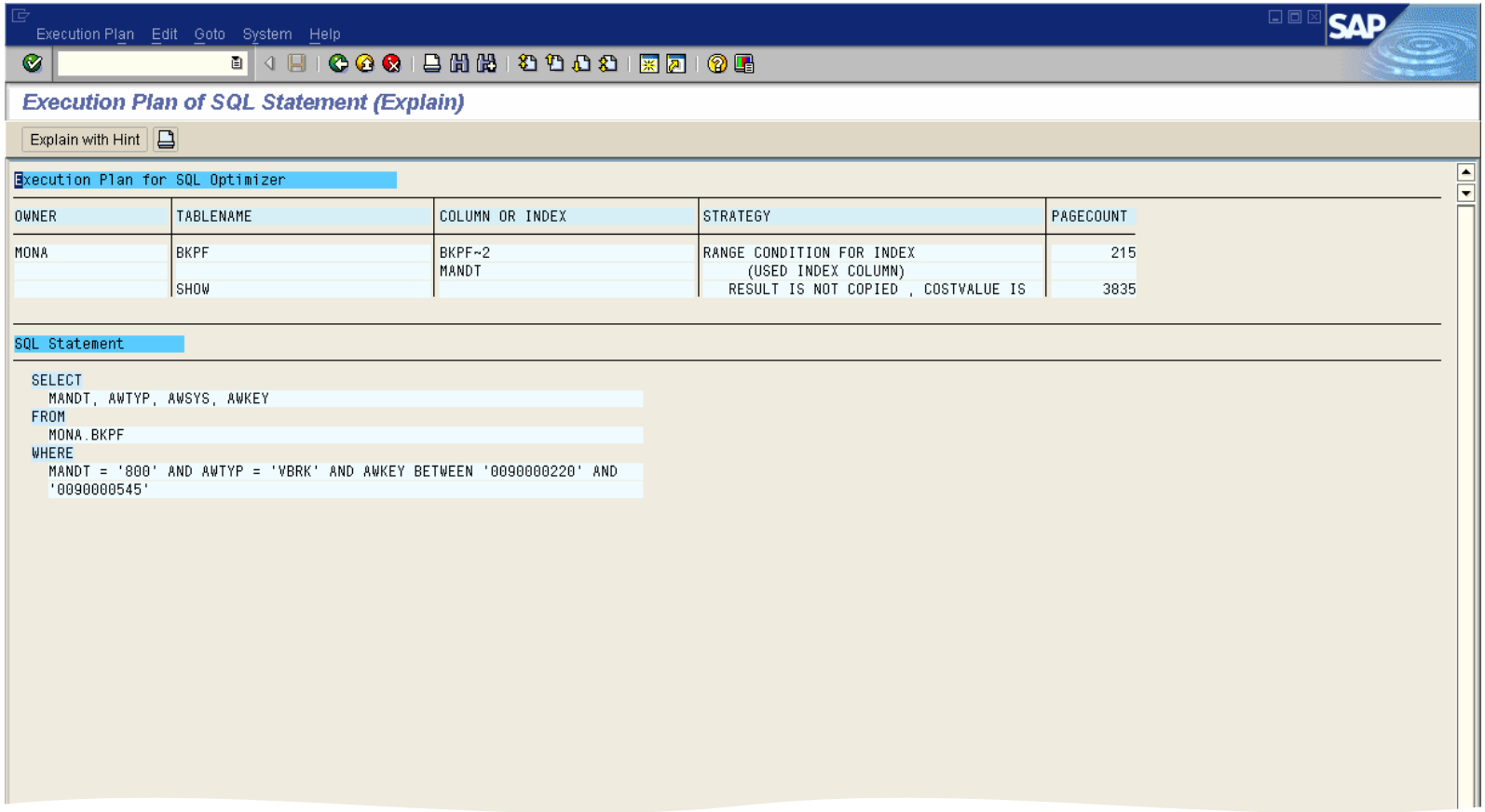


In the command monitor you can see e.g.

- the runtime of the statement,
- the number of rows read and rows qualified and
- the number of disk I/O during statement execution.

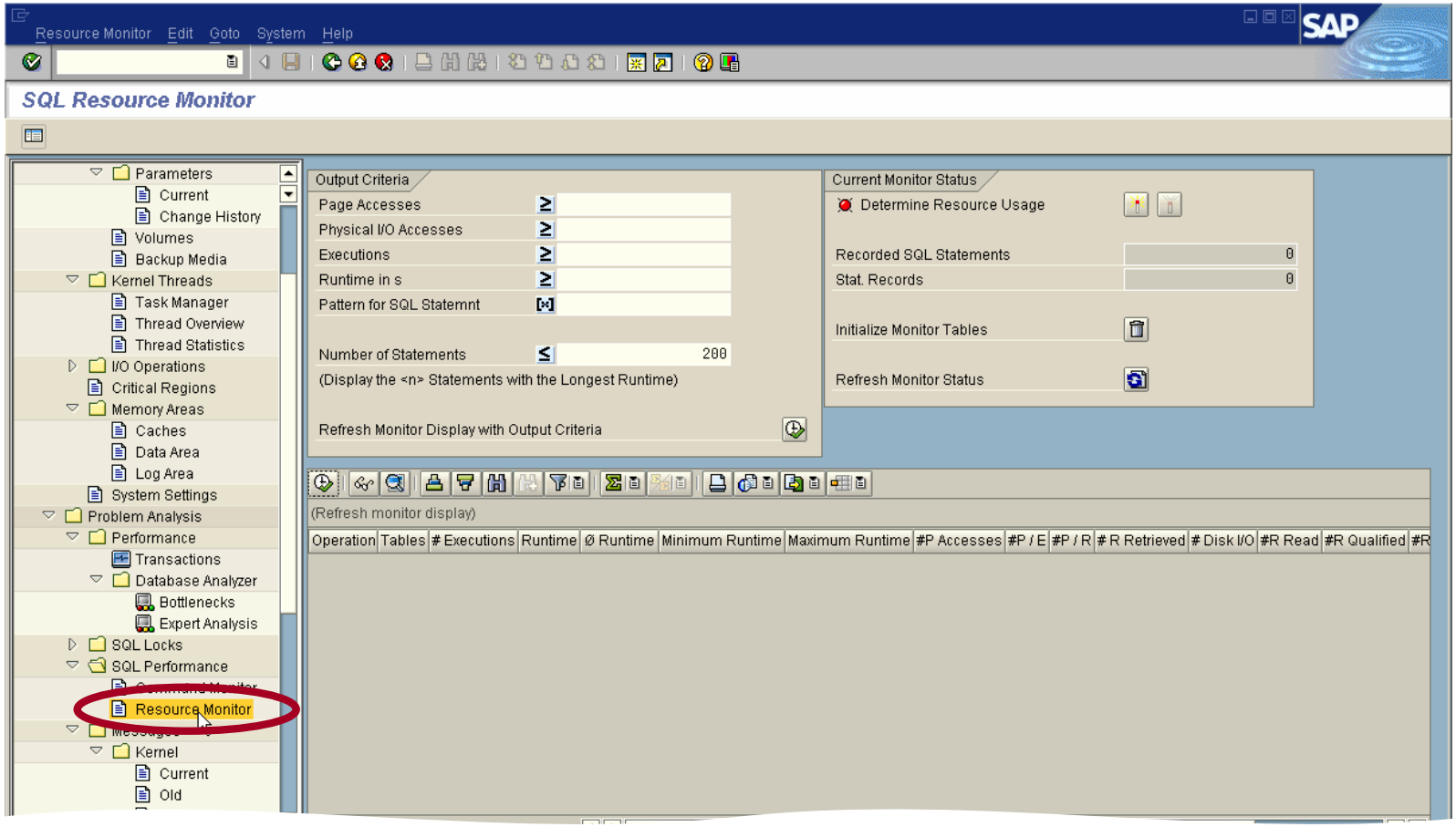
To view the complete statement perform a double click on the corresponding entry.

Execution Plan



The execution plan of a SQL statement can be displayed by pressing *Display Execution Plan for SQL Statement*.
In this example the optimizer uses a table scan to get the requested results.
A lot of rows have to be read to find the few matching rows.
That's the explanation for the entries in the database analyzer log file and in the command monitor.

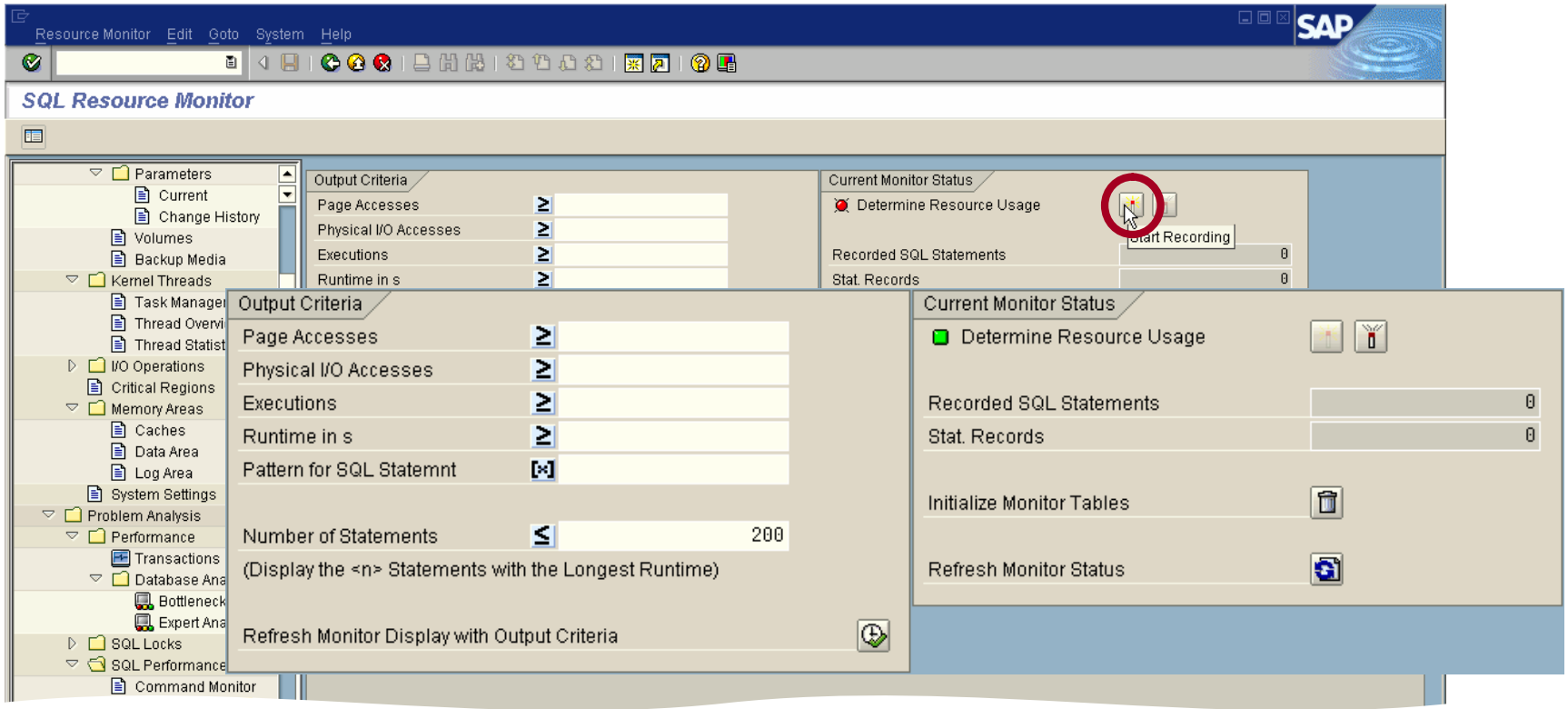
Resource Monitor



The resource monitor collects data of all executed statements independent of the single execution time.

You can restrict the number of displayed rows specifying lower limits e.g. for the runtime or the number of statement executions.

Starting the Resource Monitor

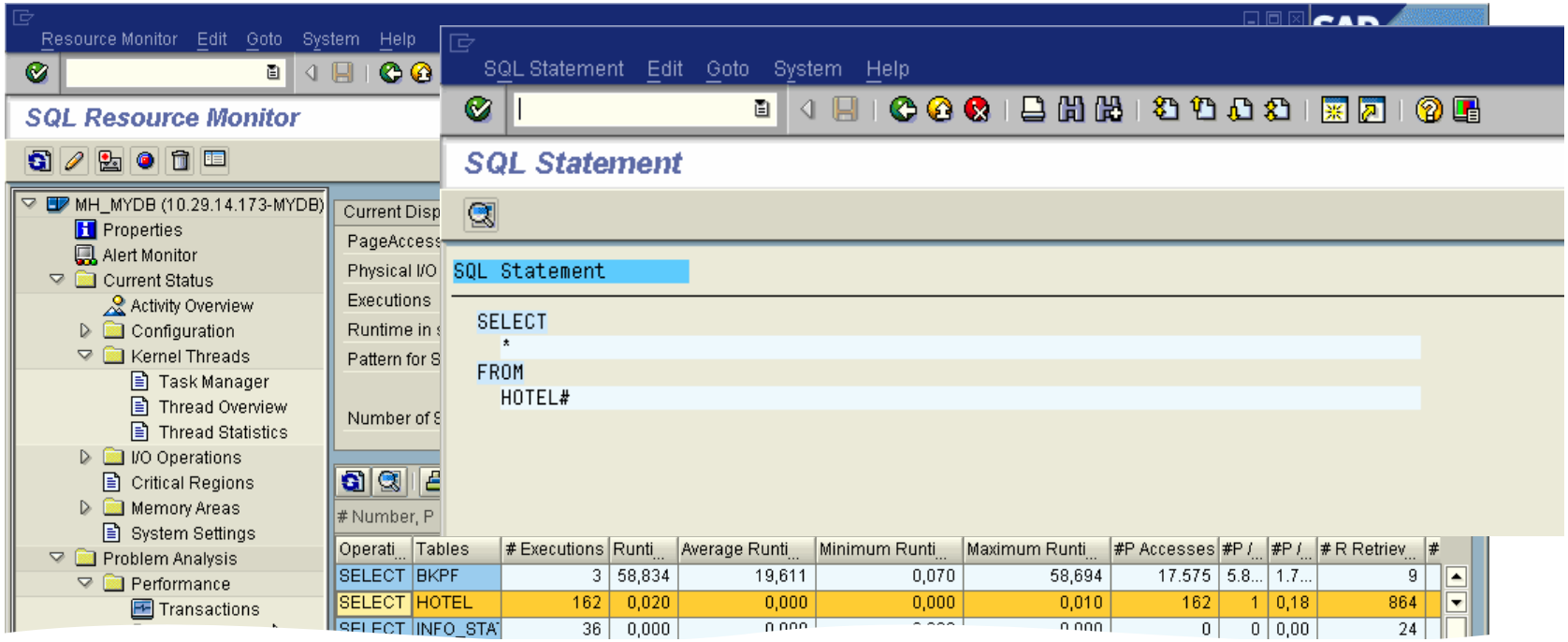


The resource monitor aggregates the resource consumption over all executions of a statement.

It helps to identify SQL statements with cheap individual execution (e.g. through primary key access), which are executed very often and therefore cause a high aggregated runtime and workload.

The optimization of these statements promises the highest overall effect.

Resource Monitor Output

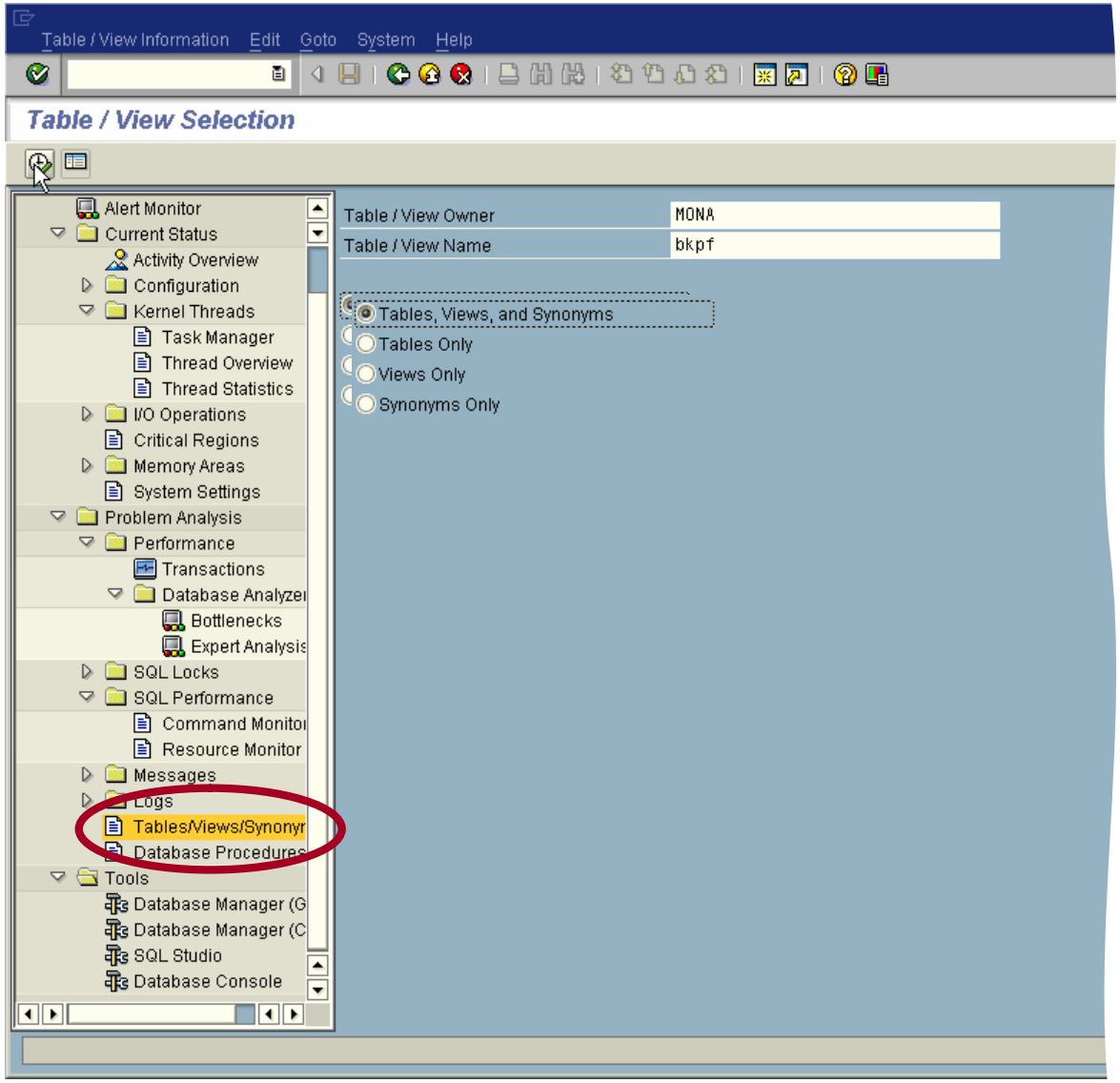


You can see e.g.

- the number of executions,
- the overall runtime and
- the number of page accesses.

To view the statement perform a double click on the corresponding entry.
Often these statements cannot be optimized with database methods because they are already executed in the most efficient way. Then the application developer has to check if the statement has to be executed so often.

Tables/Views



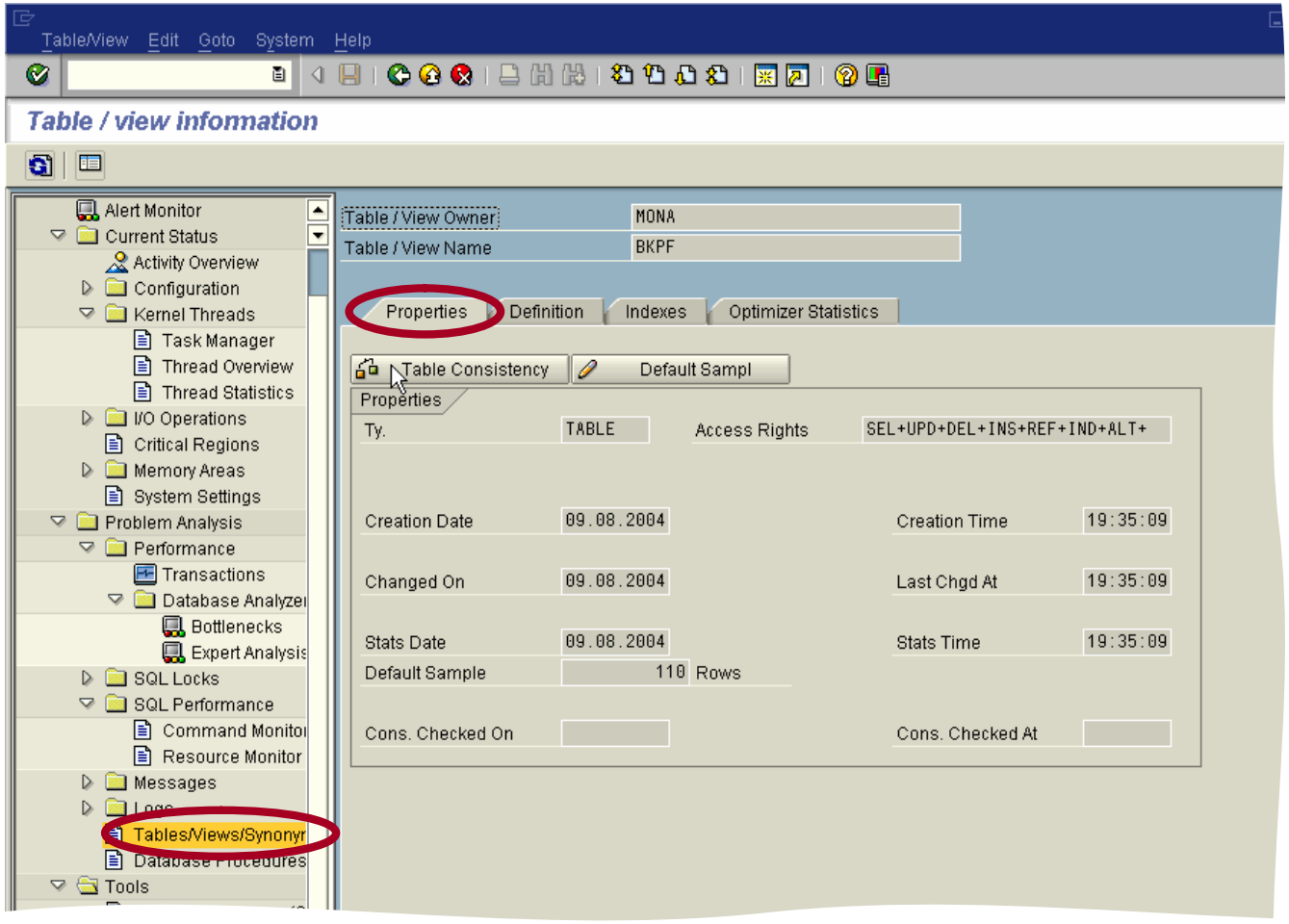
To determine the best optimizer strategy for an SQL statement you have to analyze all tables involved.

You have to check

- the table definition (especially the primary key definition)
- the existing indexes and
- the optimizer statistics.

To decrease the runtime of a statement it might be necessary to update the optimizer statistics or to create a new index.

Tables/Views: Properties

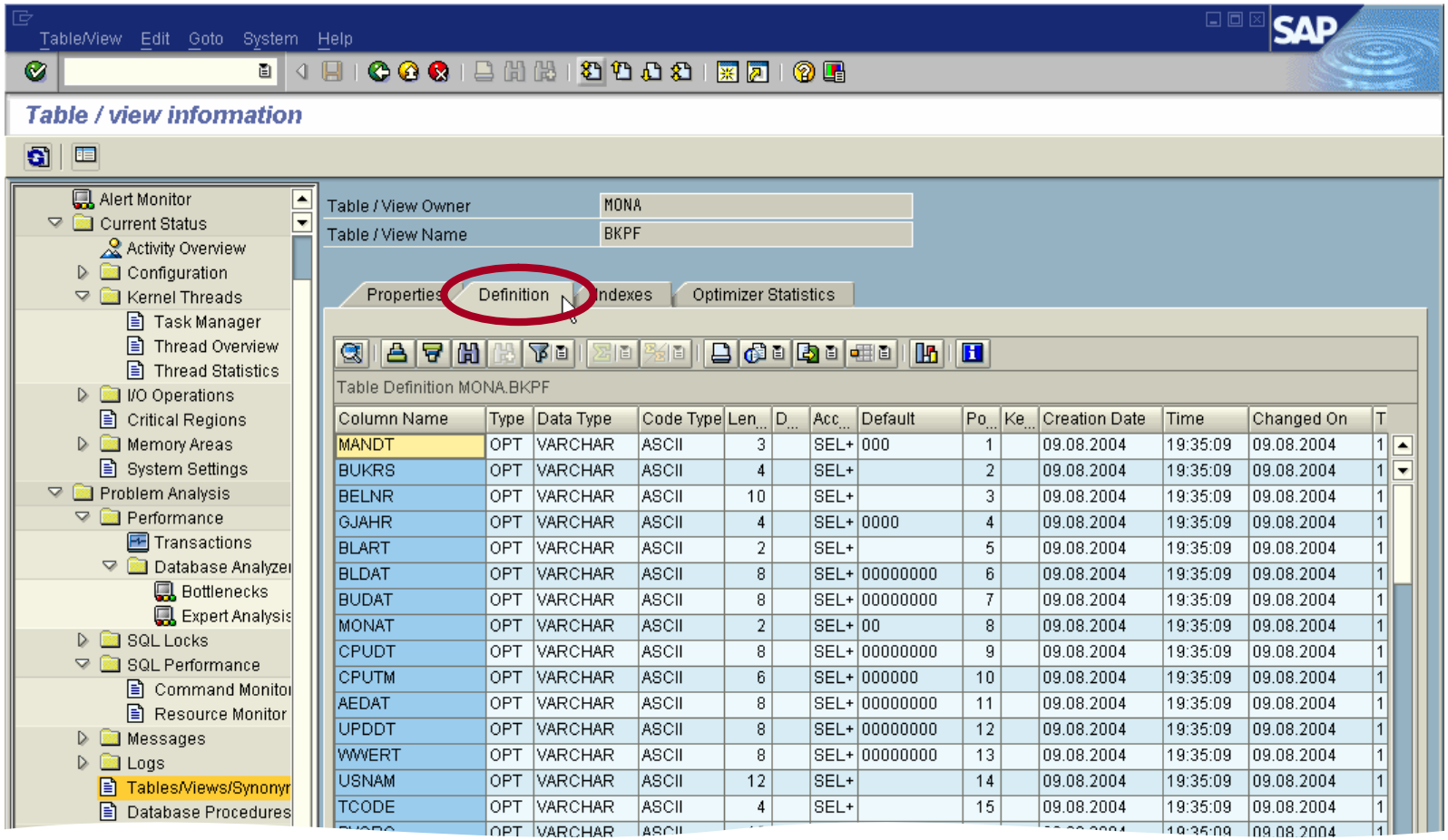


In the *Properties* section you can see when the table has been created and altered and when the optimizer statistics have been updated the last time.

It is also possible to check the table consistency and to change the default sample value for the creation of the optimizer statistics.

For large tables the sample value should be set to 10% -
for smaller tables a sample value of 20000 rows is sufficient (note #808060).

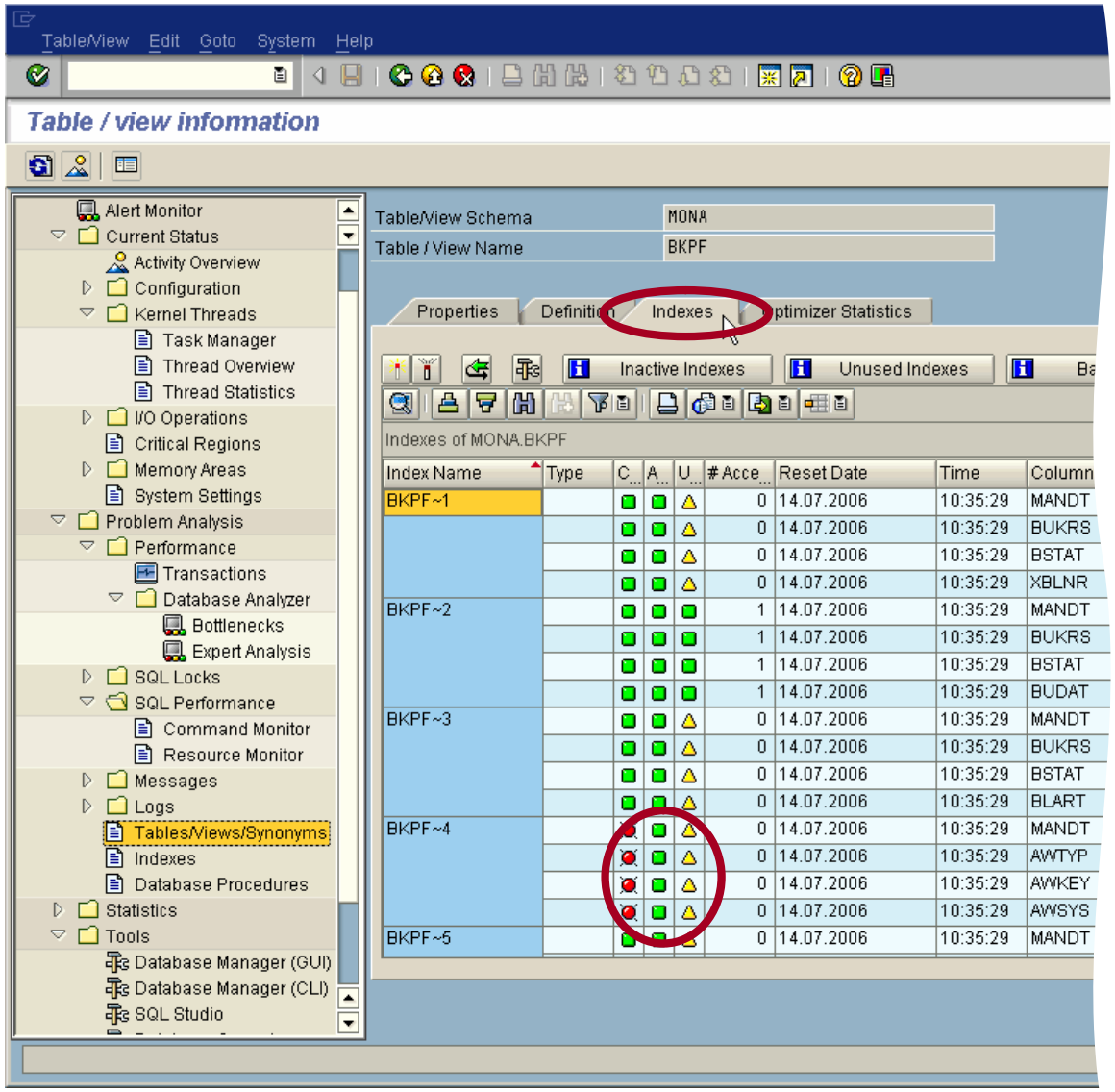
Tables/Views: Definition



Primary Key columns have the *Type* KEY, in column *Key Position* you can see if this is the first, second, … key column.

Furthermore you can see the data type and length of a column.

Tables/Views: Indexes



All Indexes defined for a table are listed in the *Indexes* section. You can see

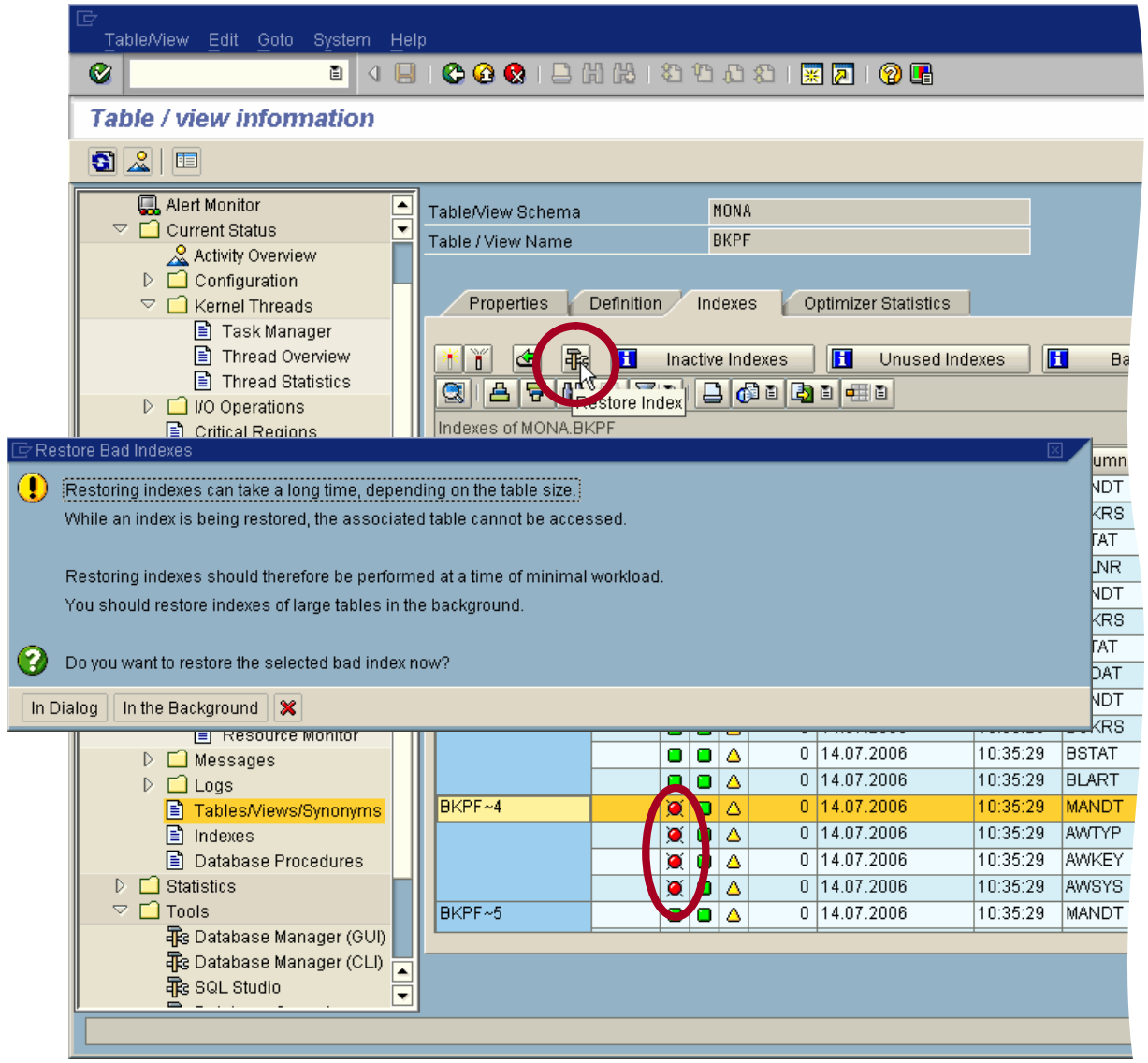
- of which columns the index consists,
- if it has already been used,
- if the index is activated,
- if the index is corrupted.

It is also possible to activate and deactivate indexes.

If an index is deactivated it is still maintained during insert, update or delete operations but it cannot be used to access the data.

Corrupted indexes can be recreated.

Tables/Views: Restore Index

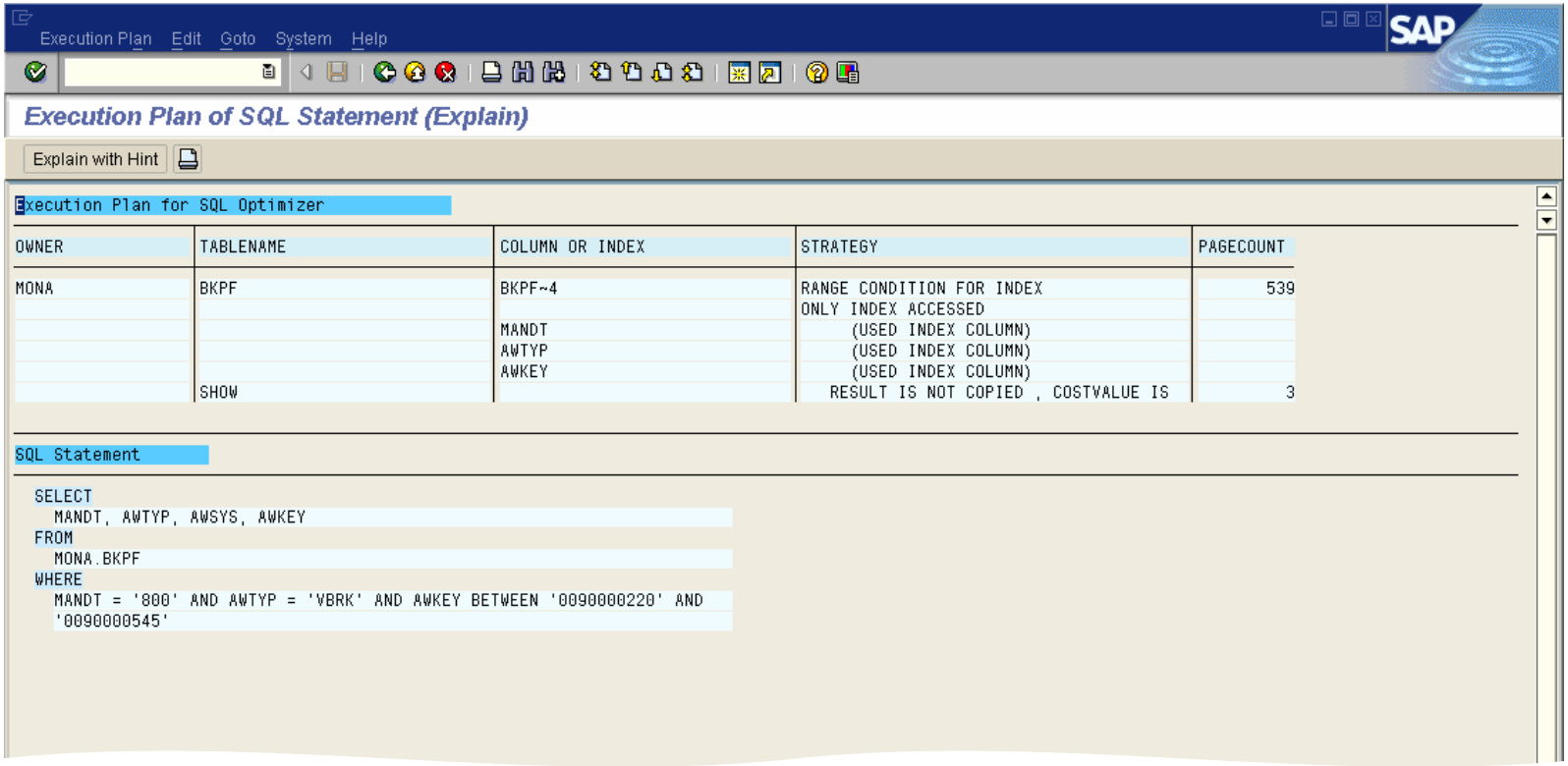


If an index is corrupted it cannot be used to access the data.

Mark the corrupted index and choose *Restore Index* to recreate it.

Attention: During the index rebuild the corresponding table is locked for write transactions. Depending on the size of the table the rebuild might take a long time.

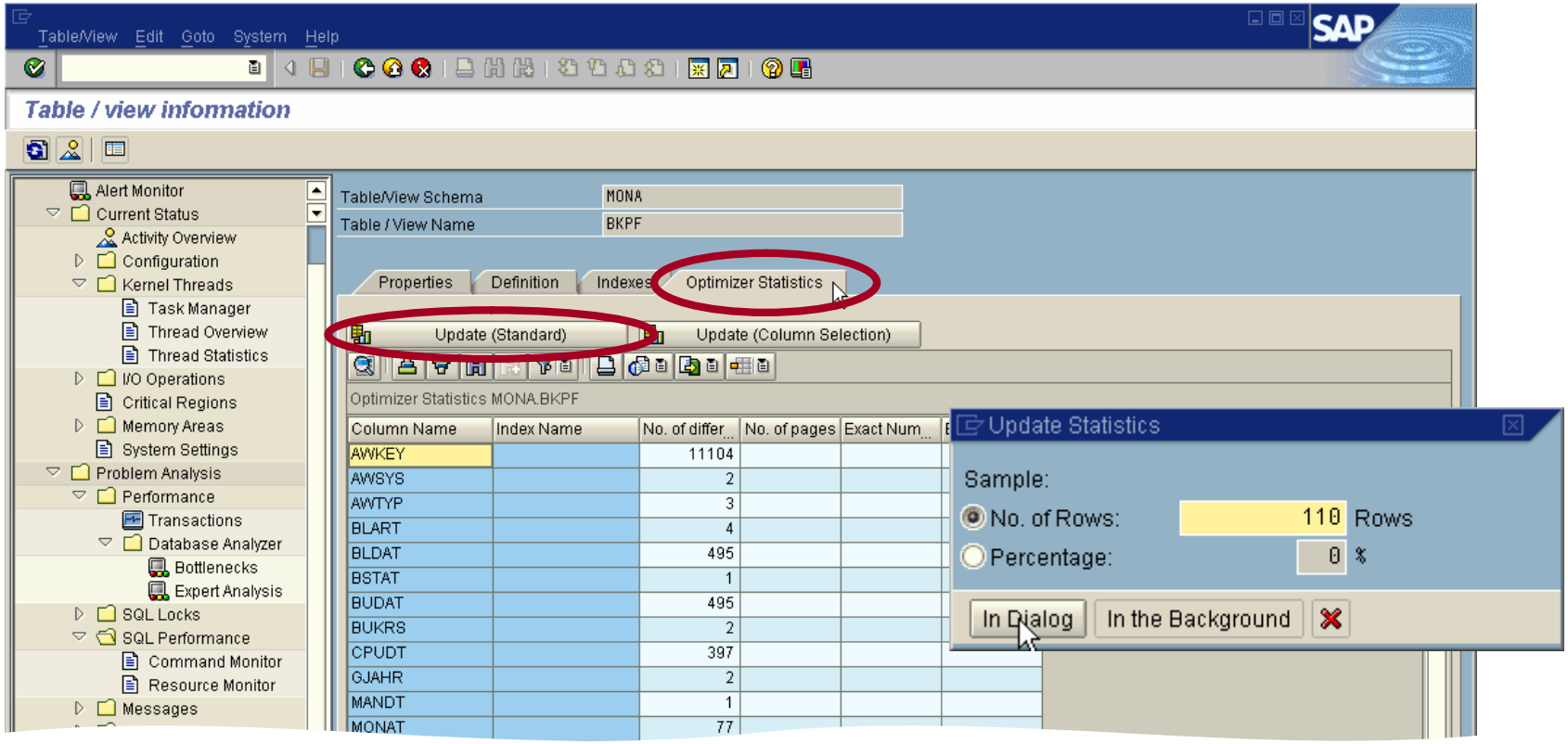
Showing the Execution Plan



After the index has been recreated it is used by the optimizer when the statement found in the command monitor is executed again.

Now the execution of this statement is much faster.

Tables/Views: Optimizer Statistics



To determine the best access strategy for a JOIN the optimizer needs statistical data. If this data is not updated regularly the optimizer might not choose the best strategy.

UPDATE STATISTICS determines information about the size and the value distribution of tables and indexes. These values are not counted but estimated on the basis of sample rows.

Chapter

What Has to be Monitored?

Standby Database

Central Monitoring – Transaction DB59

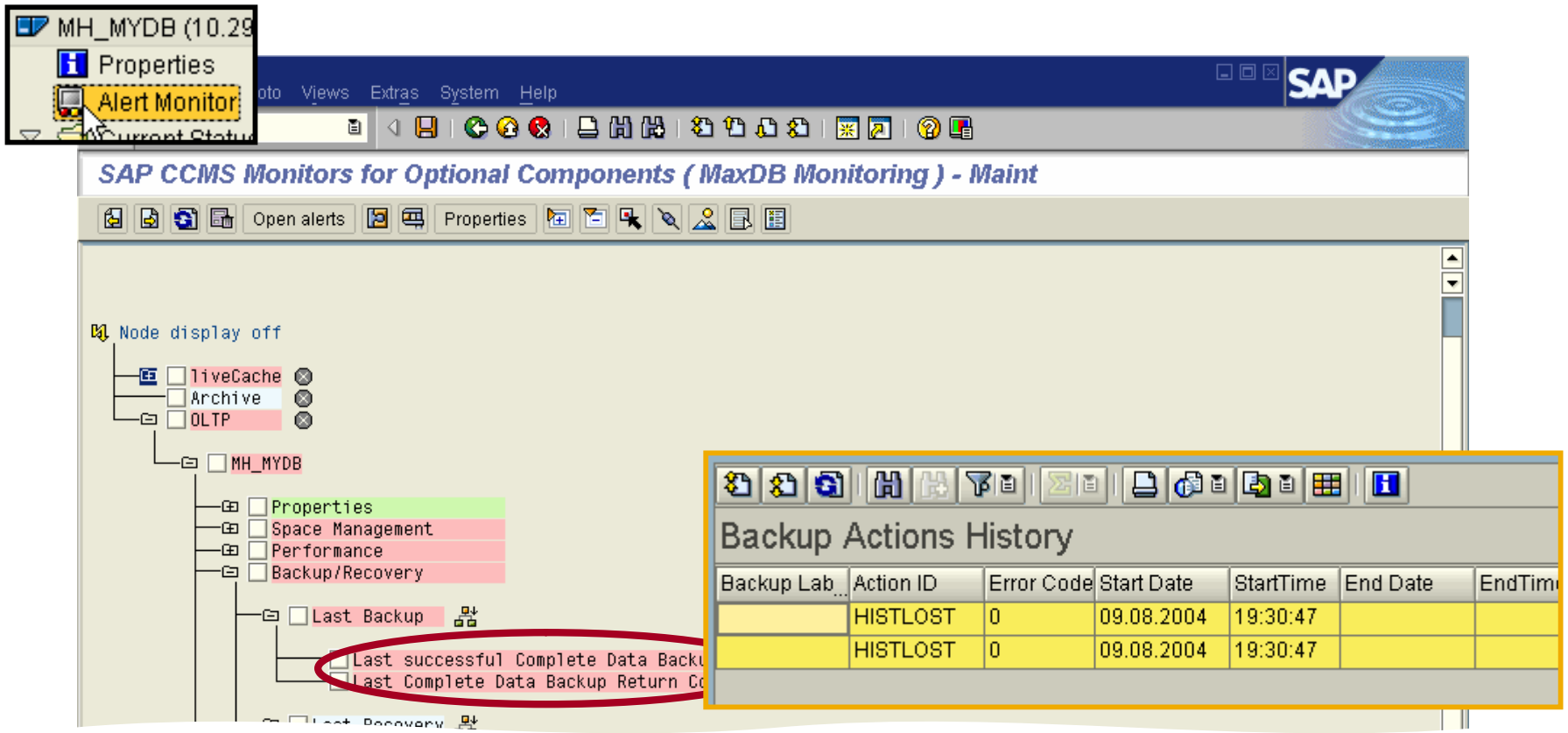
Database Assistant – Transaction DB50

Performance Analysis Tools

Alert Monitor – Transaction RZ20

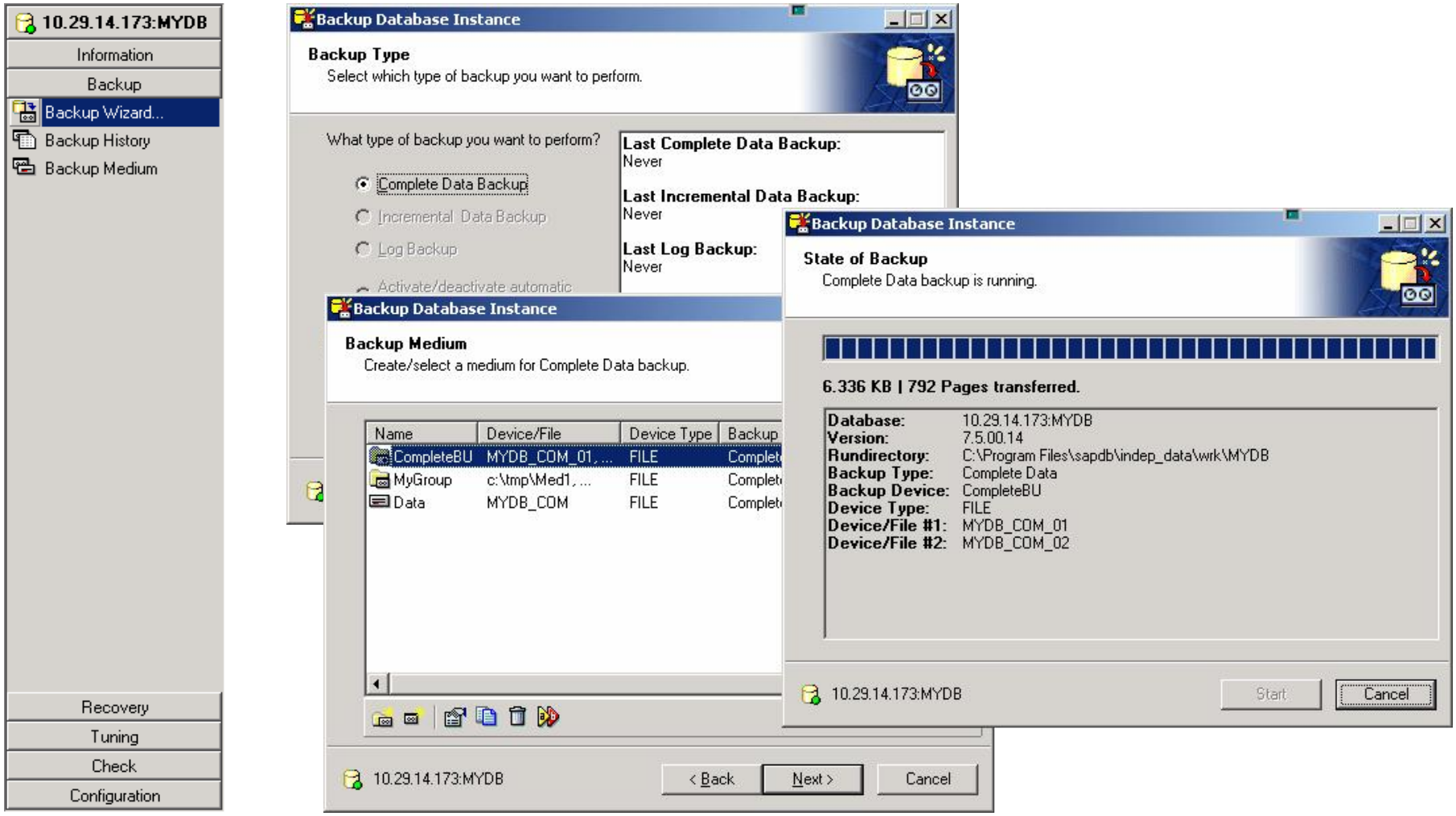
Snapshot

Alert Monitor – RZ20



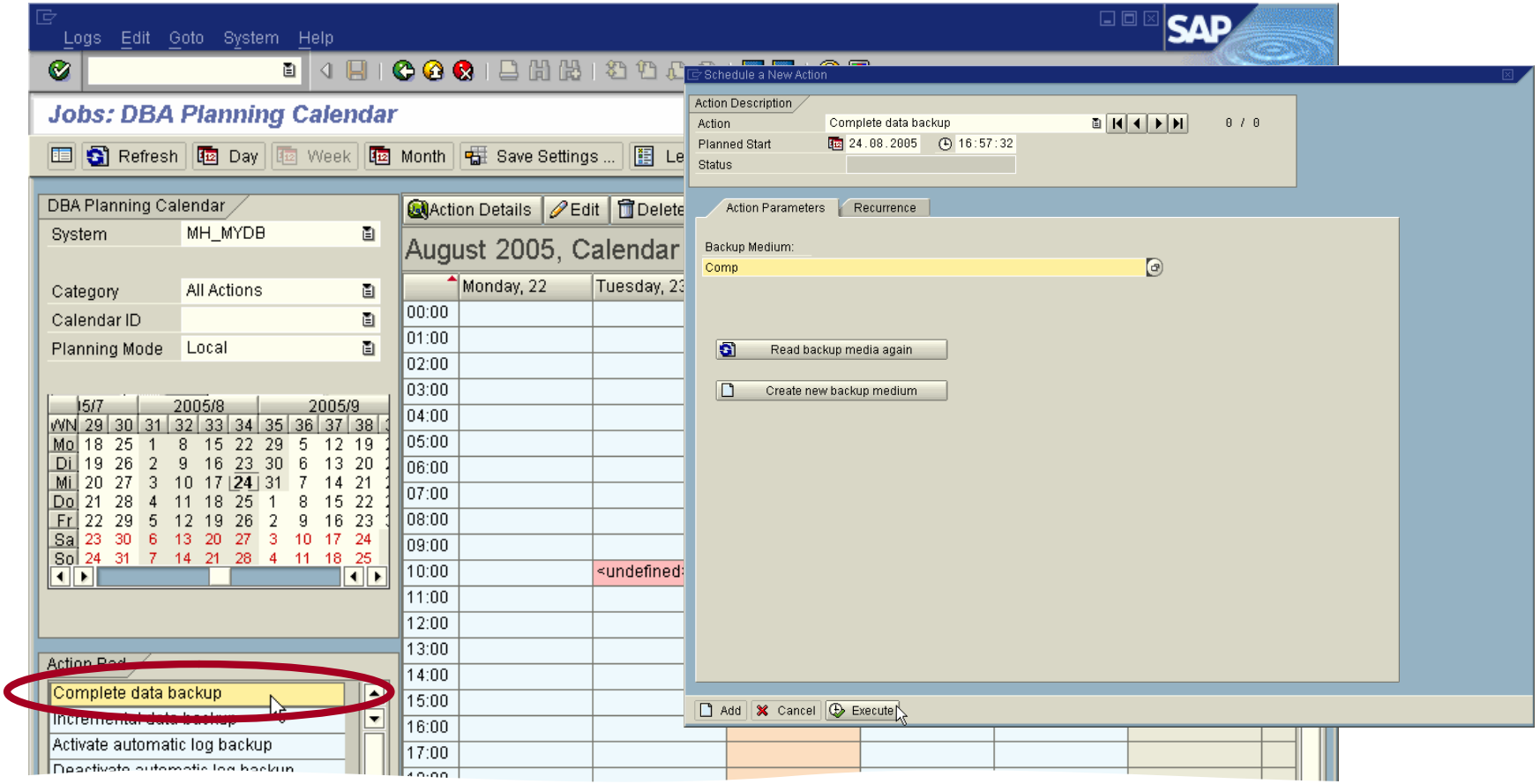
The alert monitor collects e.g. data concerning the filling level of the log and the data area, the cache hit ratio and the creation of backups. If you perform a double-click on a node in the tree, you get detailed information about this node - e.g. the backup history. In this example no data backup exists.

Creating a Complete Backup I



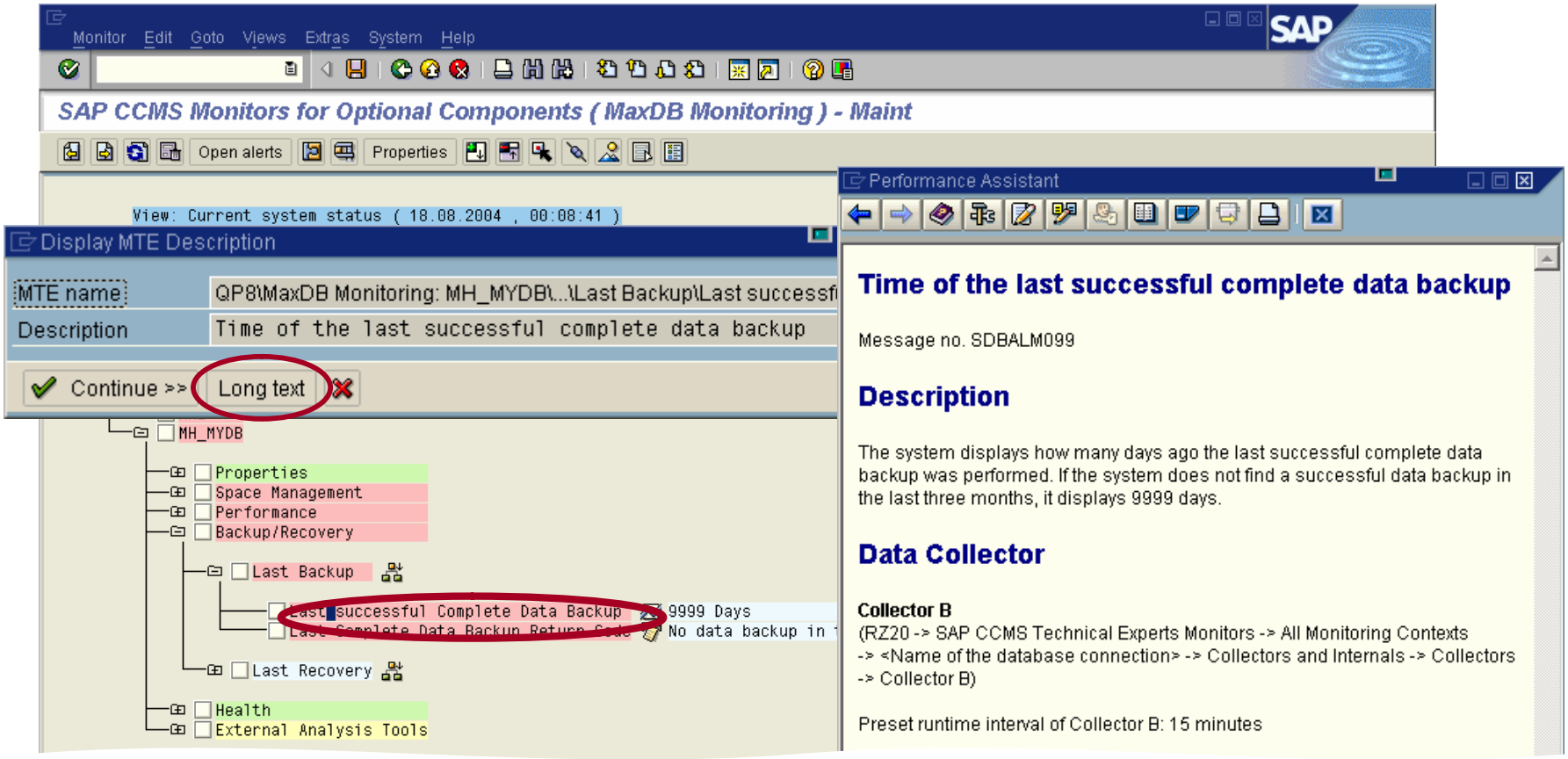
To create a data backup use the Database Manager GUI and choose *Backup → Database*. The Backup Wizard will guide you through the backup process – including the creation of a backup medium, if there is none.

Creating a Complete Backup II



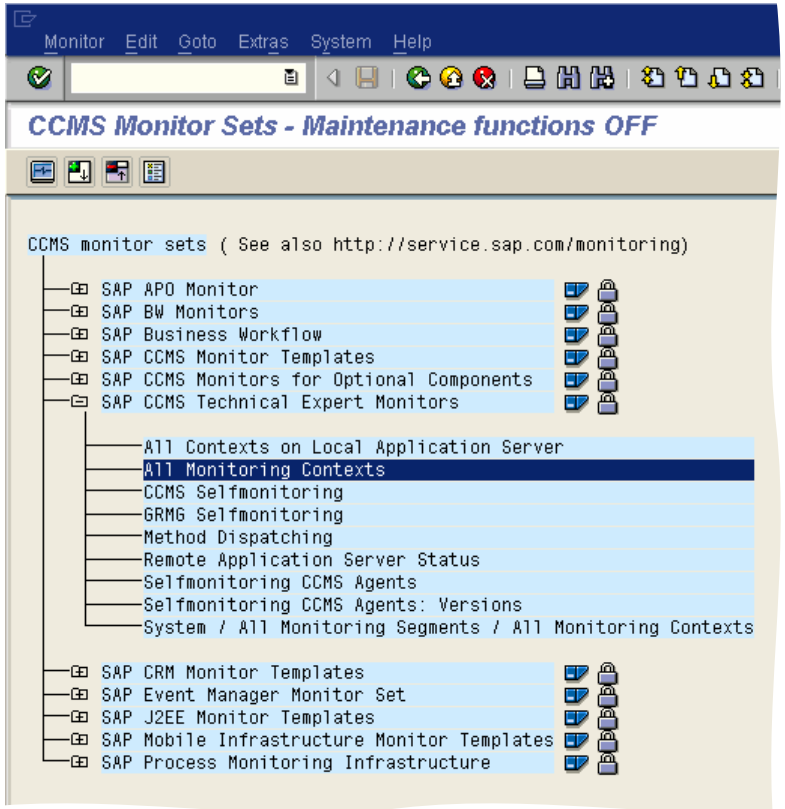
It is also possible to use the DBA Planning Calendar to create the backup. With this transaction you can schedule different actions or execute them immediately.

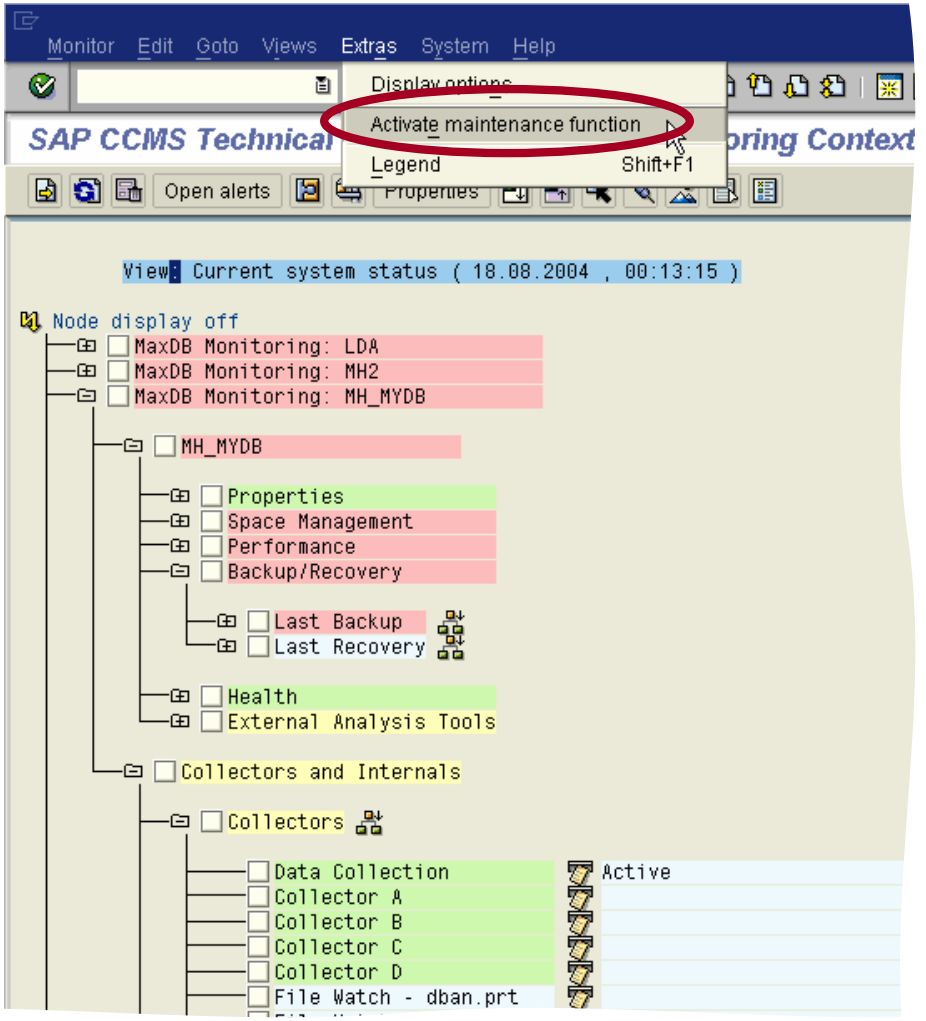
Data Collector



After you have solved the problem, you can either wait until the data collector runs the next time or start the data collector manually. To figure out which data collector is responsible for this node, place the cursor on this node and press *F1* then choose *Long Text*.

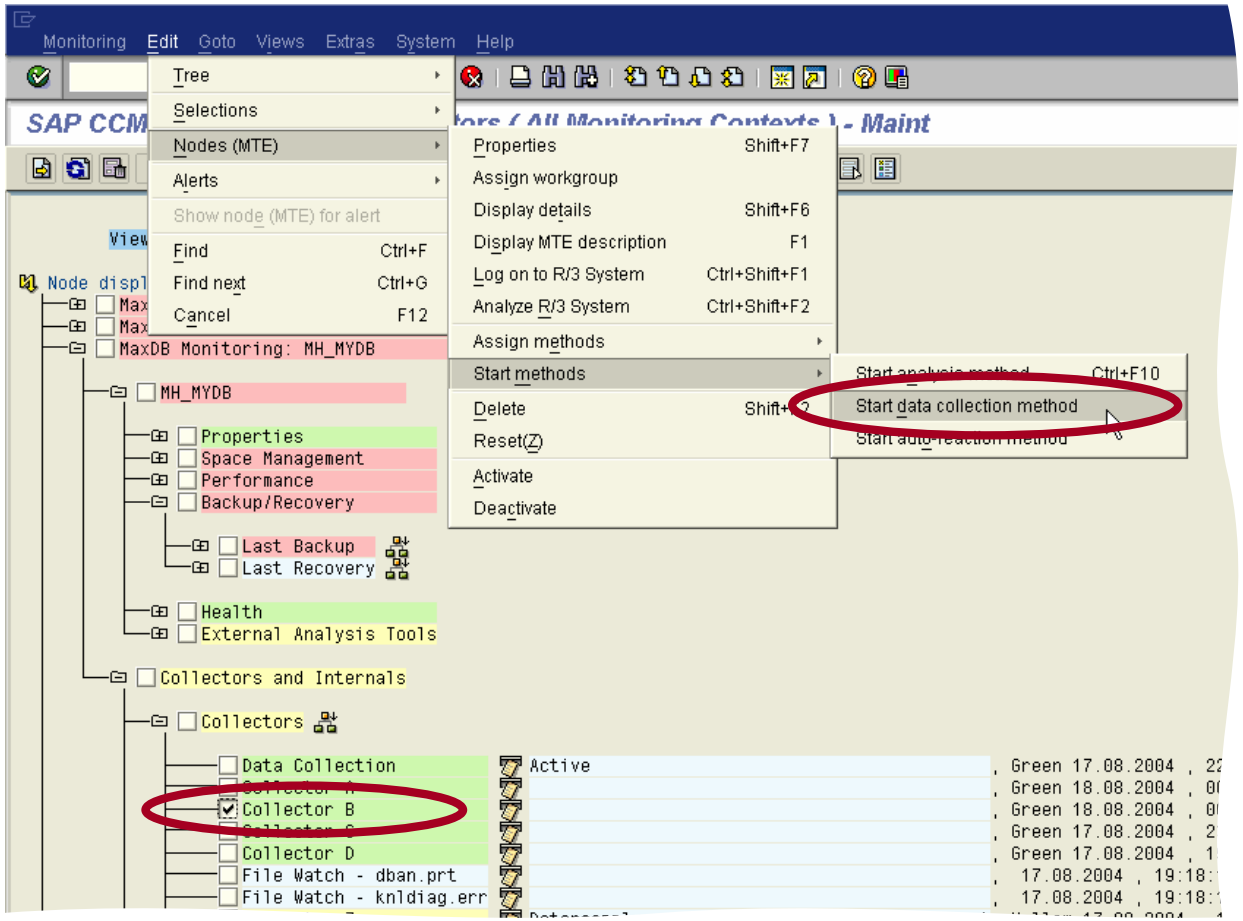
Manual Start of Data Collection for Alert Monitor





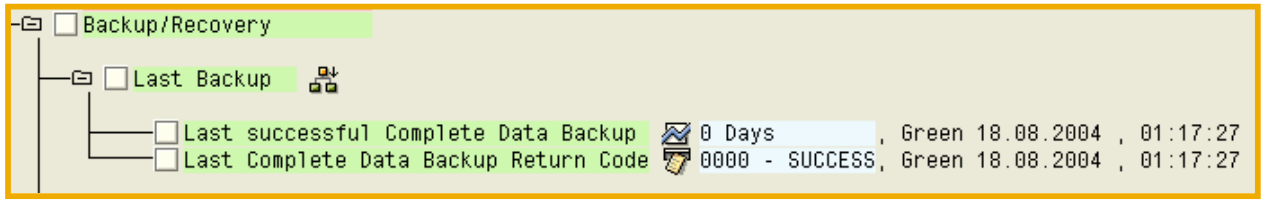
To be able to start the data collector, you have to activate the maintenance function in transaction RZ20. Choose *CCMS monitor sets → SAP CCMS Technical Expert Monitors → All Monitoring Contexts*. Then choose *Extras → Activate maintenance function*.

Start Data Collection Method

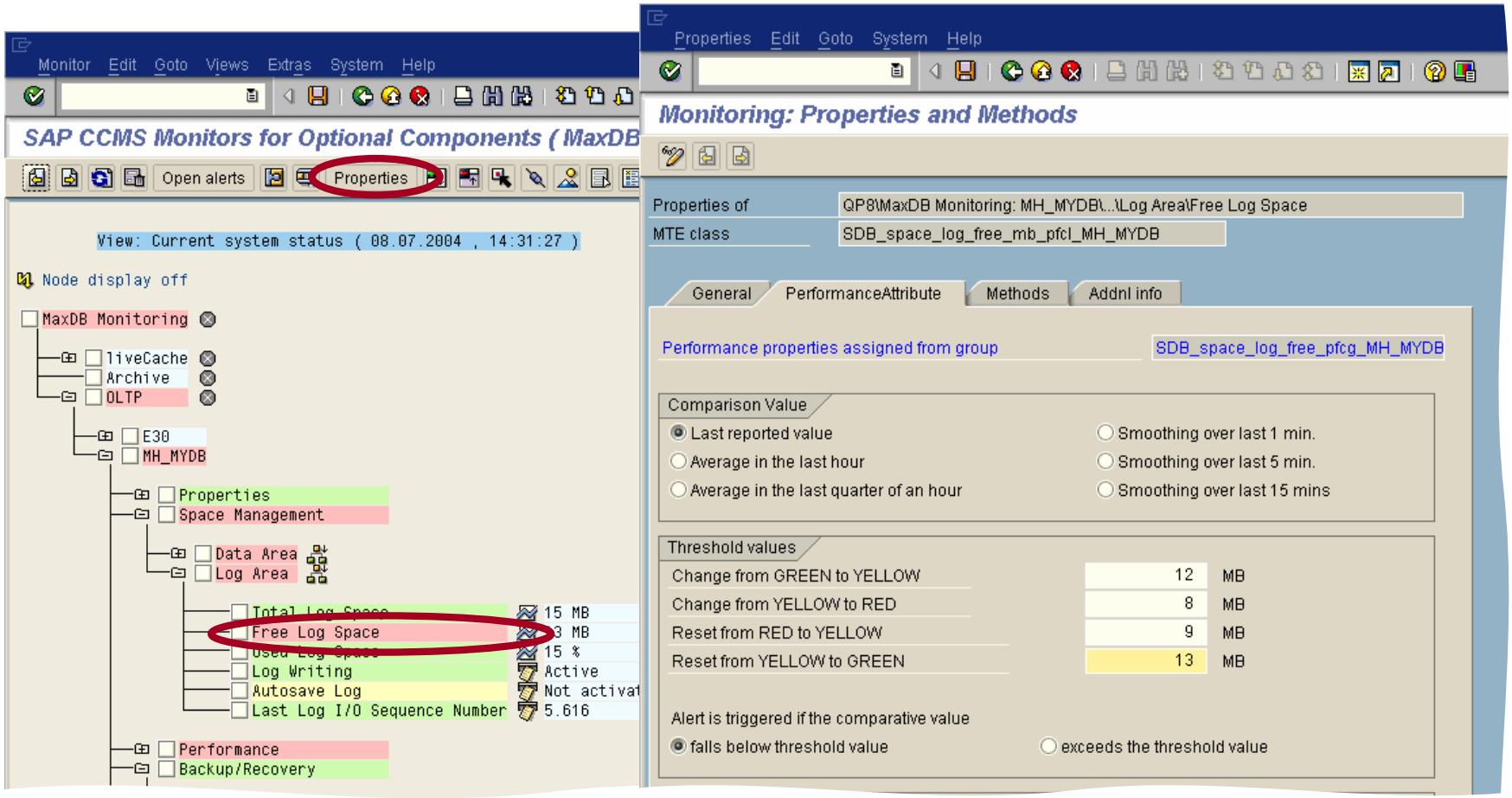


Then you can select the data collector (make sure the checkbox is marked) and choose *Edit → Nodes (MTE) → Start methods → Start data collection method*.

If you refresh the display in RZ20 then, the backup node will be green.

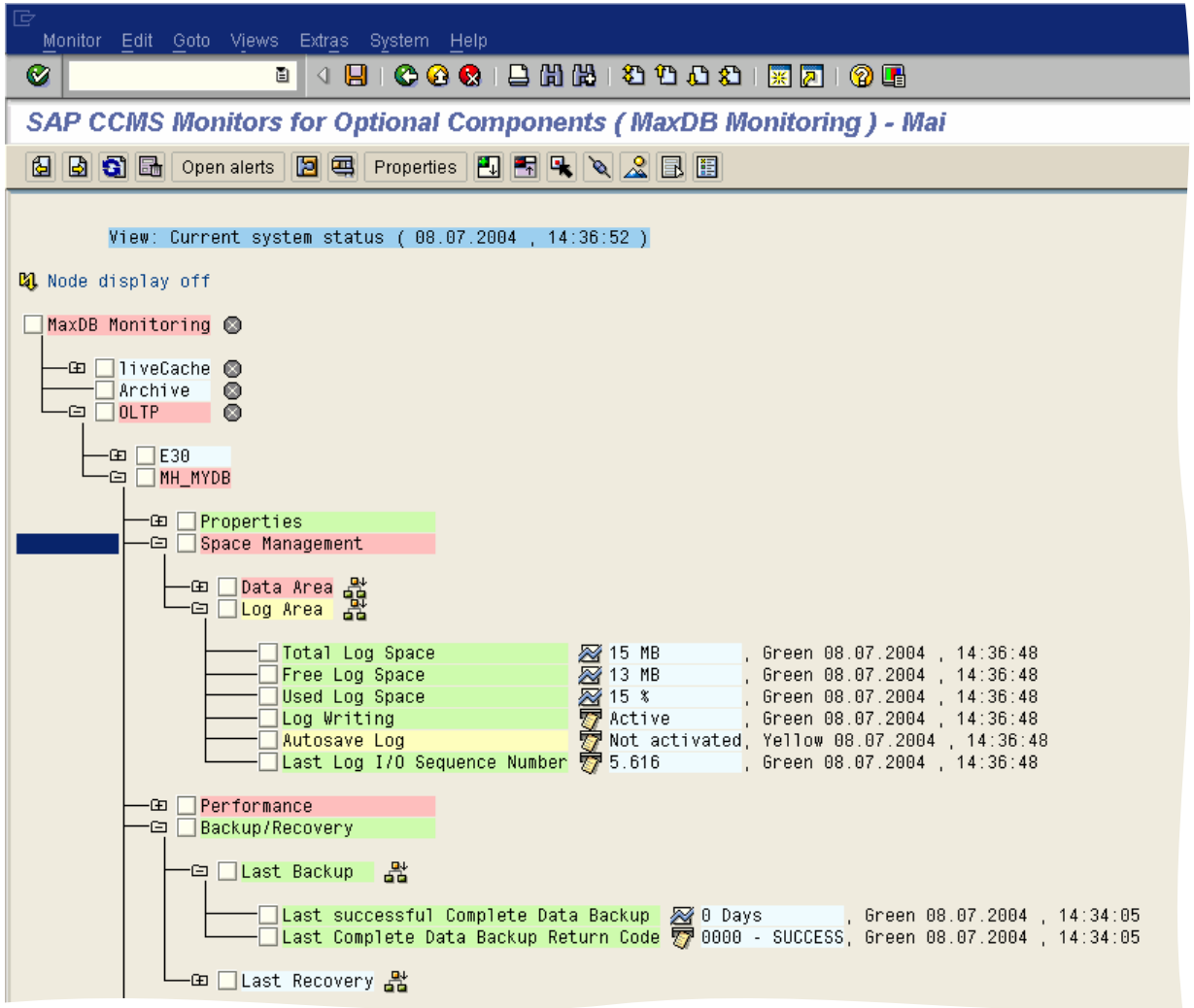


Changing Threshold Values



Sometimes the default threshold values for the nodes do not fit your requirements.
In this example a very small test database is monitored.
The log volume is nearly empty but the alert is red.
Mark the concerning node and choose *Properties* to adapt the threshold values.

Green Alerts



After changing the threshold values the node concerning the Free Log Space is green.

Chapter

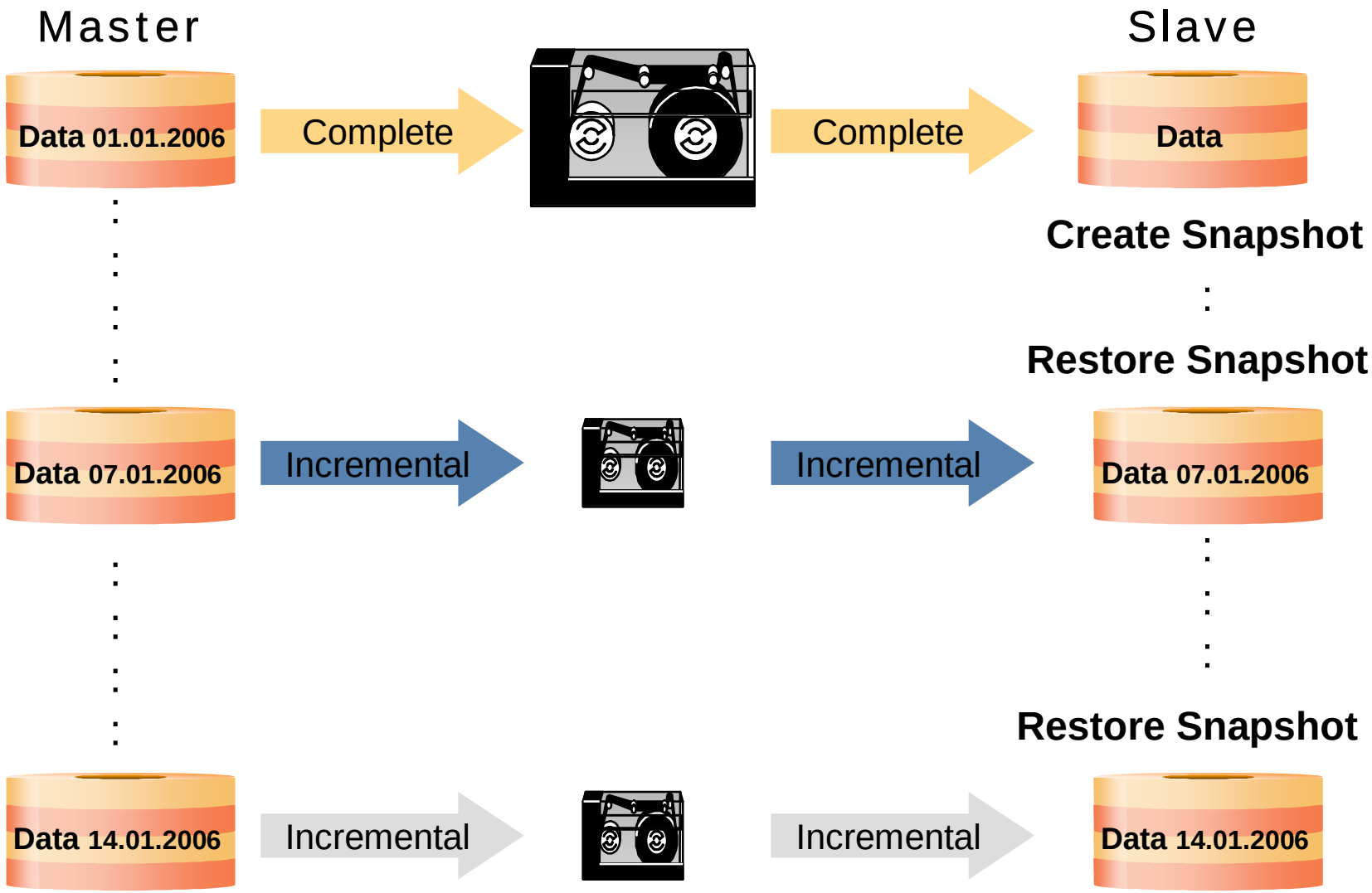
Snapshot

Freezing a database image

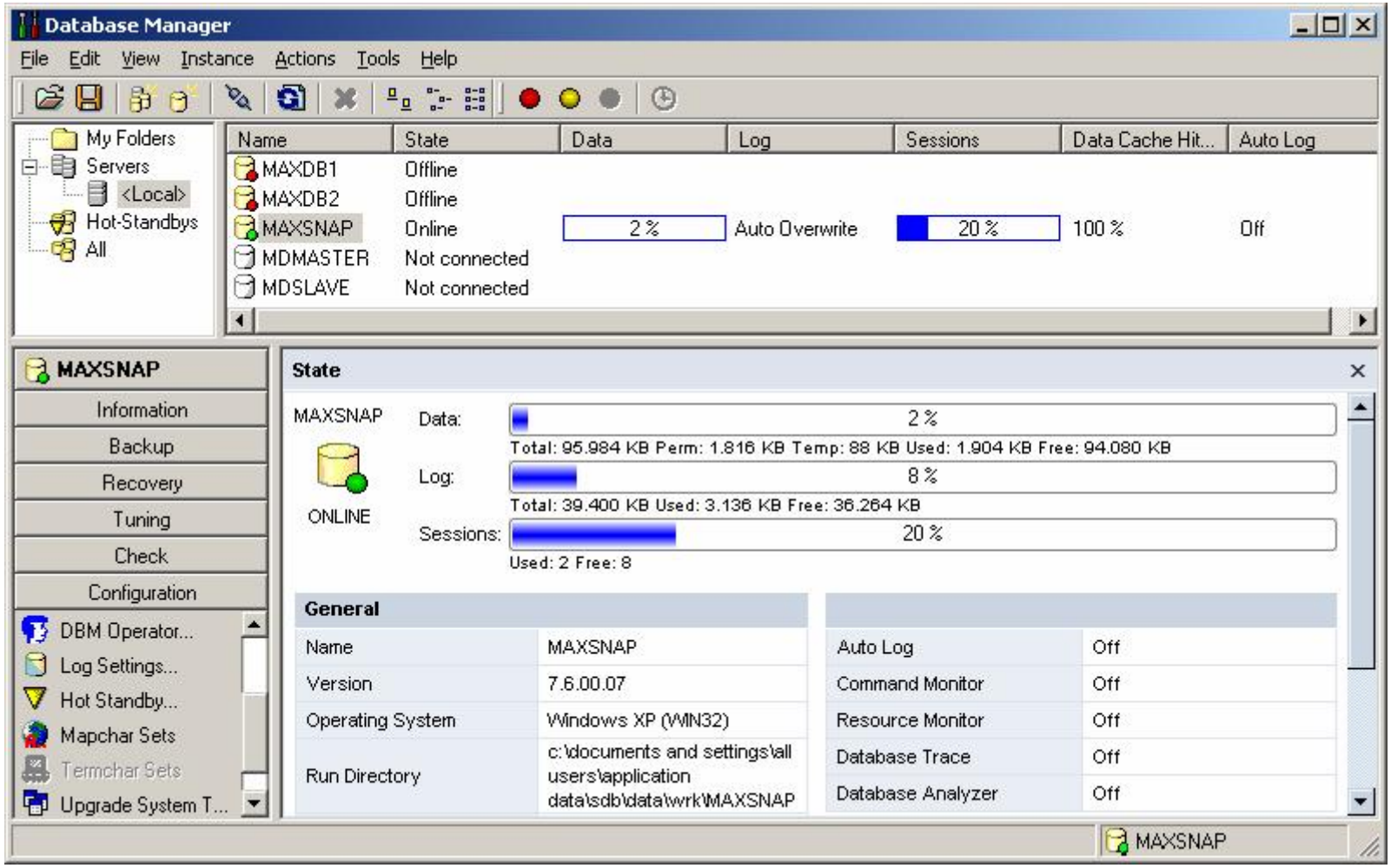
- Create Snapshot (ADMIN)
- Revert to Snapshot (ADMIN)
- Drop Snapshot (ADMIN)

- Ideas of use:
  - Very fast point in time resetting (e.g. during upgrades)
  - Restoring training-systems to a defined status

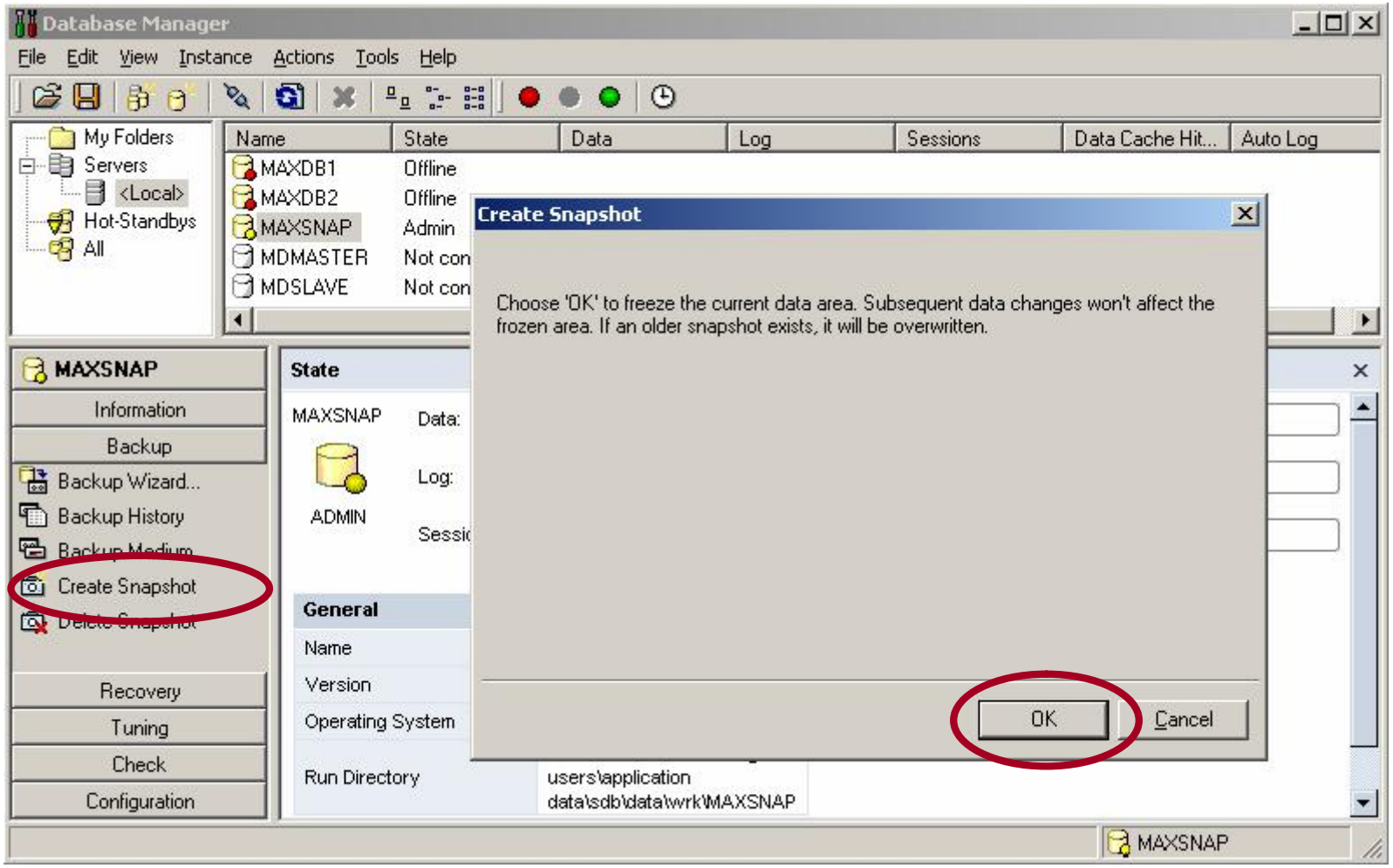
Master – Slave Support w/ Snapshots



Snapshots: Database After Setup

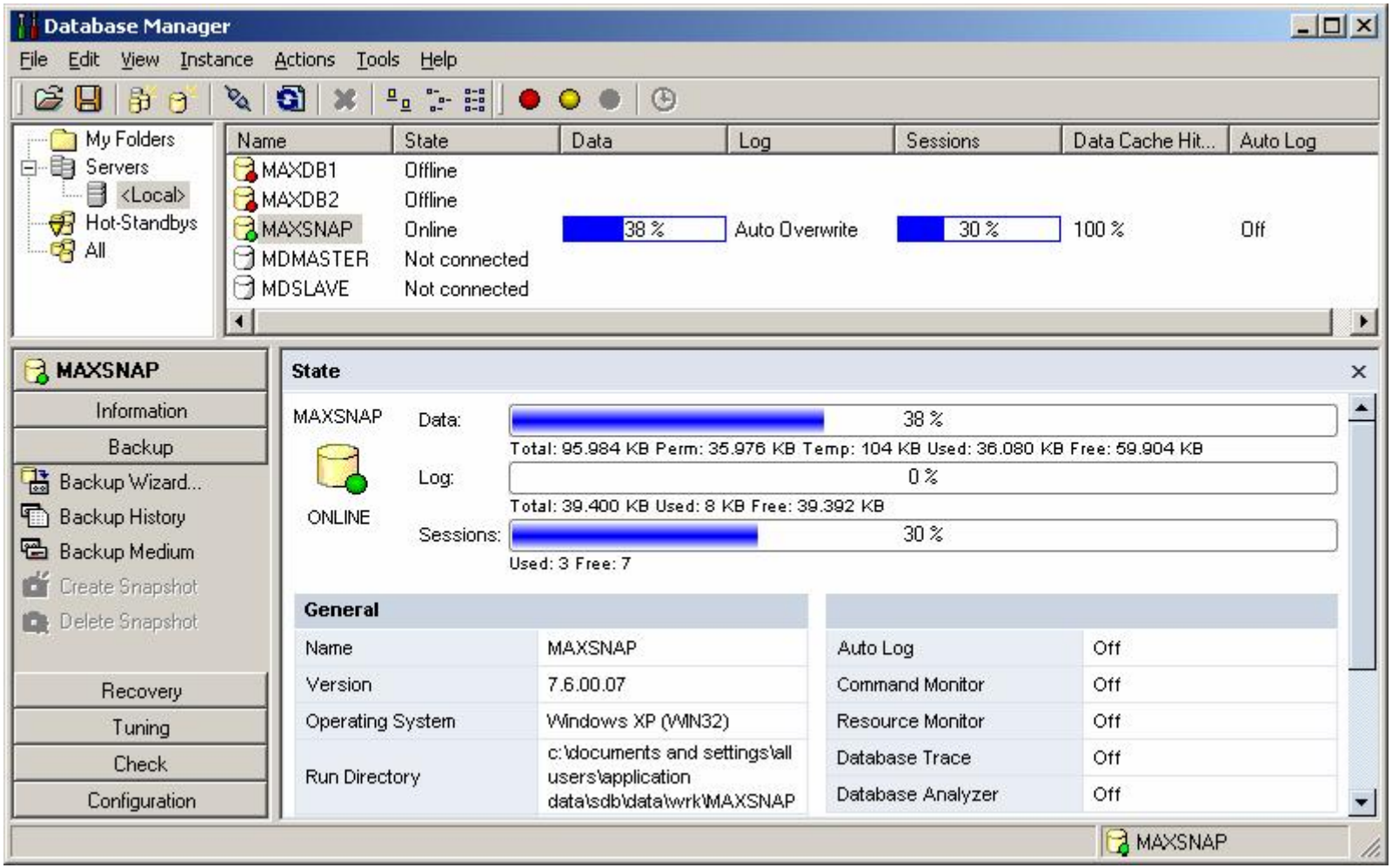


Taking a Snapshot



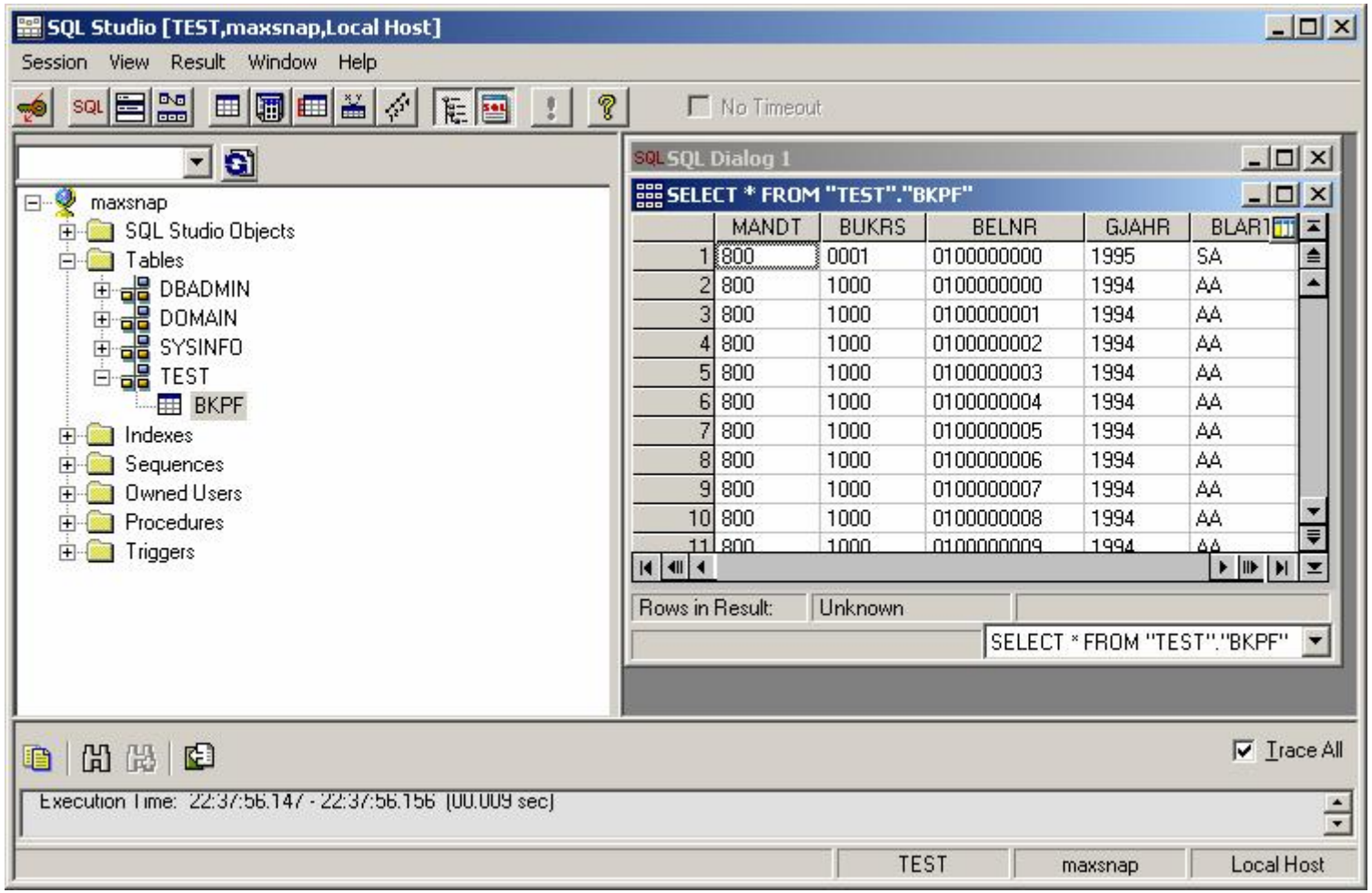
If you would like to create a snapshot, the database has to be in ADMIN mode. *DBMGUI* allows to create the snapshot with menu *Backup g Create Snapshot*.

Some Data Has Been Loaded

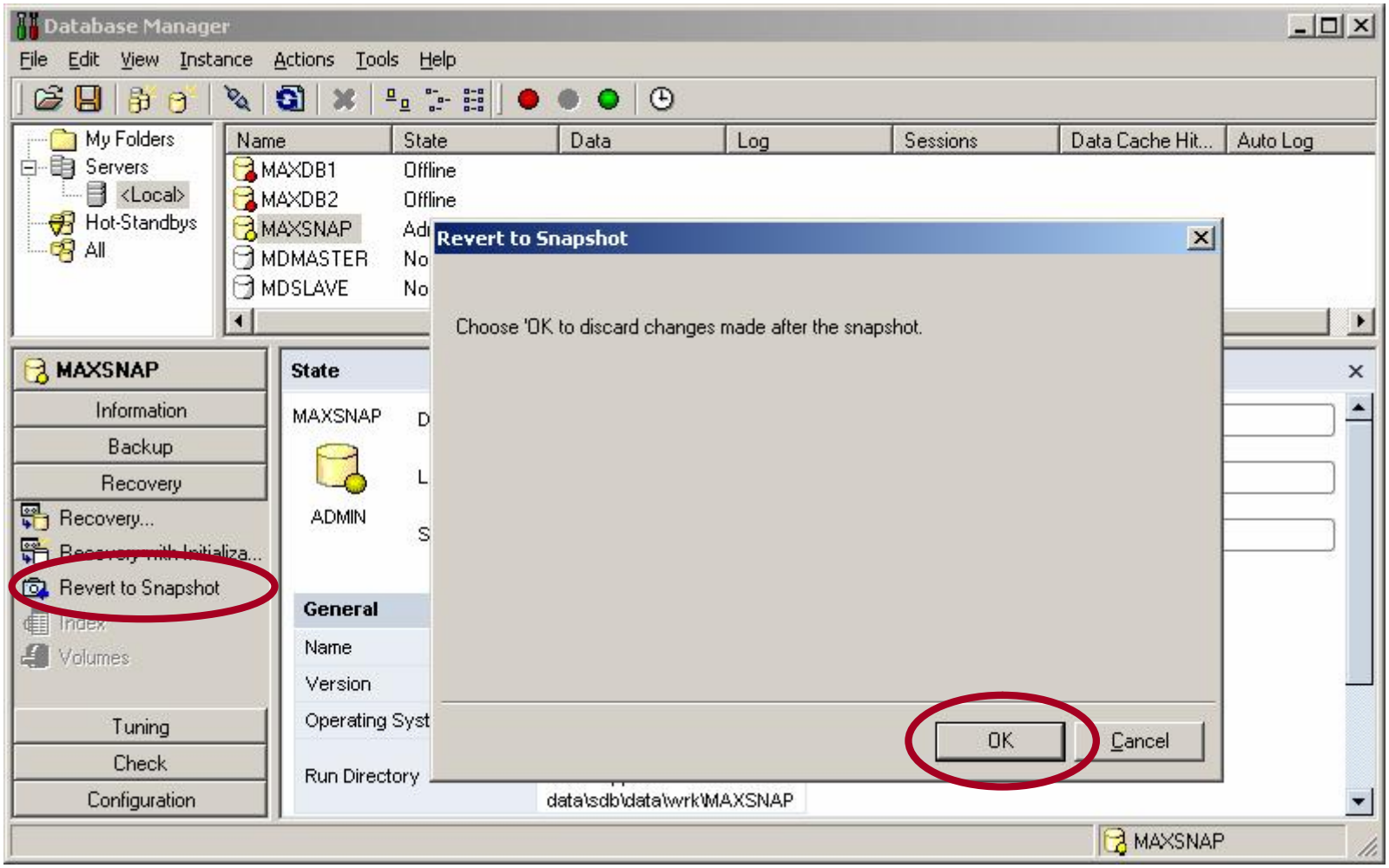


Then you can use your database instance normally – e.g. you can load some data.

Checking Data in SQL Studio

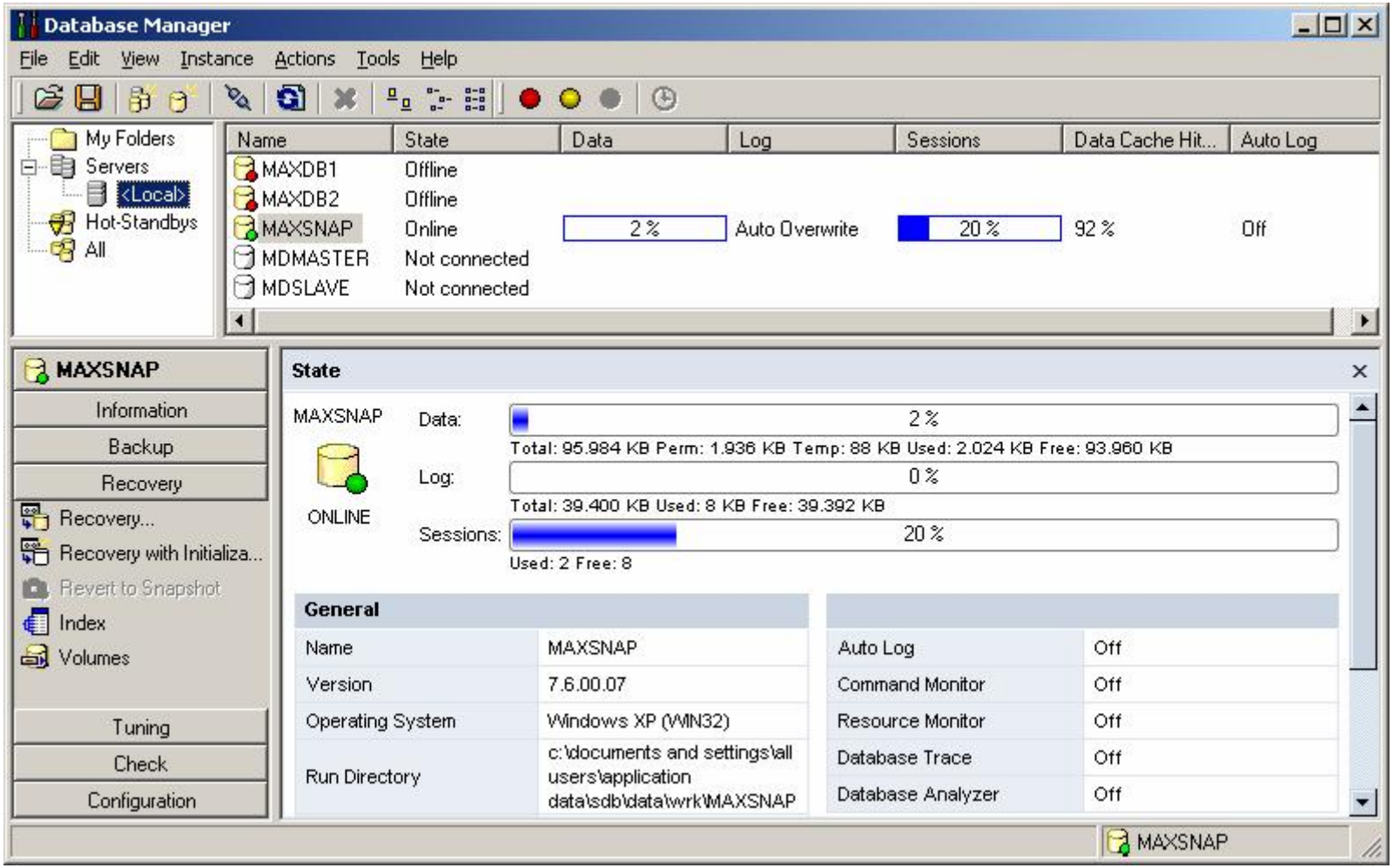


Admin Mode: Revert to Snapshot



If you would like to revert to the snapshot, choose menu Recovery → Revert to Snapshot. This can be done in ADMIN mode, only.

After Reverting to Initial State



After you restored the snapshot, all data loaded after the creation of the snapshot is gone, data which was deleted, is available again.

Summary

Transactions DB59, DB50 and RZ20 help you to monitor and administer all MaxDB database instances in your system landscape.

- Transaction DB59 is the central entry point where the instances can be integrated into the monitoring system. Using this transaction you can easily switch to the Database Assistant for each of your database instances.
- Transaction DB50 is the Database Assistant which allows to monitor your MaxDB instance. Performance problems can be analyzed and solved using this transaction.
- The DBA Planning Calendar enables you to schedule important database tasks like backups or consistency checks.
- In transaction RZ20 a special branch for the MaxDB instances can be created, so that critical situations are reported using the SAP Alert concept.

Database Manager (GUI) provides all necessary functions to administer existing database instances and to create new instances – e.g. standby databases.

Thank you!

Copyright 2007 SAP AG
All rights reserved