

SAP® MaxDB™
Error Diagnosis
Version 7.7

Heike Gursch

THE BEST-RUN BUSINESSES RUN SAP™ 



Types of errors / classification of errors

Errors during installation

Log files / traces / dumps

Check table / check backup

Analysis examples

- Connection problems
- DB full / log full situations
- Hanger situations
- Backup/restore problems
- Analyzing system errors -9028 / -9026

Extracting pages (x_diagnose)



Installation problems

Connect problems

Crash situations

Hanger situations

System errors (-10000 < error number <= -9000)

- with or without crash; reproducible ?

SQL errors (e.g. wrong result sets)

- reproducible effect ?

Save / restore errors

- valid backup ?

When an error occurs, the first step is to categorize it, as different types of errors call for different corrective measures. In some of the cases described here, categorization is simple. In some cases, however, it may not be immediately clear which of the active components is responsible for the undesirable system behavior. It may not even be clear whether the problem has to do with the database, and thus whether the diagnosis options described in this chapter will be helpful.

Problems arising from the installation of the software are easy to identify as such.

In an SAP environment, connection problems generally manifest themselves in that the database is, in principle, ready for operation, but the client processes cannot get a connection.

If the database, without any conscious action having been taken, is no longer ready for operation, it may have crashed. However, it is also possible that the database has consciously performed an emergency shutdown due to an existing error situation.

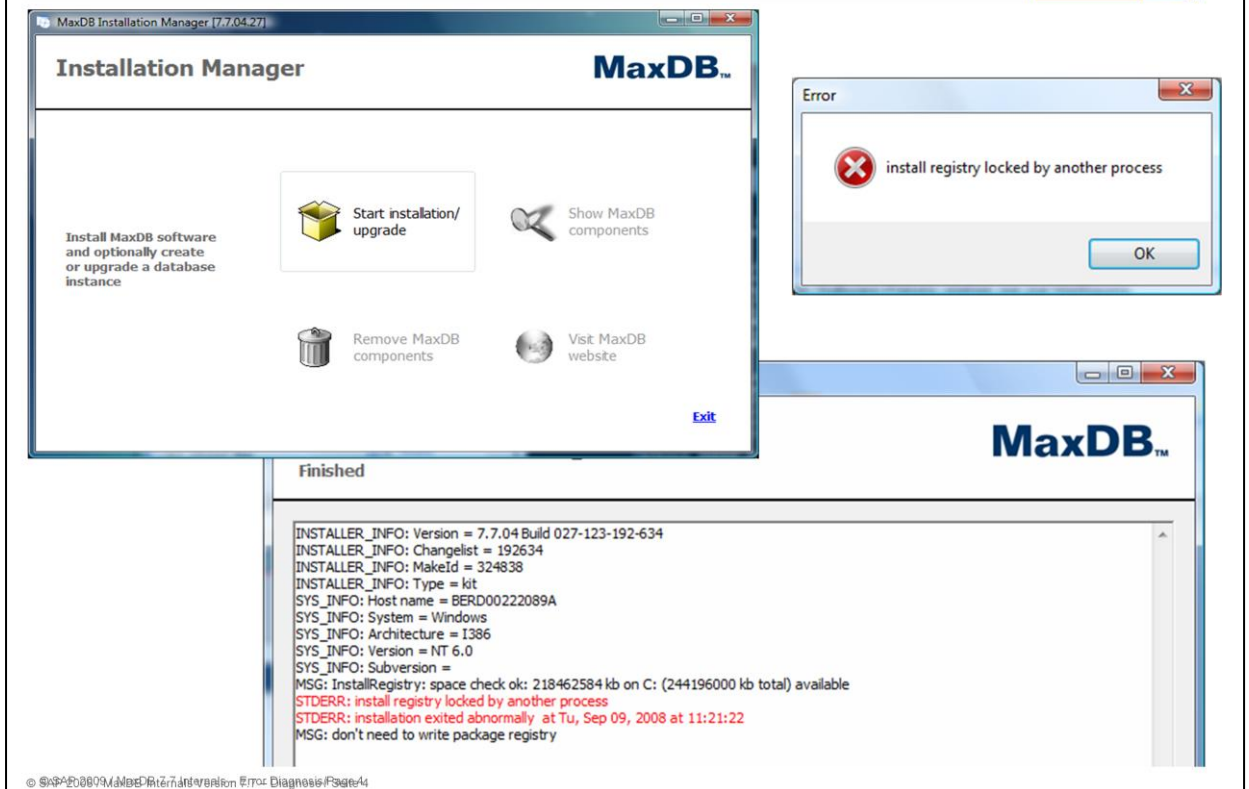
In the SAP environment, a hang situation is indicated by the presence of an hourglass. Determining the area in which a problem exists (lock collision, "blockage" of the system resources, etc.) is not necessarily insignificant.

System errors are serious errors and in the SAP system are often logged as error -602. The unique error number can be found in the **KnIMsg (knldiag)** file.

As a rule, transactions terminate with a short dump if an SQL error occurs.

Backup or restore problems are recognizable as such; however, problems in this area frequently are due not to the database itself, but rather have an external cause (operating system, external backup tool).

Installation - sdbsetup



As of version 7.6 the software installation for MaxDB is done with the installation manager (**sdbsetup**). It is available for use after unpacking the software package provided on the Service Marketplace.

Both initial installations and software updates can be done by choosing “Start Installation/Upgrade”.

Deinstallations of MaxDB software can be done via “Remove MaxDB components”. Do not delete the MaxDB software with operating system resources under any circumstances!

“Show MaxDB components” provides information about installation directories, versions of the installed software elements, databases and additionally guides you to the installation log files.

Installation logs are stored in the directory <indepdatapath>/wrk ; these can be useful, even at a later point in time. The name contains the type of installation as well as a time stamp, which makes it possible to determine the corresponding package.

The <indepdatapath> can be determined by using the following dbmcli command:

```
■ dbmcli -d <SID> -u <dbm-user>,<password> dbm_getpath indepdatapath
```

More extensive migration work may be necessary if you are making a big release-jump; the procedure is described in the corresponding notes.

In the example shown above at the beginning of the installation an error occurred. The installation manager guides the user to the log file where he finds a more exact and often self-explaining error message. The installation failed as another **sdbupd** was started at the same time.



Call: `sdbinst/sdbupd`

Option `-help` lists valuable options.

Installation logs are written in `<indepdatapath>/wrk`.

```
MaxDBPrecompiler_install-13.09.2004-11.34.log
MaxDBRuntimeForSAPAS_install-26.07.2004-14.47.log
MaxDBServer_install-02.03.2004-18.42.log
MaxDBServer_install-29.07.2004-14.32.log
MaxDBUninstall_install-02.12.2004-15.23.log
MaxDBUpdate_install-01.09.2004-09.13.log
```

Deinstallation of MaxDB software: `sdbuninst` or `SDBSETUP`

No deletion with operating system!

As of version 7.6, the tool `SDBSETUP` with GUI interface is available for all supported platforms

Installation of the MaxDB software in versions 7.5 and smaller is done with the tools ***sdbinst*** and ***sdbupd***. In 7.6 and 7.7 they are also part of the delivered software and can be used if the installation is not done interactively. For an initial installation, ***sdbinst*** is used; overwriting an older version with a current one is done with ***sdbupd***.

You get the call options by entering the option `-help`.

If you encounter problems during installation, a corresponding message is issued. The message should inform you as to what error has occurred and how to fix it.

Installation logs are stored in the directory `<indepdatapath>/wrk`.

`sdbinst/sdbupd` enable you to install individual components from the overall package. You use the `-package` option to specify the component(s).

You use ***sdbuninst*** or ***SDBSETUP*** to deinstall the software.

Examples (look for STDERR):

```
STDOUT: start extraction test run of "SAPDBBAS.TGZ"  
STDERR: test run failed: cannot extract bin/dbmrfc: cannot write to /sapdb/pro  
grams/bin/dbmrfc: Text file busy - no file(s) of "SAPDBBAS.TGZ" extracted!  
STDERR: maybe any sap db software is running... please stop all  
STDERR: installation exited abnormally at Th, Aug 14, 2003 at 14:46:41
```

```
...  
STDOUT: start extraction test run of "SAPDBUTL.TGZ"  
STDERR: maybe any sap db software is running... please stop all  
STDERR: installation exited abnormally at Th, Aug 14, 2003 at 14:48:30  
...
```

```
WRN: try to install release "7.3.00.32" over existing "7.3.0.36"  
MSG: update test: installed release newer  
MSG: update from "7.3.0.36" to "7.3.00.32" not allowed  
STDERR: cannot downgrade package  
STDOUT: skipping package
```

The option `-force_extract` also substitutes files being in use.

To quickly find errors in the installation logs, look for the key word STDERR.

A common cause of errors in Windows systems is that DLLs cannot be overwritten or - more generally - that a component to be overwritten is still running (for example the `x_server`).

In the first two cases you get the message that the database software has not been completely stopped, so a re-installation would fail. To identify such problems ahead of time and avoid having the installation fail half-way through, the actual installation is preceded by a test run.

As of version 7.6.01 the option `-force_extract` cares for the substitution of programs and libraries being in use by copying the existing files.

In the third example, the attempt was made to install an older version over a newer one. This is not permitted since downward compatibility cannot be ensured, for instance when loading the system tables.

```

Select Command Prompt - telnet p34777
PCR 7250 7.2.05.24 in /sapdb/programs
-----
check files... ok
check dependencies... ok
package data is consistent

Database Kernel 7.4.03.28 32 bit in /sapdb/TEST/db
-----
check files... failed
mode of /sapdb/TEST/db/pgm/lserver was modified
mode of /sapdb/TEST/db/pgm/dbmsrv was m

check dependencies... ok

check rte registration of package... ok
package data is inconsistent
...

```

```

Select Command Prompt - telnet p34777
VERIFICATION SUMMARY:
*****
INVALID PACKAGES:           0
VALID PACKAGES:            26
INCONSISTENT PACKAGES:     10

TOTAL FILES:                1286
MISSED FILES:               5
MODIFIED FILES:             1
FILES WITH MODIFIED PERMISSIONS: 5

p34777:e30adm 175>

```

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 7

sdbverify is a tool that checks all installations on a computer for completeness. During the check, any inconsistencies due to impermissible software combinations are detected.

Using the registry entries, **sdbverify** checks whether the status in the file system still corresponds to the contents of the installation package.

In the example above, it is noted that the access rights (under Unix) were subsequently changed.

The result is a summary of the installations that have been checked.

Start **sdbverify** with user root under Linux/UNIX.

To get an overview of the registered installations, you can also use the tool **sdbregview**. If you call it with the option **-l**, it outputs a short list; without the option it outputs comprehensive information about every installation.



```

e70adm on ld1032: /home/e70adm - Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help
ld1032:e70adm 73> xinstinfo E70
IndepData      : /sapdb/data
IndepPrograms  : /sapdb/programs
InstallationPath : /sapdb/E70/db
Kernelversion   : KERNEL 7.6.00 BUILD 026-123-123-737
Rundirectory   : /sapdb/data/wrk/E70
ld1032:e70adm 74> dbmcli inst_enum
OK
7.6.00.26      /sapdb/MYWEBDAU/db
7.6.00.26      /sapdb/E70/db
7.6.00.27      /sapdb/SHADOW/db
7.5.00.35      /sapdb/SHADOW75/db
7.6.00.26      /sapdb/XYZ/db
ld1032:e70adm 75> dbmcli db_enum
OK
MYWEBDAU      /sapdb/MYWEBDAU/db      7.6.00.26      fast      running
MYWEBDAU      /sapdb/MYWEBDAU/db      7.6.00.26      quick     offline
MYWEBDAU      /sapdb/MYWEBDAU/db      7.6.00.26      slow      offline
MYWEBDAU      /sapdb/MYWEBDAU/db      7.6.00.26      test      offline
E70           /sapdb/E70/db           7.6.00.26      fast      running
E70           /sapdb/E70/db           7.6.00.26      quick     offline
E70           /sapdb/E70/db           7.6.00.26      slow      offline
E70           /sapdb/E70/db           7.6.00.26      test      offline
SHADOW2       /sapdb/SHADOW/db       7.6.00.27      fast      offline
SHADOW2       /sapdb/SHADOW/db       7.6.00.27      quick     offline
SHADOW2       /sapdb/SHADOW/db       7.6.00.27      slow      offline
SHADOW2       /sapdb/SHADOW/db       7.6.00.27      test      offline
SHADOW1       /sapdb/SHADOW75/db     7.5.00.35      fast      running
SHADOW1       /sapdb/SHADOW75/db     7.5.00.35      quick     offline
SHADOW1       /sapdb/SHADOW75/db     7.5.00.35      slow      offline
SHADOW1       /sapdb/SHADOW75/db     7.5.00.35      test      offline
XYZ           /sapdb/XYZ/db           7.6.00.26      fast      running
XYZ           /sapdb/XYZ/db           7.6.00.26      quick     offline
XYZ           /sapdb/XYZ/db           7.6.00.26      slow      offline
XYZ           /sapdb/XYZ/db           7.6.00.26      test      offline
ld1032:e70adm 76>

```

The tool **xinstinfo** provides a quick overview of the installation paths used on a system. It displays the "Independent Data Path" and the "Independent Program Path", or in other words, the paths that are valid for all the databases installed on the computer. The programs found there are always operated in the highest installed version (for example the x_server).

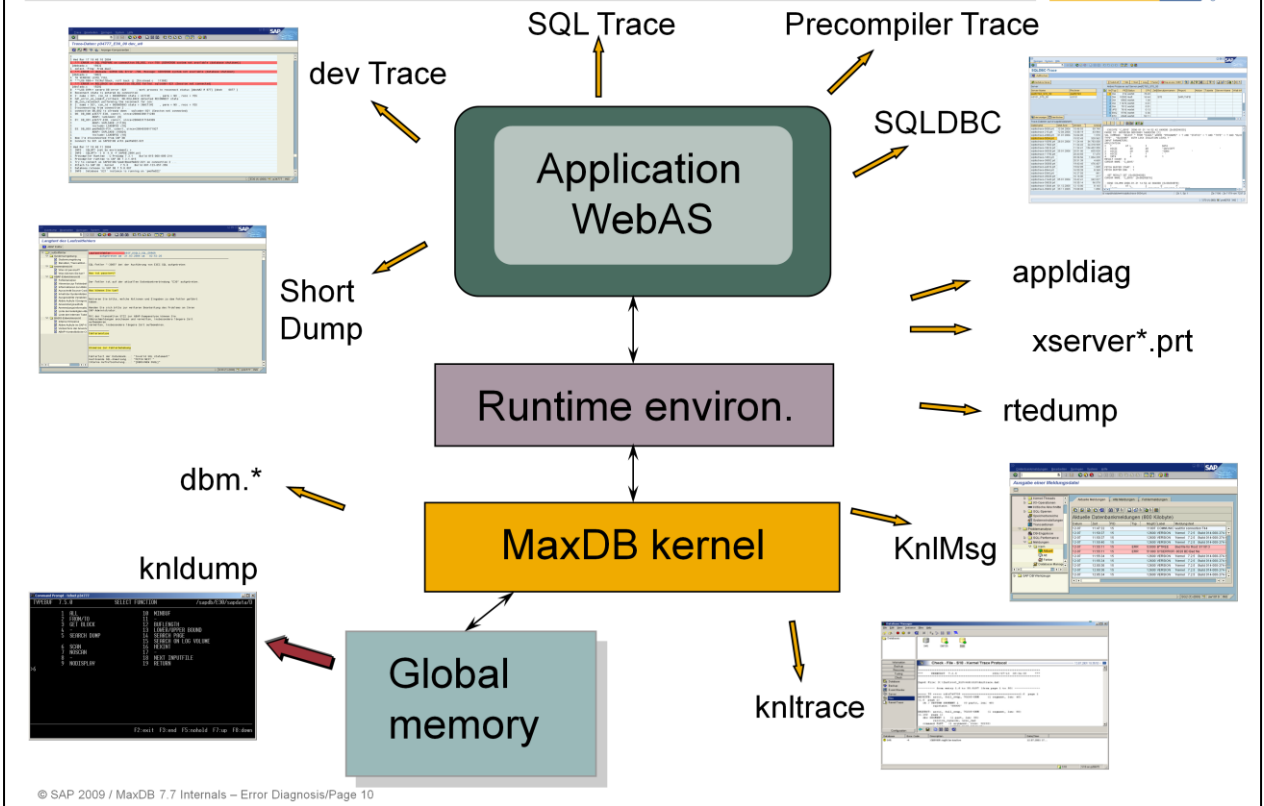
If **xinstinfo** is called with a database name, you also get database-specific information.

Using **dbmcli** with the entry **inst_enum**, you get a list of the versions installed on the computer (dependent paths). The command **db_enum** lists the databases in their different variants (fast, quick, slow, test) as well as their current operational states.



- System log (SM21)
- ABAP Short Dump (ST22)
- dev logs
- SQL Trace (ST05)
- Precompiler Trace
- SQLDBC
- appldiag
- xserver_<hostname>.prt
- dbm.prt, dbm.utl, dbm.knl, dbm.ebp, dbm.ebl
- KnIMsg (knldiag)
- Event Viewer
- knltrace
- knldump
- rtedump

Diagnostic Files



© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 10

Short Dump

ABAP short dumps are generated by the WebAS or R/3 system when unexpected return codes occur in the SQL environment.

dev-Trace

The Developer Traces are logs of the disp+work processes of the SAP system.

SQL Trace

SQL commands and their runtimes are logged.

Precompiler Trace

SQL trace of the order interface.

SQLDBC

Trace for the SQLDBC (SQL Database Connectivity) interface.

appldiag

If errors between the runtime environment and the kernel occur, they are entered in the *appldiag* file. This file is created for each operating system user.

xserver_<hostname>.prt

If errors occur during communications via the x_server, they are entered in the *xserver_<hostname>.prt* file.

rtdump

If a crash occurs, the runtime environment writes its status in the *rtdump* file. It is an ASCII output of the command `x_cons <SERVERDB> show all`

dbm.*

Various log files for the backup environment or for logging DBM server commands.

KnIMsg (knldiag)

The kernel writes information and messages to the *KnIMsg* file. It has a fixed size and is overwritten cyclically. After a crash, it also contains the backtrace.

knitrace

This file is written by the kernel when the Vtrace is activated and following a crash. It has a predefined, fixed length.

knldump

During an emergency shutdown, the global memory is written to the *knldump* file. The corresponding file system should be sufficiently large.



System Log: Local Analysis of pwwf2763

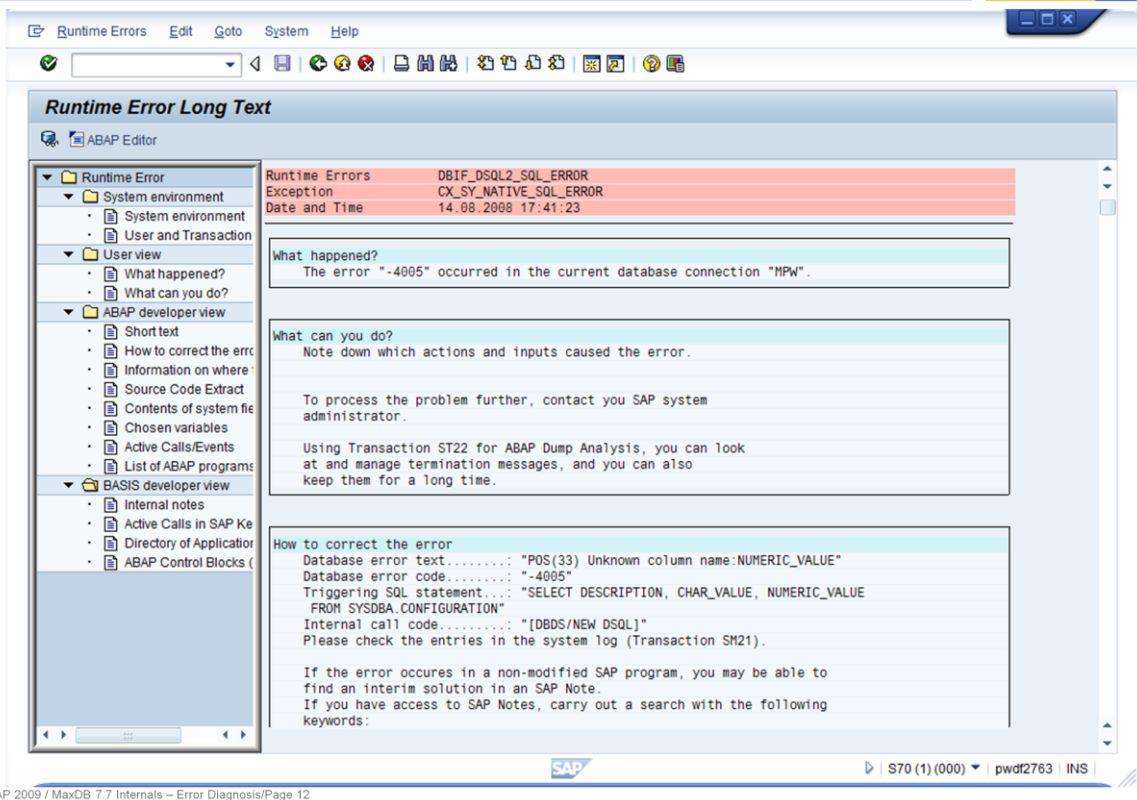
System Log: Local Analysis of pwwf2763 16

Date : 15.08.2008

Time	Type	Nr	Clt	User	TCode	Priority	Grp	N	Text
17:41:23	DIA	000	000	S70		AB	0	0	Run-time error "DBIF_DSQ2_SQL_ERROR" occurred
17:41:23	DIA	000	000	S70		AB	1	1	> Short dump "080814 174123 pwwf2763_S70_50 " generated
17:46:01	DIA	000	000	S70	DB50	AB	0	0	Run-time error "DBIF_DSQ2_OBJ_UNKNOW" occurred
17:46:01	DIA	000	000	S70	DB50	AB	1	1	> Short dump "080814 174601 pwwf2763_S70_50 " generated
01:01:23	BTC	000	000	DDIC		BY	2	2	Database error -10799 at CON
01:01:23	BTC	000	000	DDIC		BY	0	0	> Connection failed (RTE:database not running)
03:03:41	DIA	000	000	SAPSYS		PM	2	2	Certificate with PSE type >System-PSE< has been invalid for 733270 days
08:01:11	BTC	000	000	DORN		BY	2	2	Database error -4005 at OPC
08:01:11	BTC	000	000	DORN		BY	0	0	> POS(33) Unknown column name:NUMERIC_VALUE
08:01:12	BTC	000	000	DORN		AB	0	0	Run-time error "DBIF_DSQ2_SQL_ERROR" occurred
08:01:12	BTC	000	000	DORN		AB	1	1	> Short dump "080815 080112 pwwf2763_S70_50 " generated
08:01:12	BTC	000	000	DORN		D0	1	1	Transaction Canceled 00 671 (DBIF_DSQ2_SQL_ERROR 20080815080112pwwf2763_S70_50 DORN 00
08:03:31	DIA	000	000	SAPSYS		BY	2	2	Database error -4005 at OPC
08:03:31	DIA	000	000	SAPSYS		BY	0	0	> POS(33) Unknown column name:NUMERIC_VALUE
08:03:31	DIA	000	000	SAPSYS		AB	0	0	Run-time error "DBIF_DSQ2_SQL_ERROR" occurred
08:03:31	DIA	000	000	SAPSYS		AB	1	1	> Short dump "080815 080331 pwwf2763_S70_50 " generated
08:03:31	DIA	000	000	SAPSYS		D0	1	1	Transaction Canceled 00 671 (DBIF_DSQ2_SQL_ERROR 20080815080331pwwf2763_S70_50 SAPSYS
08:09:52	DP					Q0	1	1	Operating system call rcv failed (error no. 10054)
08:09:52	DP					Q0	4	4	Connection to user 2803 (S70), terminal 16 (BERD00145763) lost
16:21:02	DIA	001	000	FUCHSU	DB50	AB	0	0	Run-time error "CONVT_NO_NUMBER" occurred
16:21:02	DIA	001	000	FUCHSU	DB50	AB	1	1	> Short dump "080815 162102 pwwf2763_S70_50 " generated
16:25:29	DIA	001	000	FUCHSU	DB50	AB	0	0	Run-time error "DBIF_DSQ2_OBJ_UNKNOW" occurred
16:25:29	DIA	001	000	FUCHSU	DB50	AB	1	1	> Short dump "080815 162529 pwwf2763_S70_50 " generated
16:43:02	DIA	001	000	FUCHSU	DB50	AB	0	0	Run-time error "CALL_FUNCTION_CONFLICT LENG" occurred
16:43:02	DIA	001	000	FUCHSU	DB50	AB	1	1	> Short dump "080815 164302 pwwf2763_S70_50 " generated

S70 (1) (000) | pwwf2763 | INS

Transaction sm21 displays the system log of the SAP system. The system log is not written under the control of the database, but it does contain information about database errors.



Runtime Error Long Text

ABAP Editor

Runtime Errors	DBIF_DSQ2_SQL_ERROR
Exception	CX_SY_NATIVE_SQL_ERROR
Date and Time	14.08.2008 17:41:23

What happened?
The error "-4005" occurred in the current database connection "MPW".

What can you do?
Note down which actions and inputs caused the error.

To process the problem further, contact you SAP system administrator.

Using Transaction ST22 for ABAP Dump Analysis, you can look at and manage termination messages, and you can also keep them for a long time.

How to correct the error
Database error text.....: "POS(33) Unknown column name: NUMERIC_VALUE"
Database error code.....: "-4005"
Triggering SQL statement...: "SELECT DESCRIPTION, CHAR_VALUE, NUMERIC_VALUE FROM SYSDBA.CONFIGURATION"
Internal call code.....: "[DBDS/NEW DSQ2]"
Please check the entries in the system log (Transaction SM21).

If the error occurs in a non-modified SAP program, you may be able to find an interim solution in an SAP Note.
If you have access to SAP Notes, carry out a search with the following keywords:

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 12

In the SAP system, SQL errors in the database result in APAB short dumps when unexpected return codes occur.

They are not written under the control of the database, but they can be useful for analyzing error situations because they present a full picture of the error. SQL errors are otherwise not logged by the database, unless the Vtrace has been explicitly activated.

You can get a list of the short dumps that have occurred with transaction st22.

The short dump itself records which program and which ABAP command within it caused the error. You can then search for the error code in the notes.

If an unknown error occurs, it is often desirable to identify the command in "native SQL." To do this, in a reproducible case the SQL trace must then be activated with st05. At the same time, an analysis with the Vtrace may also be useful.

If necessary, you can find helpful information about the versions you are using in the section "How to correct the error".

Dev Logs (1)



```
Trace Data: Id1032_A1S_02_dev_w0
Display Components
C
C DBSDBSLIB : version 710.16, patch 0.083 (Make PL 0.85)
C MAXDB shared library (dbsdbslib) patchlevels (last 10)
C (0.083) TREX 7.1: serialization a XML-Datenstroms (note 1121417)
C (0.083) Return DS_SQLERR instead of DS_CONNECTERR (note 1121076)
C (0.060) Minimal Kernel Patch Level for NW 7.10 SP3 (note 1051559)
C
C
C Loading SQLDBC client runtime ...
C SQLDBC SDK Version : SQLDBC.H 7.6.1 BUILD 010-123-142-759
C SQLDBC Library Version : 11bSQLDBC 7.6.5 BUILD 006-123-187-279
C SQLDBC client runtime is MaxDB 7.6.5.006 CL 187279
C SQLDBC supports new DECIMAL interface : 1
C SQLDBC supports VARIABLE INPUT data : 1
C SQLDBC supports keepAlive indicator : 0
C INFO : SLOOPT= -I 0 -t 0 -S SAPR3
C Try to connect (DEFAULT) on connection 0 ...
C Attach to SAP DB : Kernel 7.7.02 Build 019-123-170-429
C Database release is SAP DB 7.7.02.019
C INFO : Database 'A1S' instance is running on 'Id1032'
C INFO : SAP DB Packet_Size = 131072
C INFO : SAP DB Min_Reply_Size = 4096
C INFO : SAP DB Comm_Size = 126976
C INFO : DBSL buffer size = 126976
C INFO : SAP DB MaxLocks = 250000
C INFO : Connect to DB as 'SAPA1S'
C Command info enabled
C Now I'm connected to SAP DB
C 00: Id1032-A1S, since=20080731085702, ABAP= <unknown> (0)
B Connection 0 opened (DBSL handle 0)
B Wp Hd1 ConName ConId ConState TX BC HC PRM RCT TIM MAX OPT Date Time DBHost
B 000 000 R/3 000000000 ACTIVE NO NO YES NO 000 255 255 20080731 085702 Id1032
C INFO : SAP RELEASE (DB) = 710
M ThInit: db_connect o.K.
M ICT: exclude compression: *.zip,*.rar,*.arj,*.z,*.gz,*.tar,*.lzh,*.cab,*.hqx,*.ace,*.jar,*.ear,*.war,*.css,*.pdf,*.gzip,*.uue,*.b
```

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 13

The developer traces are not written directly by the database, but rather log actions of the disp+work processes of the SAP system.

The dev logs are stored in the work directory of the SAP system and have the designation **dev_w***. You can access them directly with transaction st11.

They are active by default; only higher trace levels have to be activated explicitly. This is generally done by the developers themselves.

However, they were included in this unit because other information relevant for the database is also stored there.

If connection problems between the disp+work processes and the DB occur, developer traces are often helpful.

Errors have a red background and thus are easy to find.

Among other things, it is easy to find the version of SQLDBC or the precompiler environment being used.

Dev Logs (2)



Trace Edit Goto System Help



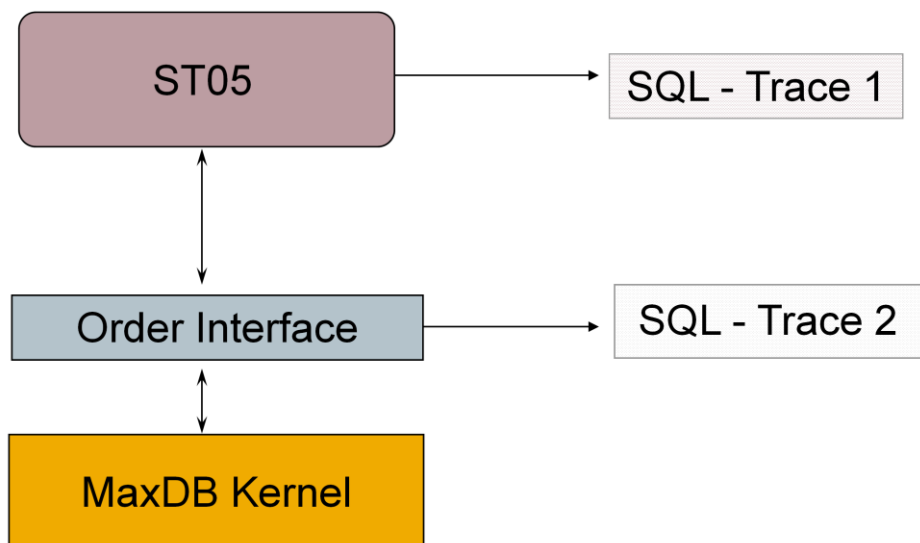
Trace Data: pdf2763_S70_50 dev_w0

Display Components

```
B 000 007 SDB_INTER      000006935 INACTIVE    NO NO NO 004 255 255 20080811 180322 pdfdf0759
B 000 008 DEM00B        000006802 DISCONNECTED NO NO NO 000 255 255 20080814 162319 BERD00146828A
B 000 009 MPW           000006936 ACTIVE      NO NO NO 004 255 255 20080814 174033 BERD00184088A
B 000 010 S70          001000125 DISCONNECTED NO NO NO 000 255 255 20080813 152816 pdfdf2763
C
C Thu Aug 14 17:41:23 2008
C *** ERROR => prepare() of C_0773, rc=1, rcSQL=-4005 (POS(33) Unknown column name:NUMERIC_VALUE)
[dbdsbsq1.cpp 1488]
C *** ERROR => PREPARE on connection 8, rc=-4005 (POS(33) Unknown column name:NUMERIC_VALUE)
[dbdsbdb.cpp 1911]
C sc_p=048A4874,no=773,idc_p=00000000,con=0,act=0,slen=71,smax=256,#vars=0,stmt=0817E498,table= SYSDBA.CONFIGURATION
C SELECT DESCRIPTION, CHAR_VALUE, NUMERIC_VALUE FROM SYSDBA.CONFIGURATION;
C *** ERROR => dbdsbdb: SQL Error -4005, Message: POS(33) Unknown column name:NUMERIC_VALUE
[dbdsbdb.cpp 2403]
B ***LOG BY2=> sq] error -4005 performing OPC [dbds#1 @ 482] [dbds 0482 ]
B ***LOG BY0=> POS(33) Unknown column name:NUMERIC_VALUE [dbds#1 @ 482] [dbds 0482 ]
A TH VERBOSE LEVEL FULL
A ** RABAX: level LEV_RX_PXA_RELEASE_MTX entered.
A ** RABAX: level LEV_RX_PXA_RELEASE_MTX completed.
A ** RABAX: level LEV_RX_COVERAGE_ANALYSER entered.
A ** RABAX: level LEV_RX_COVERAGE_ANALYSER completed.
A ** RABAX: level LEV_RX_ROLLBACK entered.
A ** RABAX: level LEV_RX_ROLLBACK completed.
A ** RABAX: level LEV_RX_DB_ALIVE entered.
A ** RABAX: level LEV_RX_DB_ALIVE completed.
A ** RABAX: level LEV_RX_HOOKS entered.
A ** RABAX: level LEV_RX_HOOKS completed.
A ** RABAX: level LEV_RX_STANDARD entered.
A ** RABAX: level LEV_RX_STANDARD completed.
A ** RABAX: level LEV_RX_STOR_VALUES entered.
A ** RABAX: level LEV_RX_STOR_VALUES completed.
A ** RABAX: level LEV_RX_C_STACK entered.
```



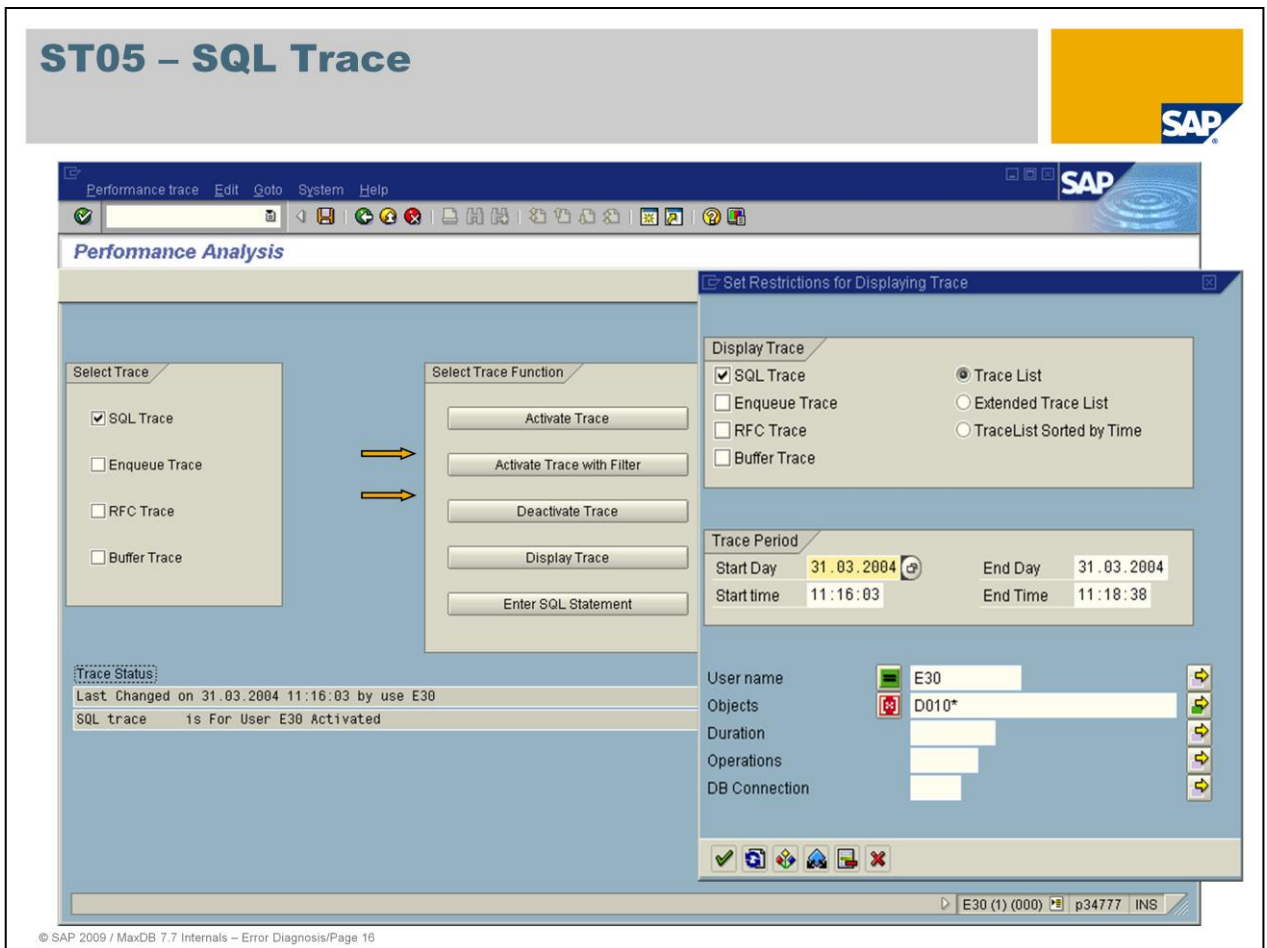
S70 (1) (000) pdf2763 | INS



In the SAP system, you activate the SQL trace with transaction st05. The log is written by the database interface. Along with the statements, you'll find the variables, their values and the runtime. The Explain button in transaction st05 displays the database's Optimizer strategy for the command.

This transaction is discussed further in the section on SAP system transactions that are useful for error analysis.

The order interface of the database also writes an SQL trace. New versions of the WebAS ABAP kernel use the new interface SQLDBC instead of the Precompiler starting with version 6.20.



You can use transaction st05 to activate an SQL trace. This is useful for performance analyses or for identifying a command that leads to incorrect result sets.

The SQL trace displays all SQL statements in the form in which they were sent to the database.

Note that when you use the button shown here, the trace must be deactivated before being displayed so that it can be formatted. You can display the trace directly by choosing *Performance Trace->Display Trace Or Deactivate First*

For a more manageable amount of information, you can restrict the display to a specific time period or a particular user or by omitting information about specified tables.

ST05 – SQL Trace (2)



Trace List

DDIC information Explain

Transaction	Work process no	Proc. type	DIA	Client	User
ST05	0	DIA		000	E30

Duration	Obj. name	Op.	Recs.	RC	Statement
1.308	D347T	REEXEC	1	0	SELECT WHERE "PROGNAME" = 'SAPLSS00' AND "SPRSL" = 'E' AND "OBJ_CODE" = '000'
745		EXECSTA	0	0	COMMIT WORK ON CONNECTION 0

Transaction	Work process no	Proc. type	DIA	Client	User
SESSION_MANAGER	1	DIA		000	E30

Duration	Obj. name	Op.	Recs.	RC	Statement
49.386	D347T	REEXEC	1	0	SELECT WHERE "PROGNAME" = 'MENUS000' AND "SPRSL" = 'E' AND "OBJ_CODE" = '000'
17.175	AGR_USERS	REOPEN	0	0	SELECT WHERE "MANDT" = '000' AND "UNAME" = 'E30'
41.024	AGR_USERS	FETCH	22	100	
16.244	USR21	REEXEC	1	0	SELECT WHERE "MANDT" = '000' AND "BNAME" = 'E30' AND ROWNUM <= 1
3.276	CVERS	REOPEN	0	0	SELECT FROM "CVERS" ORDER BY "COMPONENT"
360	CVERS	FETCH	2	100	
1.609	V_ADDR_USR	REOPEN	0	0	SELECT WHERE "CLIENT" = '000' AND "ADDRNUMBER" = '0000010005' AND "PERSNUMBER" = '0000010005'
232.174	V_ADDR_USR	FETCH	1	0	
9.099	WWWPARAMS	REEXEC	1	0	SELECT WHERE "RELID" = 'MI' AND "OBJID" = 'SESS_IMAGE' AND "NAME" = 'version' AND "LANGU" = 'EN'
605	LICENSE	REEXEC	1	0	SELECT FROM MLCHECK
869	LICENSE	REOPEN	0	0	SELECT FROM MLCHECK
399	LICENSE	FETCH	2	100	
164	AGR_USERS	REOPEN	0	0	SELECT WHERE "MANDT" = '000' AND "UNAME" = 'E30'
3.730	AGR_USERS	FETCH	22	100	

Transaction ? Work process no 1 Proc. type DIA Client 000 User E30

E30 (1) (000) p34777 INS

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 17

The list of commands identifies which transaction was started, which work process is affected, its type, as well as the client and user.

The command list contains

- information about the duration of the command, where the long-running ones are marked red,
- the affected database object,
- the number of records found,
- the return code of the database and
- the statement.

The statement can be expanded by double-click. It is also possible to insert parameter values in the placeholders so that the statement can be used directly for further testing, for example in the SQL Studio.

The 'Explain' button displays the execution plan of the Optimizer.



SQL – Trace of the order interface:

Profile parameter

■ <code>db/ada/sql_trace =</code>	<code>0</code>	<code>no trace</code>
	<code>1</code>	<code>short trace</code>
	<code>2</code>	<code>long trace</code>

Environment variable

■ <code>SQLOPT =</code>	<code>-F</code>	<code>file name</code>
	<code>-T</code>	<code>short trace</code>
	<code>-X</code>	<code>long trace</code>
	<code>-Y</code>	<code>statement count</code>

Irtrace

- `irtrace -p all -t <trace type>`
- `irtrace -p <prozess-id> -t <trace type>`

The order interface trace is set for Disp+Work processes using a profile parameter. On Windows systems, after changing the profile parameter, only the work process has to be restarted. On Unix systems, the SAP system or the affected application server has to be restarted. The trace files are stored in the work directory of the SAP instance. The name is comprised of the process ID of the work process and the ending `pct`.

Other tools that utilize the order interface read the environment variable `SQLOPT`. Unless otherwise specified with the `-F` option, the trace file is written to the current directory. The name is comprised of the name of the corresponding C module and the ending `pct`.

You can use `irtrace` to activate the trace without needing to restart the system/application server.

The tool gives you the following options for changing the trace:

- Activate/deactivate/switch trace for a particular process:
`irtrace -p <process id> -t <trace type>`
- The following trace types are available:
long
short
off
- Activating/deactivating the trace for all interface processes on the application server:
`irtrace -p all -t <trace type>`

Precompiler Trace (2)



PRODUCT : liveCache C/C++ Precompiler Runtime
VERSION : 7.1.4
BUILD : 032-000-055-840

version : P_1, P_2

SQL STATEMENT : FROM MODULE : dbslada AT LINE : 4186
OUTPUT : LZU : NT/INTEL 7.1.4 Build 032-000-055-840
OUTPUT : PCR : C-PreComp 7.1.4 Build 032-000-055-840
START : DATE : 2001-07-13 TIME : 0013:01:01
END : DATE : 2001-07-13 TIME : 0013:01:01

SESSION : 1;

SQLMODE : SAPR3 AT DATABASE : DB_000

SERVERDB : S10

SERVERNODE:

OPTION-CONNECT :

CONNECT "SAPR3 " IDENTIFIED BY :A SQLMODE SAPR3 ISOLATION LEVEL 0
TIMEOUT 0

SQL STATEMENT : FROM MODULE : dbslada AT LINE : 6390

START : DATE : 2001-07-13 TIME : 0013:01:01
END : DATE : 2001-07-13 TIME : 0013:01:01

Runtime library libSQLDBC for the development of database applications and interfaces

- Version-independent runtime library libSQLDBC.dll or libSQLDBC.so
- Version-dependent, e.g. libSQLDBC76.dll
- Runtime library libSQLDBC_C.dll for the support of native C applications
- Stored in directory <indep_program_path>/pgm

Software Development Kit SQLBDC SDK

- Header file SQLDBC.h for C++
- Header file SQLDBC_C.h for C
- Static and dynamic link libraries for C and C++
- Stored in directory <indep_program_path>/sdk/sqldbc

sqldbc_cons

- Creation and control of traces

SQL Database Connectivity (SQLDBC) is a runtime environment for the development of database applications and database interfaces for MaxDB. Through SQLDBC, applications can access MaxDB database instances, execute SQL statements and edit data. SQLDBC is comprised of the three abovementioned components, which are part of the standard and stored in the said directories.

Traces can be created either directly with sqldbc_cons or using transaction db50.

Configuration and creation of traces for the SQLDBC interface

Properties of the traces:

- sqldbctrace_<pid>.prt
- The trace files are overwritten cyclically.

Traces storage in the following directories:

- <user_home>\Application Data\sdb (Windows)
- <user_home>/.sdb (UNIX, Linux)

Trace types:

- SQL: SQL statements (sufficient for most analyses)
- Short: method calls
- Long: method calls with call parameters (most comprehensive trace)
- Packet: communication packages

Syntax: sqldbc_cons [<command>] [<option>]

sqldbc_cons is a tool for the configuration and control of traces.

The trace files contain a file name of the form sqldbctrace-<pid>.prt, where <pid> is the process ID. It is also possible to choose a name; %p in the name is replaced by the process ID. Traces are stored in the directories <user_home>\Application Data\sdb (Windows) and <user_home>\.sdb (UNIX, Linux). When the configured trace size is reached, the trace is cyclically overwritten.

Possible commands for sqldbc_cons:

TRACE SQL ON/OFF:	Turns the SQL trace on/off
TRACE PACKET ON/OFF:	Turns the PACKET trace on/off
TRACE SHORT ON/OFF:	Turns the SHORT trace on/off
TRACE LONG ON/OFF:	Switches the detailed LONG trace on/off
TRACE OFF:	Switches all SQLDBC traces off
TRACE FILENAME <file_name>:	determines the name of the trace file
TRACE SIZE <size>:	defines the size (in Bytes) of the trace file
SHOW ALL:	displays the configuration of the traces and current information about the traces
SHOW CONFIG:	displays the configuration of the traces
SHOW TRACESETTINGS	displays current information about the traces

Possible options for sqldbc_cons:

- f: forces the execution of the command
 - h: Help information
 - p <pid>: executes the command only for the process with the process ID <pid>
 - v: displays detailed information (verbose)
- (These options cannot be combined, but only used individually.)

In transaction db50, choose the path *Tools-> SQLDBC Trace*.

Activating the trace involves three steps:

- Selection of the desired process
- Selection of the trace type (SQL, Short, Long, Packet)
- Specification of trace size (*Goto-> Maximum File Size*)

To switch the trace off, select the process and press the button *Switch off*.

Via menu item *Goto-> Trace Directory* you can choose a trace file name that differs from the default.

To display the trace, select the trace file and press the button *Display File*.

Example 1:

```
06.01 12:53:46 18286 -11205 sqlexec:system error, not enough space
06.01 12:58:25 18286 -11205 sqlexec:system error, not enough space
06.01 18:46:23 19025 -11109 database 'S10' is not running
07.01 11:47:37 10959 -11987 sql33_con_msl: task limit
07.01 11:47:59 12031 -11987 sql33_request: connection broken, kernel
    cleared
08.02 13:11:07 18899 -11987 connection closed by communication partner
08.29 13:13:13 11199 -11987 comseg given away, assuming timeout
```



Example 2:

```
2008-01-25 10:28:55 8323 ERR -11987 COMMUNIC kernel aborted connection!
2008-01-25 10:28:55 8325 ERR -11987 COMMUNIC kernel aborted connection!
2008-02-18 10:03:13 18474 ERR -11608 COMMUNIC sql03_request: wrong connection st
ate, state is 'requested'
2008-03-17 16:00:07 26429 ERR -11987 COMMUNIC semid 206602379 disappeared!
2008-03-17 16:00:13 9231 ERR -11987 COMMUNIC semctl (setval 1084391433) error: Invalid argument
2008-03-17 16:00:13 9232 ERR -11987 COMMUNIC semctl (setval 1084391433) error: Invalid argument
2008-03-17 16:01:55 9237 ERR -11987 COMMUNIC semctl (setval 1084391433) error:
Identifier removed
2008-04-11 08:00:43 16194 ERR -11608 COMMUNIC sql03_request: wrong connection st
ate, state is 'requested'
2008-06-06 19:00:34 5216 ERR -11608 COMMUNIC sql03_request: wrong connection st
ate, state is 'requested'
```

The file **appliediag** contains error messages that can occur during communication between the applications and the runtime environment.

The file **appliediag** or **<pid>.dia** (SAP WebAS) is stored in the directory

<indepdatapath>/wrk/<unix user>	(UNIX)
<indepdatapath>/wrk	(Windows)
/usr/sap/<SID>/D*/work	(SAP WebAS)

The <indepdatapath> can be determined using the following dbmcli command:

```
dbmcli -d <SID> -u <dbm-user>,<password> dbm_getpath indepdatapath
```

Under Windows, **appliediag** is only activated if the environment variable is set to DIAGFILE=yes.

The file **appliediag** can get very large since it is not cyclically overwritten.

If this file already exists, further messages are added to it; otherwise it is created.

```

xserver.prt - Notepad
File Edit Format Help
Directory /sapdb/data/wrk
Name: xserver_p34777.prt
-----
Date      Time      PID Typ  MsgID  Label      Message-Text
-----
2004-03-09 16:44:54 1504 12902 XSERVER started, 'x32/LINUX 7.5.0 Build 010-111-066-035'
2004-03-09 16:44:54 1504 12904 XSERVER service port is 7210
2004-03-09 16:44:54 1504 12922 ENVIRON Command line arguments
2004-03-09 16:44:54 1504 12923 ENVIRON Command line argument dump completed
2004-03-09 16:44:54 1504 12898 ENVIRON Resource limit dump start
2004-03-09 16:44:54 1504 12898 ENVIRON Started by user id 1724 group id 79
2004-03-09 16:44:54 1504 12898 ENVIRON Current user id 3488 effective id 3488
2004-03-09 16:44:54 1504 12898 ENVIRON Current group id 1008 effective id 1008
2004-03-09 16:44:54 1504 12898 ENVIRON cpu time unlimited
2004-03-09 16:44:54 1504 12898 ENVIRON number of processes 4096
2004-03-09 16:44:54 1504 12898 ENVIRON number of open files 1024
2004-03-09 16:44:54 1504 12898 ENVIRON core size 0 kbytes
2004-03-09 16:44:54 1504 12898 ENVIRON file size unlimited
2004-03-09 16:44:54 1504 12898 ENVIRON heap memory size unlimited
2004-03-09 16:44:54 1504 12898 ENVIRON stack memory size unlimited
2004-03-09 16:44:54 1504 12898 ENVIRON lockable memory size unlimited
2004-03-09 16:44:54 1504 12898 ENVIRON virtual memory size unlimited
2004-03-09 16:44:54 1504 12898 ENVIRON resident set size unlimited
2004-03-09 16:44:54 1504 12898 ENVIRON Resource limit dump completed
2004-03-09 16:44:54 1504 12898 ENVIRON Environment dump start
2004-03-09 16:44:54 1504 12898 ENVIRON LESSKEY=/etc/lesskey.bin
2004-03-09 16:44:54 1504 12898 ENVIRON DIR_LIBRARY=/usr/sap/E30/sys/exe/run
2004-03-09 16:44:54 1504 12898 ENVIRON NITPSERVER=news
2004-03-09 16:44:54 1504 12898 ENVIRON INFODIR=/usr/share/info:/usr/local/info
2004-03-09 16:44:54 1504 12898 ENVIRON MANPATH=/opt/gnome/man:/usr/share/man:/u
2004-03-09 16:44:54 1504 12898 ENVIRON sr/x11r6/man:/usr/local/man
2004-03-09 16:44:54 1504 12898 ENVIRON KDE_MULTIMEDIA=false
2004-03-09 16:44:54 1504 12898 ENVIRON HOSTNAME=p34777
2004-03-09 16:44:54 1504 12898 ENVIRON XKEYSYMDB=/usr/x11r6/lib/x11/xkeysymDB
2004-03-09 16:44:54 1504 12898 ENVIRON SAPDBHOST=p34777
2004-03-09 16:44:54 1504 12898 ENVIRON dbms_type=ADA
2004-03-09 16:44:54 1504 12898 ENVIRON HOST=p34777
2004-03-09 16:44:54 1504 12898 ENVIRON TERM=xterm
2004-03-09 16:44:54 1504 12898 ENVIRON SHELL=/bin/tcsh
2004-03-09 16:44:54 1504 12898 ENVIRON PROFILEREAD=true
2004-03-09 16:44:54 1504 12898 ENVIRON XDM_MANAGED=/var/run/xdmctl/xdmctl-:0
2004-03-09 16:44:54 1504 12898 ENVIRON GS_LIB=/share/Fonts
2004-03-09 16:44:54 1504 12898 ENVIRON GNOME_PATH=/opt/gnome:/usr
2004-03-09 16:44:54 1504 12898 ENVIRON GTK_RC_FILES=/etc/gtk/gtkrc:/home/e30adm
2004-03-09 16:44:54 1504 12898 ENVIRON /.gtkrc
2004-03-09 16:44:54 1504 12898 ENVIRON KDEHOME=/home/e30adm/.kde
2004-03-09 16:44:54 1504 12898 ENVIRON MORE=-s1
2004-03-09 16:44:54 1504 12898 ENVIRON DBNAME=p34777:E30
2004-03-09 16:44:54 1504 12898 ENVIRON XSESSION_IS_UP=yes
2004-03-09 16:44:54 1504 12898 ENVIRON JRE_HOME=/usr/lib/java/jre
2004-03-09 16:44:54 1504 12898 ENVIRON USER=e30adm
2004-03-09 16:44:54 1504 12898 ENVIRON GROUP=sapsys
2004-03-09 16:44:54 1504 12898 ENVIRON LS_COLORS=no=00:fi=00:di=01:34:ln=00:36:

```

© SAP 2009 / MaxDB 7.7 Internals - Error Diagnosis/Page 24

The file **xserver_<hostname>.prt** contains error messages involving the communication via **x_server**. The **x_server** is used for remote communication and starts **vserver** processes for each new user who connects to the database remotely.

If multiple database software versions are installed on a computer, the **x_server** must always be started with the highest version. You can display the highest version with

```
x_server -V
```

You can display the possible options for installing, starting and stopping with

```
x_server -h
```

If, f.e., there are network problems between the application and database server error messages are written to this log file.

The start information (see slide) contains additional information about operating system settings that are significant for database operation.

A time stamp, a process ID, in the label the affected software component and an explanatory message text are delivered.

If a return code is reported by the operating system, its meaning can be determined with

```
xsysrc <rc>
```

The **xserver_<hostname>.prt** is stored in the directory **<indepdatapath>/wrk**.

Access via DB50: ***Problem Analysis -> Messages -> Kernel – Remote SQL Server***

Access via Database Studio: ***Diagnosis Files – XServer Messages / Xserver Messages (OLD)***

```

dbm.prt - Notepad
File Edit Format Help
-----
Date      Time      TID(hex)  Typ  MsgID    Label  Message
-----
[...]
2003-08-06 09:43:17 0x00004999 ERR  -24964 DBM  32512,/sapdb/E30/db/bin/xload -S INTERNAL -n p34777 -d E30 -u SUPERDBA,* -b /sapdb/E30/db/,
0 DBM  command load_r3tab
2003-08-06 09:43:18 0x000049A0 ERR  -24964 DBM  ERR_EXECUTE: error in program execution
0 DBM  ERR_EXECUTE: error in program execution
2003-08-06 17:24:36 0x000053e2 0 DBM  command backup_save "data_tmp" DATA RECOVERY
0 DBM  command backup_save "data_tmp" DATA RECOVERY
2003-08-07 11:21:27 0x00002d00 0 DBM  command param_startsession
0 DBM  command param_startsession
2003-08-07 11:21:33 0x00002d00 0 DBM  command param_abortsession
0 DBM  command param_abortsession
2003-08-07 16:56:30 0x000062a2 0 DBM  command backup_media_put "save_data" "d:\tmp\data" FILE DATA 0 0 NO ""
0 DBM  command backup_media_put "save_data" "d:\tmp\data" FILE DATA 0 0 NO ""
2003-08-07 16:58:50 0x000062a2 0 DBM  command backup_label
0 DBM  command backup_label
2003-08-07 16:58:50 0x000062a2 ERR  -24988 DBM  ERR_SQL: sql error
-24988 DBM  -903,Host File I/O error,Could not open devspace
2003-08-08 07:40:11 0x00006b48 0 DBM  command backup_save "data" DATA RECOVERY
0 DBM  command backup_save "data" DATA RECOVERY
2003-08-08 07:41:35 0x00006b48 ERR  -24988 DBM  ERR_SQL: sql error
-24988 DBM  -903,Message not available,Could not open devspace
2003-08-08 07:41:57 0x00006b70 0 DBM  command backup_save "data_tmp" DATA RECOVERY
0 DBM  command backup_save "data_tmp" DATA RECOVERY
2003-08-13 10:43:23 0x00001574 0 DBM  command sql_updatestat * ESTIMATE SAMPLE 1000 ROWS
0 DBM  command sql_updatestat * ESTIMATE SAMPLE 1000 ROWS
2003-08-13 10:50:30 0x00001574 0 DBM  command sql_updatestat * ESTIMATE SAMPLE 20000 ROWS
0 DBM  command sql_updatestat * ESTIMATE SAMPLE 20000 ROWS
2003-08-13 11:09:27 0x00001574 0 DBM  command sql_updatestat * ESTIMATE SAMPLE 20000 ROWS
0 DBM  command sql_updatestat * ESTIMATE SAMPLE 20000 ROWS
2003-08-14 12:40:31 0x00002645 0 DBM  command param_startsession
0 DBM  command param_startsession
2003-08-14 12:40:59 0x00002645 0 DBM  command param_directput CACHESIZE 21000
0 DBM  command param_directput CACHESIZE 21000
2003-08-14 12:40:59 0x00002645 ERR  -24974 DBM  ERR_XPSESSION: param session active
-24974 DBM  ERR_XPSESSION: param session active
2003-08-14 12:41:17 0x00002645 0 DBM  command param_put CACHE_SIZE 21000
0 DBM  command param_put CACHE_SIZE 21000
2003-08-14 12:41:41 0x00002645 0 DBM  command param_commitssession
0 DBM  command param_commitssession
2003-08-14 12:41:41 0x00002645 ERR  -24973 DBM  ERR_XPCHECK: param check failure/request
-24973 DBM  ERR_XPCHECK: param check failure/request
[...]
2003-10-24 10:57:13 0x0000478a 0 DBM  command db_offline
0 DBM  command db_offline
2003-10-24 11:37:55 0x00004b46 0 DBM  command db_clear
0 DBM  command db_clear
2003-10-24 11:37:55 0x00004b49 0 DBM  command db_start
0 DBM  command db_start
2003-10-24 11:38:02 0x00004bad 0 DBM  command db_warm
0 DBM  command db_warm
2003-10-24 11:45:28 0x00004ec3 0 DBM  command file_getfirst
0 DBM  command file_getfirst
2003-10-24 11:45:28 0x00004ec3 ERR  -24994 DBM  ERR_RTE: runtime environment error
-24994 DBM  ERR_RTE: runtime environment error
2003-10-24 11:45:44 0x00004e6f 0 DBM  command dban_stop
0 DBM  command dban_stop
2003-10-24 11:45:44 0x00004e6f ERR  -24964 DBM  ERR_EXECUTE: error in program execution
-24964 DBM  ERR_EXECUTE: error in program execution
2003-10-24 11:45:44 0x00004e6f ERR  -24964 DBM  256,dbanalyzer -d E30 -u *,* -stop
-24964 DBM  256,dbanalyzer -d E30 -u *,* -stop
[...]
2003-11-27 16:21:46 0x00002001 0 DBM  command util_execute diagnose monitor CLEAR
0 DBM  command util_execute diagnose monitor CLEAR
2003-11-27 16:22:12 0x00002005 0 DBM  command util_execute diagnose monitor READ OFF
0 DBM  command util_execute diagnose monitor READ OFF
2003-11-27 16:22:13 0x00002009 0 DBM  command util_execute diagnose monitor TIME OFF
0 DBM  command util_execute diagnose monitor TIME OFF
2003-11-27 16:22:14 0x0000200d 0 DBM  command util_execute diagnose monitor SELECTIVITY 100
0 DBM  command util_execute diagnose monitor SELECTIVITY 100
2003-11-27 16:22:15 0x00002011 0 DBM  command util_execute diagnose monitor DATA ON
0 DBM  command util_execute diagnose monitor DATA ON
2003-11-27 16:22:16 0x00002015 0 DBM  command util_execute diagnose monitor ROWNO 255 100
0 DBM  command util_execute diagnose monitor ROWNO 255 100
2003-11-27 16:28:22 0x00002019 0 DBM  command util_execute diagnose monitor OFF
0 DBM  command util_execute diagnose monitor OFF

```

© SAP 2009 / MaxDB 7.7 Internals - Error Diagnosis/Page 25

The Database Manager log **dbm.prt** comprises the command history of the Database Manager. All change actions and all actions that return error messages are logged.

Because messages show the exact date and time, they can easily be referenced against the outputs of other log files.

If errors occur in the action being executed, they are marked ERR.

The file is stored in the run directory (default: <indepdatapath>/wrk/<SID>).

Access via DBMGUI: **Check -> Diagnosis Files -> Database Manager Log File (DBMPRT)**

Access via DB50: **Problem Analysis -> Messages -> Database Manager**

Access via Database Studio: **Diagnosis Files - Database Manager Log File**

```

ld1032_e70_dbm.utl.txt - Notepad
File Edit Format View Help
-----
Date      Time      Message-Text
-----
2006-03-20 18:07:58 441EE16E0001 0000 ICL CREATE INSTANCE SYSDBA SUPERDBA
2006-03-20 18:15:05 441EE16E0001 0001 RET RETURNCODE 0
2006-03-20 18:17:11 441EE3970004 0000 SHT SHUTDOWN
2006-03-20 18:17:19 441EE39F0001 0000 REQ SET LOG WRITER OFF
2006-03-20 18:17:19 441EE39F0001 0001 RET RETURNCODE 0
2006-03-20 18:17:19 441EE39F0003 0000 RST RESTART
2006-03-20 18:17:20 441EE39F0003 0001 RET RETURNCODE 0
2006-03-20 18:41:58 441EE9660004 0000 SHT SHUTDOWN
2006-03-20 18:42:07 441EE96F0001 0000 REQ SET LOG WRITER ON
2006-03-20 18:42:07 441EE96F0001 0001 RET RETURNCODE 0

[...]

2006-03-24 17:25:21 44241D710003 0000 SDB SAVE DATA QUICK TO '/sapdb/E70/saparch/E70_COM' FILE BLOCKSIZE 8 NO CHECKPOINT MEDIANAME
2006-03-24 17:25:21 44241D710003 0001 RET RETURNCODE -903
2006-03-24 17:25:40 0000 SDB SAVE DATA CANCEL
2006-03-24 17:25:40 0001 RET RETURNCODE -104
2006-03-24 17:25:40 44241D840004 0000 SDB SAVE DATA QUICK TO '/sapdb/E70/saparch/E70_COM' FILE BLOCKSIZE 8 NO CHECKPOINT MEDIANAME
2006-03-24 17:33:46 44241D840004 0001 RET RETURNCODE 0
2006-03-24 17:33:46 44241D840004 0002 TAP DATE..... 2006-03-24
2006-03-24 17:33:46 44241D840004 0003 TAP TIME..... 17:25:40
2006-03-24 17:33:46 44241D840004 0004 TAP SERVERID..... E70
2006-03-24 17:33:46 44241D840004 0005 TAP SERVERNODE..... ld1032
2006-03-24 17:33:46 44241D840004 0006 TAP KERNEL VERSION.... kernel 7.6.00 Build 018-123-119-055
2006-03-24 17:33:46 44241D840004 0007 TAP PAGES TRANSFERRED. 856784
2006-03-24 17:33:46 44241D840004 0008 TAP PAGES LEFT..... 0
2006-03-24 17:33:46 44241D840004 0009 TAP NO OF VOLUMES..... 1
2006-03-24 17:33:46 44241D840004 000A TAP MEDIA NAME..... Full
2006-03-24 17:33:46 44241D840004 000B TAP TAPE NAME..... /sapdb/E70/saparch/E70_COM
2006-03-24 17:33:46 44241D840004 000C TAP TAPE ERRORTEXT.... UNDEF
2006-03-24 17:33:46 44241D840004 000D TAP TAPE LABEL..... DAT_000000002
2006-03-24 17:33:46 44241D840004 000E TAP IS CONSISTENT..... TRUE
2006-03-24 17:33:46 44241D840004 000F TAP FIRST IO SEQUENCE. 128017
2006-03-24 17:33:46 44241D840004 0010 TAP LAST IO SEQUENCE.. UNDEF
2006-03-24 17:33:46 44241D840004 0011 TAP DBSTAMP1 DATE.... 2006-03-24
2006-03-24 17:33:46 44241D840004 0012 TAP DBSTAMP1 TIME.... 17:25:40
2006-03-24 17:33:46 44241D840004 0013 TAP DBSTAMP2 DATE.... UNDEF
2006-03-24 17:33:46 44241D840004 0014 TAP DBSTAMP2 TIME.... UNDEF
2006-03-24 17:33:46 44241D840004 0015 TAP BD PAGE COUNT.... 856760
2006-03-24 17:33:46 44241D840004 0016 TAP TAPEDEVICES USED.. 1
2006-03-24 17:33:46 44241D840004 0017 TAP DB_IDENT..... ld1032:E70_20060324_172540
2006-03-24 17:33:46 44241D840004 0018 TAP MAX USED DATA PNO. 0
2006-03-24 17:33:46 44241D840004 0019 TAP CONV PAGE COUNT... 464
2006-03-24 17:34:37 44241F9D0005 0000 SHT SHUTDOWN
2006-03-24 17:35:43 44241FDD0001 0000 RST RESTART
2006-03-24 17:35:45 44241FDD0001 0001 RET RETURNCODE 0

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 26

```

As of version 7.7 no special utility log file is written. The information is now written to the **dbm.prt**.

Up to version 7.6 all commands sent to the database kernel by the utility task are logged in the file **dbm.utl**. As of 7.5 user tasks executing utility statements also write into this file. The file is written by the database kernel.

This file contains detailed information about backup and restore processes, configuration changes such as the addition of volumes, information about update-statistics processes and so on.

In **dbm.utl** you can see whether operations have been successful from the point of view of the database kernel. When using external backup tools, it is important to take account of the corresponding log files as well, since errors can also occur on other levels during the transfer of backup information from the kernel to the tools.

The file is stored in the run directory of the database (default: <indepdatapath>/wrk/<SID>).

Access via DBMGUI: **Check -> Diagnosis Files -> Utility Statements (UTLPRT)**

Access via DB50: **Problem Analysis -> Logs -> Kernel Administration**

Access via Database Studio: **Diagnosis Files -> Utility Statements**

File Edit Format Help		hsap_dbmkn1.txt - Notepad																				
3ED21ALD0003			HISTLOST								2003-05-26 15:43:57											
3EE5FA7B0004	DAT_00001	SAVE	WARM	2003-06-10 17:34:19	2003-06-10 17:34:19	2003-06-10 17:34:21	2003-06-10 18:59:15	20981													NO	Data
3EE891540002	DAT_00001	RESTORE		2003-06-10 17:34:19	2003-06-10 17:34:19	2003-06-12 16:42:28	2003-06-12 16:53:37	20981													NO	Data_tmp
3EE8947A0004			HISTLOST								2003-06-12 16:55:54											
3EE894B70001	DAT_00002	SAVE	COLD	2003-06-12 16:56:22	2003-06-12 16:56:22	2003-06-12 16:56:57	2003-06-12 17:01:46	21057													NO	Data
3EE896F20001	DAT_00003	SAVE	COLD	2003-06-12 17:06:03	2003-06-12 17:06:03	2003-06-12 17:06:28	2003-06-12 17:10:58	21596													NO	Data
3EE89AB60001			HISTLOST								2003-06-12 17:22:30											
3EE898B60002	LOG_00000	SAVE	WARM	?	?	?	?	-1			2003-06-12 17:22:40	2003-06-12 17:25:15										
	LOG_00000	SAVE	WARM	?	?	?	?	-1			2003-06-12 17:25:58	2003-06-12 17:38:38										
	LOG_00000	SAVE	WARM	?	?	?	?	-1			2003-06-12 17:41:11	2003-06-12 17:41:15										
3EE8A8EE0003	DAT_00004	SAVE	WARM	2003-06-12 18:23:10	2003-06-12 18:23:10	2003-06-12 18:23:12	2003-06-12 18:34:05	1394													NO	Data_tmp
3F268EE70004	DAT_00005	SAVE	WARM	2003-07-29 17:12:40	2003-07-29 17:12:40	2003-07-29 17:12:43	2003-07-29 17:23:17	287309													NO	Data_tmp
3F2A01690006	LOG_00001	SAVE	WARM	2003-06-12 17:22:48	2003-06-29 02:12:56	2003-08-01 07:58:01	2003-08-01 07:58:46	0	103014													Log
3F2A01970007	LOG_00002	SAVE	WARM	2003-06-29 02:12:56	2003-07-16 07:33:39	2003-08-01 07:58:47	2003-08-01 07:59:35	103015	206147													Log
3F2A04430008	LOG_00003	SAVE	WARM	2003-07-16 07:38:39	2003-08-01 08:10:00	2003-08-01 08:10:11	2003-08-01 08:11:05	206148	312572													Log
3F2FA4DA0003	LOG_00004	SAVE	WARM	2003-08-01 08:10:01	2003-08-05 14:36:16	2003-08-05 14:36:42	2003-08-05 14:37:32	312573	388723													Log
3F30C9840004	LOG_00005	SAVE	WARM	2003-08-05 14:36:16	2003-08-06 11:25:41	2003-08-06 11:26:12	2003-08-06 11:27:15	388724	466692													Log
3F30F0800005	LOG_00006	SAVE	WARM	2003-08-06 11:25:43	2003-08-06 15:06:45	2003-08-06 15:07:12	2003-08-06 15:08:07	466693	529822													Log
3F3110B40006	DAT_00007	SAVE	WARM	2003-08-06 17:24:52	2003-08-06 17:24:52	2003-08-06 17:24:55	2003-08-06 17:39:47	542988													NO	Data_tmp
3F3337BB0007	DAT_00007	SAVE	WARM	2003-08-08 07:40:12	2003-08-08 07:40:12	2003-08-08 07:40:14	2003-08-08 07:41:35	549971													NO	Data
3F3338250008	DAT_00008	SAVE	WARM	2003-08-08 07:41:57	2003-08-08 07:41:57	2003-08-08 07:41:59	2003-08-08 07:56:27	549976													NO	Data_tmp
3F67A4480003	LOG_00007	SAVE	WARM	2003-08-06 15:06:45	2003-09-17 02:00:55	2003-09-17 02:01:12	2003-09-17 02:02:07	529823	749096													Log
3FC37C160003	LOG_00008	SAVE	WARM	2003-09-17 02:00:55	2003-11-25 16:57:49	2003-11-25 16:58:14	2003-11-25 16:59:32	749097	1007218													Log
401687F70004	LOG_00009	SAVE	WARM	2003-11-25 16:57:49	2004-01-27 16:46:42	2004-01-27 16:47:03	2004-01-27 16:48:04	1007219	1247065													Log
404074D40003	LOG_00010	SAVE	WARM	2004-01-27 16:46:42	2004-02-28 12:00:29	2004-02-28 12:00:36	2004-02-28 12:01:30	1247066	1399878													Log
404F0F090003	DAT_00009	SAVE	WARM	2004-03-10 13:50:37	2004-03-10 13:50:37	2004-03-10 13:50:41	2004-03-10 14:05:44	1435397													NO	Data_tmp

Two examples of complete lines:

=====

```
[LOG_00000|SAVE WARM|? ? |? ? |2003-06-12 17:22:40|2003-06-12 17:25:15| -1| -1| |
| 0| 0| -123|
```

```
3F3337BB0007|DAT_00007|SAVE WARM|2003-08-08 07:40:12|2003-08-08 07:40:12|2003-08-08 07:40:14|2003-08-08 07:41:35| 549971| |NO|Data
| 0| 0| -903|Could not open dev
```

The file **dbm.knl** contains a list of the backup and restore actions that have been executed.

The file is written by the database kernel.

You can identify what type of backup (DATA, LOG) was executed, in which time period the execution took place, up to which log page number the data was backed up, which medium was used and whether any errors occurred.

When using external backup tools, it is important to observe their logs as well, which are described in the following pages.

The file is stored in the run directory (default: <indepdatapath>/wrk/<SID>).

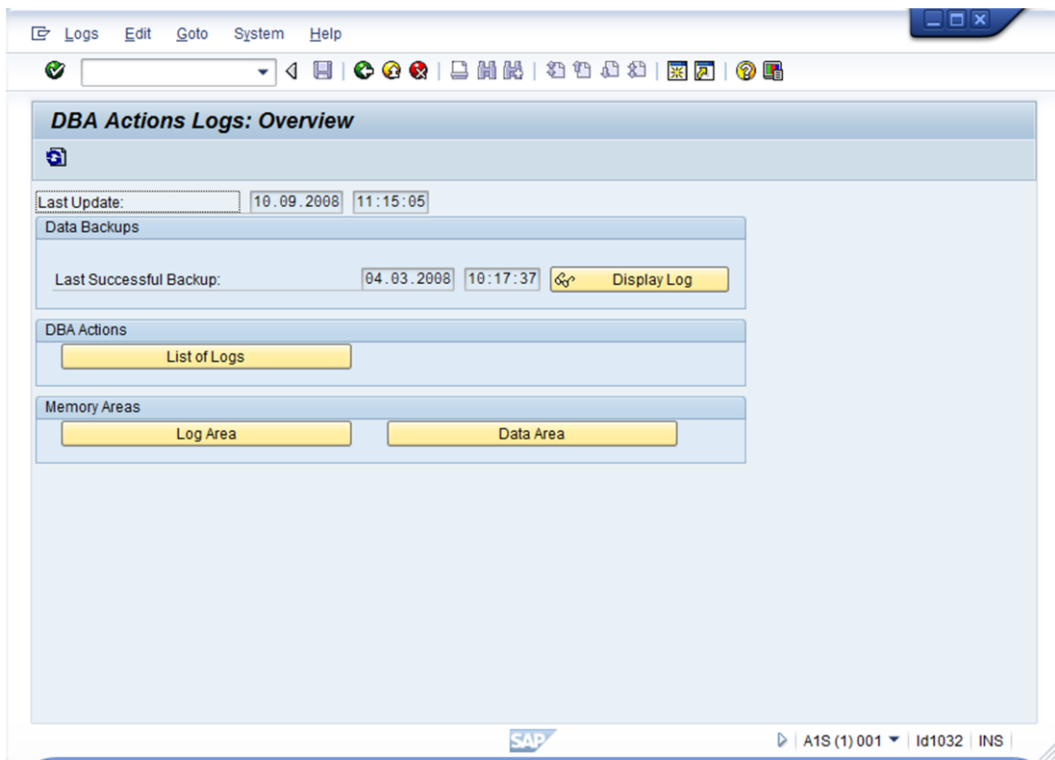
Access via DBMGUI: **Information -> Backup History** or **Check -> Diagnosis Files -> Backup History (BACKHIST)**

Access via DB50: **Problem Analysis -> Logs -> DBA History -> Backup/Restore (Kernel)**

Access via Database Studio: **Diagnosis Files -> Backup History**

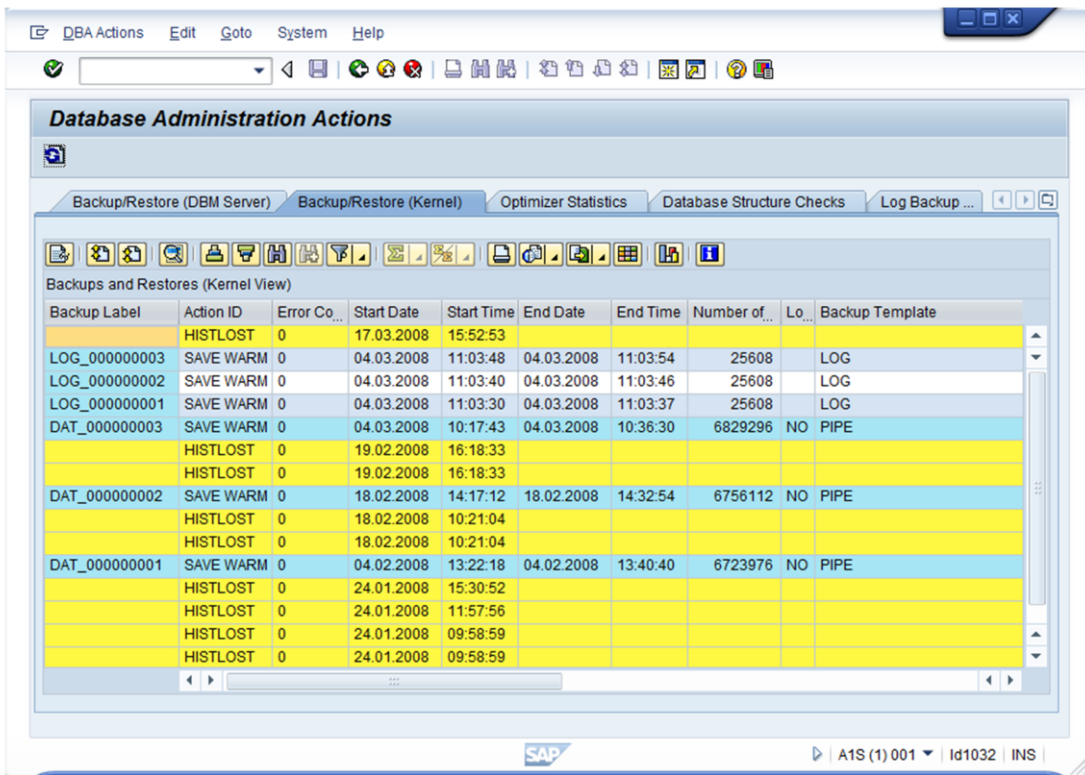
Because of the length of the output line, the file is somewhat difficult to work with; it is therefore a good idea to get a formatted display of the backup history with the Database Studio (Administration -> Backup) or transaction DB50. Errors are noted at the end of the output line.

The transactions DBACockpit (as of version 7.0), db12 and db13(c) use this information about backups to, for example, propose a recovery procedure.



SAP transaction db12 can be used to get an overview of backup and restore actions that have been executed.

Here you can also get information about the scope and frequency of Update Statistics operations as well as a history of consistency checks.

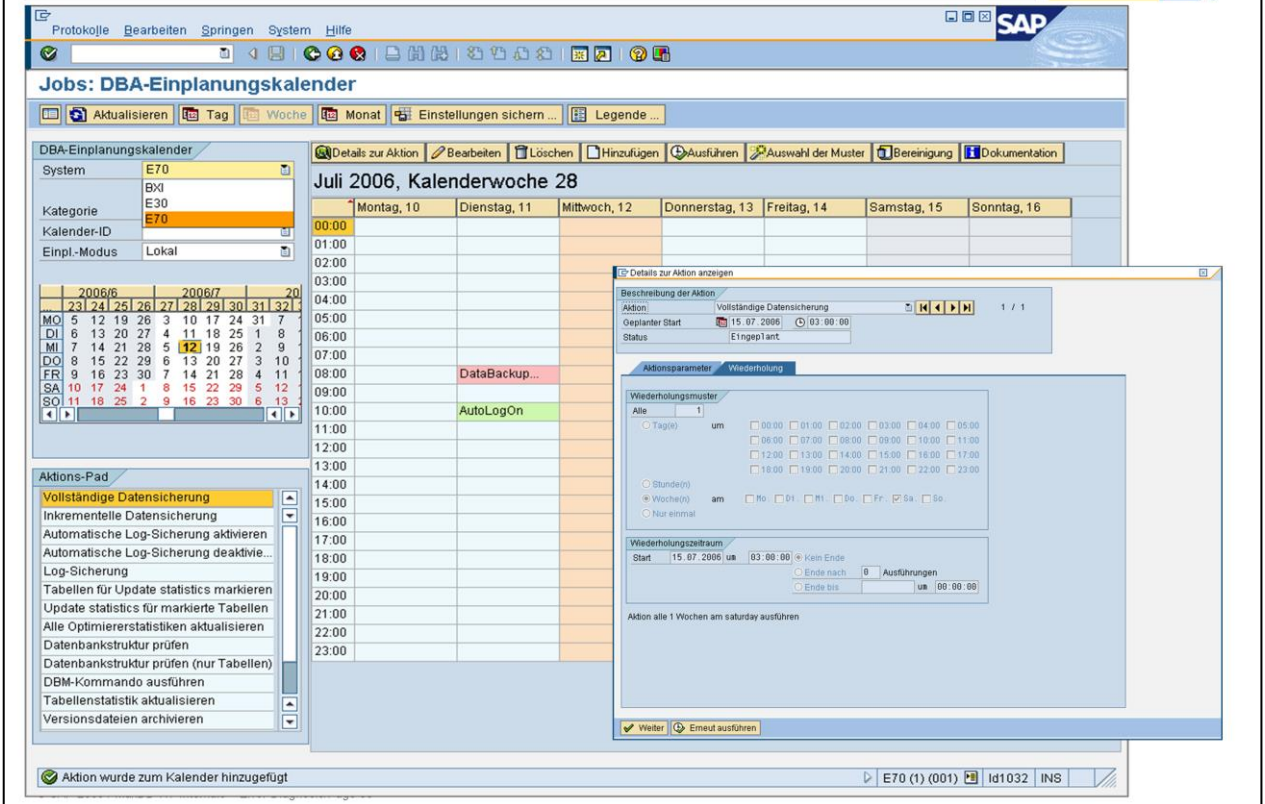


The screenshot shows the 'Database Administration Actions' window in SAP. The 'Backups and Restores (Kernel View)' tab is active, displaying a table with the following columns: Backup Label, Action ID, Error Co..., Start Date, Start Time, End Date, End Time, Number of..., Lo..., and Backup Template. The table contains several rows of backup and restore operations, including 'HISTLOST', 'SAVE WARM', and 'DAT_...' entries with their respective dates and times.

Backup Label	Action ID	Error Co...	Start Date	Start Time	End Date	End Time	Number of ...	Lo...	Backup Template
	HISTLOST	0	17.03.2008	15:52:53					
LOG_000000003	SAVE WARM	0	04.03.2008	11:03:48	04.03.2008	11:03:54	25608		LOG
LOG_000000002	SAVE WARM	0	04.03.2008	11:03:40	04.03.2008	11:03:46	25608		LOG
LOG_000000001	SAVE WARM	0	04.03.2008	11:03:30	04.03.2008	11:03:37	25608		LOG
DAT_000000003	SAVE WARM	0	04.03.2008	10:17:43	04.03.2008	10:36:30	6829296	NO	PIPE
	HISTLOST	0	19.02.2008	16:18:33					
	HISTLOST	0	19.02.2008	16:18:33					
DAT_000000002	SAVE WARM	0	18.02.2008	14:17:12	18.02.2008	14:32:54	6756112	NO	PIPE
	HISTLOST	0	18.02.2008	10:21:04					
	HISTLOST	0	18.02.2008	10:21:04					
DAT_000000001	SAVE WARM	0	04.02.2008	13:22:18	04.02.2008	13:40:40	6723976	NO	PIPE
	HISTLOST	0	24.01.2008	15:30:52					
	HISTLOST	0	24.01.2008	11:57:56					
	HISTLOST	0	24.01.2008	09:58:59					
	HISTLOST	0	24.01.2008	09:58:59					

Information from the file **dbm.knl** is optically presented which allows to recognize directly if there are failed backups or gaps in the backup history.

The output is generated when the DBM parameter DBATL is set to 1. For further information, see the Note **431508**.



db13 is the scheduling calendar for backups, Update Statistics and consistency checks. A weekly schedule can be used to plan the regular execution of activities.

Transaction DB13C is no longer required with WebAS 7.0 since transaction DB13 allows scheduling of activities for various instances. Integrate an instance with transaction DB59. Double-click to go to the database monitoring and via Tools -> DBA Planning Calendar to transaction DB13. Now the new instance will henceforth be known in transaction DB13.

Further information can be found in note **431508**.

It is also possible to call the scheduling calendar with transaction DBACockpit (Jobs -> DBA Planning Calendar).

If an error occurs during an action, it is displayed with a red background.

The causes of errors can be determined with the familiar diagnosis files. The job logs may also contain information that is useful in this regard.

External Backup Protocol (dbm.ebp)

- Is created by each action using a supported backup tool.
- Is overwritten with every start of the DBM server, if it communicates with an external backup tool.

Contents

- Configuration values
- Commands of the database kernel
- Call of the backup tools
- Returncodes of the backup tools and of the database kernel
- Output of the backup tools

For diagnosing problems with backups using external backup tools, the log file **dbm.ebp** plays a decisive role.

In addition to information about the configuration parameter of the tool, **dbm.ebp** contains information about the commands sent to the database kernel as well as the backup tool call. The error position makes it possible to identify who was responsible for the problem.

dbm.ebp is stored in the run directory of the database (default: <indepdatapath>/wrk/<SID>).

Access via DBMGUI: **Check -> Diagnosis Files -> External Backup Protocol**

Access via DB50: **Properties -> Files -> BACKEBP**

Access via Database Studio: **Diagnosis Files -> External Backup Protocol**

Note that this file is overwritten after each start of the DBM server when it communicates with the external backup tool. A new DBM server is started with each dbmcli call, to name one example.

External Backup Log

- Contains a summary of the last <n> dbm.ebp files.
- The number is configurable.

```

2004-01-16 18:10:51
----- Start of a new external backup tool action -----
2004-01-16 18:10:51
Setting environment variable 'TEMP' for the directory for temporary
files and pipes to default ''
Setting environment variable 'TMP' for the directory for temporary files
and pipes to default ''
Using connection to Legato's Networker with save, recover and mminfo.

2004-01-16 18:10:51
Checking existence and configuration of Networker.
Setting environment variable 'NSR_ENV' for the path of Networker's
configuration file to default '/nsr/sapdb/env'.
Found Networker setting for 'NSR_HOME': '/opt/nsr'.
Found Networker setting for 'NSR_HOST': 'hges100'.
Found Networker setting for 'NSR_FULL': 'SAVEDTNCCHK'.
Found Networker setting for 'NSR_EXPIRE': '30 days'.
Found Networker setting for 'NSR_IGNOREERROR': 'NSR server hges100.
sz.gesis.
de: busy'.
Found Networker setting for 'NSR_IGNOREERROR': 'save: waiting 30
seconds the
n retrying'.
Found Networker setting for 'NSR_SIZE ROUNDUP': 'NORMAL'.
Found Networker setting for 'NSR_SIZE ROUNDUP': 'NORMAL'.
Using Networker programs:
/opt/nsr/save
/opt/nsr/recover
/opt/nsr/mminfo
Check passed successfully.

```

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 32

Because the file **dbm.ebp** is promptly overwritten, there is a summary of it called **dbm.ebl**. This log file contains the last <n> logs, the number of which can be configured with the DBM parameter DBM_EBLSIZE.

The file **dbm.ebl** is stored in the run directory of the database ((default: <indepdatapath>/wrk/<SID>).

Access via DBMGUI: **Check -> Diagnosis Files -> External Backup Log**

Access via DB50: **Properties -> Files -> DBMEBL**

Access via Database Studio: **Diagnosis Files -> External Backup Log**

```

Backup-Fehler.txt - Notepad
File Edit Format Help

2004-01-16 18:10:51
Checking medium.
Check passed successfully.

2004-01-16 18:10:51
Preparing restore.
Constructed Networker call '/opt/nsr/recover -v
1074124992
-c hgessq01 -iy /nsr/sapdb/pipes6F'.
Created temporary file '/var/tmp/temp1074273051
Networker.
Created temporary file '/var/tmp/temp1074273051
for Networker.
Prepare passed successfully.

2004-01-16 18:10:51
Creating pipes for data transfer.
Creating pipe '/nsr/sapdb/pipes6F' ... Done.
All data transfer pipes have been created.
All data transfer pipes have been created.

2004-01-16 18:10:51
Starting database action for the restore.
Requesting 'RESTORE DATA QUICK FROM '/nsr/sapdb
BLOCKSIZE 8 M
EDIANAME 'nsr_full'' from db-kernel.
The database is working on the request.

2004-01-16 18:10:51
Starting Networker.
Starting Networker process '/opt/nsr/recover -v
1074124992
-c hgessq01 -iy /nsr/sapdb/pipes6F >>/var/tmp/temp1
2>>/var/tmp/temp1
074273051-1'.
Process was started successfully.
Networker has been started successfully.

2004-01-16 18:10:51
waiting for end of the restore operation.
2004-01-16 18:10:51 The backup tool is running.
2004-01-16 18:10:51 The database is working on the request.

2004-01-16 18:10:56 The backup tool is running.
2004-01-16 18:10:56 The database is working on the request.

2004-01-16 18:11:06 The backup tool is running.
2004-01-16 18:11:06 The database is working on the request.
[... ]
2004-01-16 20:26:54 The database has finished work on the
request.
Receiving a reply from the database kernel.
Got the following reply from db-kernel:
SQL-Code      :-9026 ←

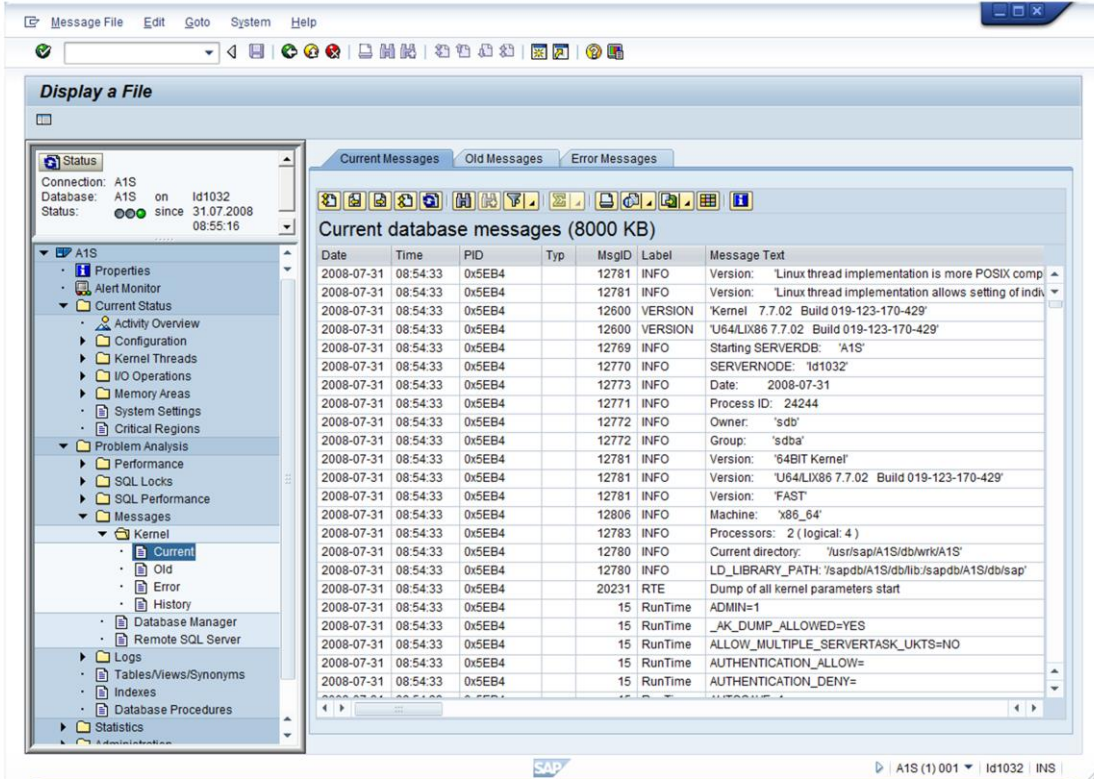
2004-01-16 20:26:54 The backup tool is running.

2004-01-16 20:27:00 The backup tool is running.
[... ]
2004-01-16 20:27:45 The backup tool is running.

2004-01-16 20:27:54 Canceling tool after the timeout
of 60 seconds elapsed.
Trying to break pipe '/nsr/sapdb/pipes6F'.
Pipe has been broken successfully.
The pipe '/nsr/sapdb/pipes6F' was removed.

2004-01-16 20:27:57 The backup tool is running.
[... ]
2004-01-16 20:28:50 The backup tool is running.

```

The file **KnIMsg** contains messages of the database kernel. It is recreated each time the database instance is started. The previous file is renamed to **KnIMsg.old**. The messages - apart from the header (start messages) - are overwritten cyclically.

Error messages are recorded in **KnIMsg** but also - due to the risk that they will be overwritten there - in the file **KnIMsgArchive (knldiag.err)**. This file is written continuously.

As of version 7.7 **KnIMsg** files replace the files **knldiag***. A specialty of the new files is that they are stored in an XML-like representation to make it possible in further states of expansion that together with the error messages directly instructions are delivered. This implies that the files – if you look at them on operating system level – have to be prepared before they can be displayed properly (see slide **protconv**). If you choose Database Studio, DBMGUI or transaction DB50 to display the **KnIMsg** the conversion to a readable format is done automatically.

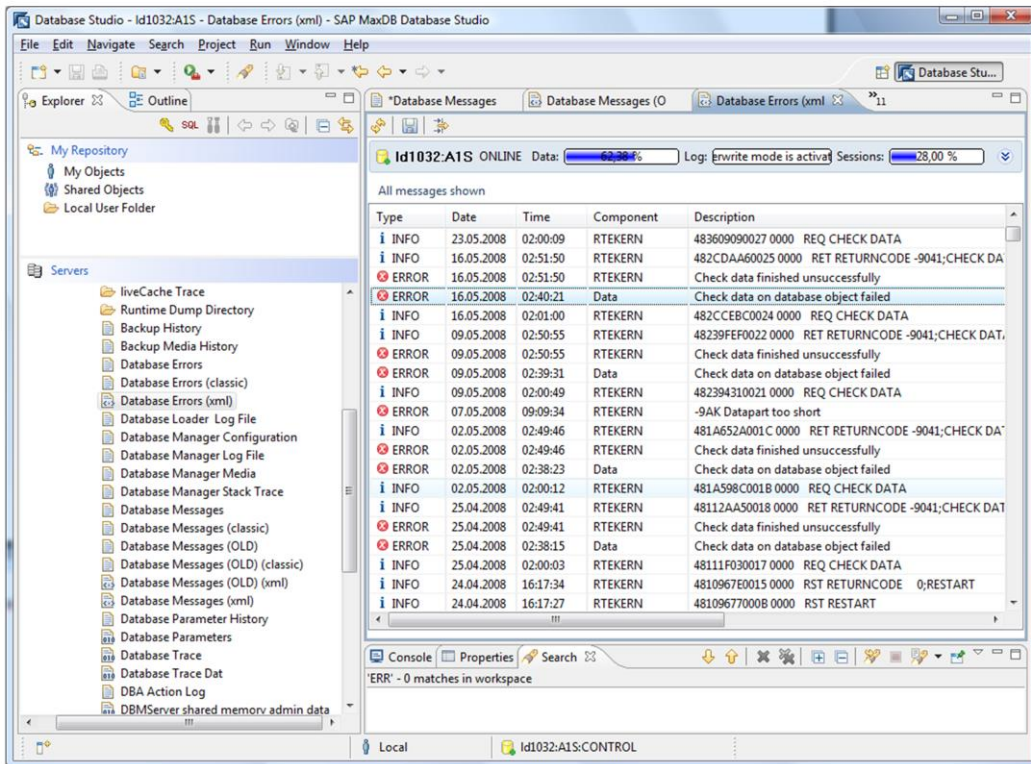
The files **KnIMsg*** are stored in the run directory of the database ((default: <indepdatapath>/wrk/<SID>). The size of **KnIMsg** can be changed by setting the parameter **KernelMessageFileSize**.

(In versions 7.5 and 7.6 location, size and name of the file can be changed with the setting for the parameters **_KERNELDIAGFILE** and **KERNELDIAGSIZE**.)

Access via DBMGUI: **Check -> Diagnosis Files -> Database Messages**

Access via DB50: **Problem Analysis -> Logs -> Kernel Administration**

Access via Database Studio: **Diagnosis Files -> Database Messages**



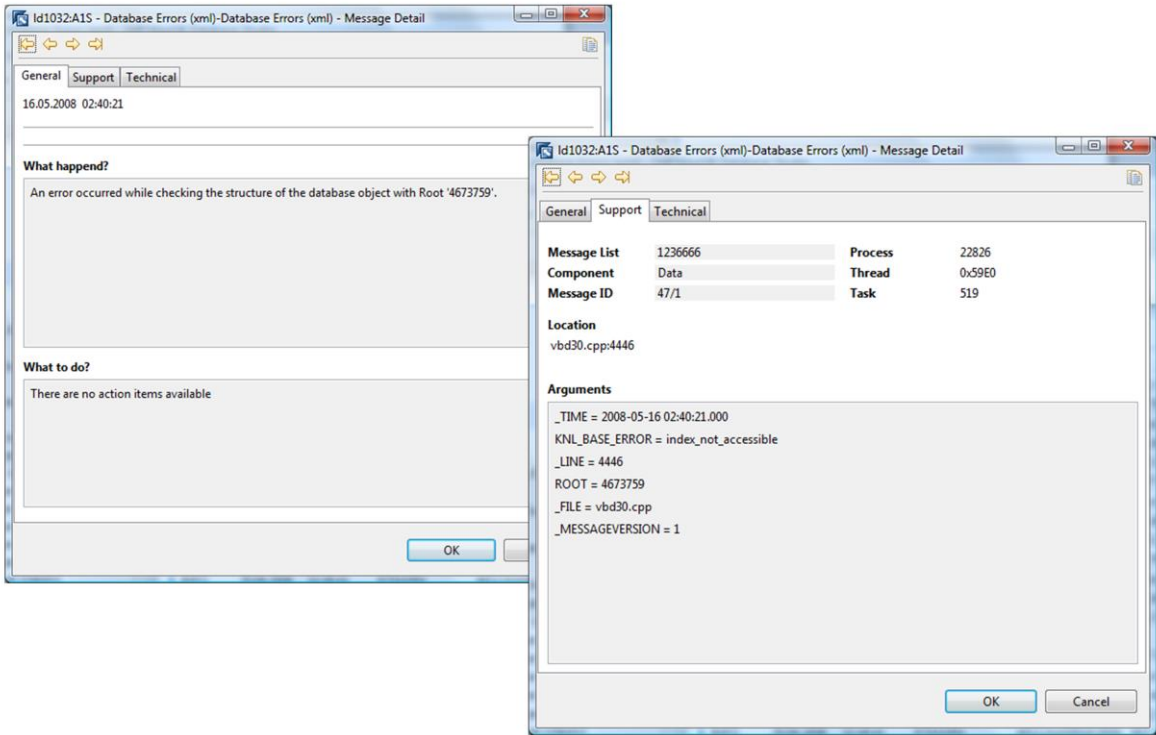
Database Studio offers to the user to either display the file **KnIMsg** in the familiar classical way or in the XML representation (see above). By double-clicking a line in the XML representation you can get more information about the error (see next slide).

Access via Database Studio:

Diagnosis Files -> Database Messages

Diagnosis Files -> Database Messages (OLD)

Diagnosis Files -> Database Errors



The following windows are displayed delivering more information about the error and proposing possibilities to correct the error. As mentioned above the windows are still partially empty and some more content is required.

```
Telnet id1032
/>
</MSG>
</MSG>
<MSG_NO="2356833" PROCESS="24250" THREAD="0x5E
04" TASK="84" TIME="2008-09-12 02:40:55.000">
<MSG_NO="1" ID="53022" COMP="RTEKERN" _TEXT="
Check Table: 7177929 (Root)">
<MSG_ARGS
  FILE="Msg_OutputKnIDiag.cpp"
  LINE="741"
  TIME="2008-09-12 02:40:55.000"
  LABEL="B-TREE"
/>
</MSG>
</MSG>
<MSG_NO="2356834" PROCESS="24250" THREAD="0x5E
04" TASK="84" TIME="2008-09-12 02:40:55.000">
<MSG_NO="1" ID="53022" COMP="RTEKERN" _TEXT="
Check Table: 7023403 (Root)">
<MSG_ARGS
  FILE="Msg_OutputKnIDiag.cpp"
  LINE="741"
  TIME="2008-09-12 02:40:55.000"
  LABEL="B-TREE"
/>
/>
```



protconv
-t KNLMMSG



```
Telnet id1032
Thread 0x5ED Task 6 2008-07-31 08:54:49 Pager 20017:
Start TaskId: 6 6 2008-07-31 08:54:49 Pager 20018:
Thread 0x5ED Task 28 2008-07-31 08:54:49 Pager 20017:
First DataCacheSegment:14
Thread 0x5ED Task 28 2008-07-31 08:54:49 Pager 20017:
Start TaskId: 28
Thread 0x5ED Task 28 2008-07-31 08:54:49 Pager 20018:
First DataCacheSegment:15
Thread 0x5ED Task - 2008-07-31 08:54:50 RunTime 3:
State changed from STARTING
to ADMIN
Thread 0x5ED Task 84 2008-09-12 02:40:10 B-TREE 53022:
Check Table: 7081664 (Root)
Thread 0x5ED Task 84 2008-09-12 02:40:10 B-TREE 53022:
Check Table: 7012718 (Root)
Thread 0x5ED Task 84 2008-09-12 02:40:10 B-TREE 53022:
Check Table: 4065226 (Root)
Thread 0x5ED Task 84 2008-09-12 02:40:10 B-TREE 53022:
Check Table: 4154968 (Root)
Thread 0x5ED Task 84 2008-09-12 02:40:10 B-TREE 53022:
Check Table: 7826164 (Root)
Thread 0x5ED Task 84 2008-09-12 02:40:10 INDEX 53022:
Thread Index: 4273150 (Root)
Thread 0x5ED Task 84 2008-09-12 02:40:10 B-TREE 53022:
Check Table: 4199377 (Root)
833.28 3%
```

Examples:

```
protconv -d A1S -t KNLMMSGARCHIVE -o KnIMsgArchive.prt
protconv -p /sapdb/data/wrk/A1S -t KNLMMSGOLD
```

The tool **protconv** (with the options shown below) is provided to allow access to the **KnIMsg** on operating system level in a readable form. If no output file is specified the text is shown directly on the screen. You can use KNLMMSG, KNLMMSGOLD and KNLMMSGARCHIVE als filekey (tag).

protconv -help

```
Usage: [-h [[<Option> | LONG]]] [-? [[<Option> | LONG]]] [-d <DBName>]
  [-o <OutputFile>] [-f <OutputFormat>] [-t <FileKey>]
  [-p <InputFilePath>] [-s <SortMode>] [-v] [-b <TimeStamp>]
  [-e <TimeStamp>]
```

- | | |
|--------------------|---|
| -h <...> | - help |
| -? <...> | - help |
| -d <DBName> | - name of the database |
| -o <OutputFile> | - output file |
| -f <OutputFormat> | - output format (classic plain(default) xml) |
| -t <FileKey> | - file key |
| -p <InputFilePath> | - path to read input files from |
| -s <SortMode> | - sort mode (!ignored!) |
| -v | - display message description from supplied files |
| -b <TimeStamp> | - output messages that were written after this time stamp only |
| -e <TimeStamp> | - output messages that were written before this time stamp only |

```

Check - Diagnosis Files - Database Messages
----- Symbolic Stack Back Trace -----
2004-04-28 17:26:48 19849 ERR 11599 BTRACE ---> 0: 0x08491782 eo670_CTraceStack +0x0012
2004-04-28 17:26:49 19849 ERR 11599 BTRACE /raid/sapdb/E30/db/pgm/kernel
2004-04-28 17:26:49 19849 ERR 11599 BTRACE Frameinfo [0x40fed260] (0x0,0x0,0x5000000c,0x1)
2004-04-28 17:26:49 19849 ERR 11599 BTRACE 1: 0x08495709 abort +0x0019
2004-04-28 17:26:49 19849 ERR 11599 BTRACE /raid/sapdb/E30/db/pgm/kernel
2004-04-28 17:26:49 19849 ERR 11599 BTRACE Frameinfo [0x40fed280] (0x402015d0,0x0,0x875a700,0x4002dd51)
2004-04-28 17:26:49 19849 ERR 11599 BTRACE 2: 0x085d5977 RTE_Crash_FRC20SAPDBErr_MessageList +0x0017
2004-04-28 17:26:49 19849 ERR 11599 BTRACE /raid/sapdb/E30/db/pgm/kernel
2004-04-28 17:26:49 19849 ERR 11599 BTRACE Frameinfo [0x40fed280] (0x402015d0,0x0,0x875a700,0x4002dd51)
2004-04-28 17:26:49 19849 ERR 11599 BTRACE 3: 0x084e0ca2 WriteVector_12IOMan_Volume1R18IOMan_IKernelPagesRC20IOMan_Cluster#
2004-04-28 17:26:49 19849 ERR 11599 BTRACE /raid/sapdb/E30/db/pgm/kernel
2004-04-28 17:26:49 19849 ERR 11599 BTRACE Frameinfo [0x40fed2e0] (0x40fed320,0x0,0x0,0x86c4382)
2004-04-28 17:26:49 19849 ERR 11599 BTRACE 4: 0x086174b3 WritePages_21IOMan_SingleLogDevice1R18IOMan_IKernelPagesRC20IOMan_
2004-04-28 17:26:49 19849 ERR 11599 BTRACE /raid/sapdb/E30/db/pgm/kernel
2004-04-28 17:26:49 19849 ERR 11599 BTRACE Frameinfo [0x40fed550] (0x42a3119c,0x2,0x42afc938,0x40fed688)
2004-04-28 17:26:49 19849 ERR 11599 BTRACE 5: 0x084d58b9 WriteLogPages_13IOMan_Manager1Rt11IOMan_Pages128Log_PageRC18Log_C1
2004-04-28 17:26:49 19849 ERR 11599 BTRACE /raid/sapdb/E30/db/pgm/kernel
2004-04-28 17:26:49 19849 ERR 11599 BTRACE Frameinfo [0x40fed590] (0x42a31198,0x2,0x42afc938,0x40fed688)
2004-04-28 17:26:49 19849 ERR 11599 BTRACE 6: 0x08588b69 FlushPages_10Log_WriterRt11IOMan_Pages128Log_Page +0x003d
2004-04-28 17:26:49 19849 ERR 11599 BTRACE /raid/sapdb/E30/db/pgm/kernel
2004-04-28 17:26:49 19849 ERR 11599 BTRACE Frameinfo [0x40fed694] (0x42a30ed0,0x2,0x42afc938,0x40fed6cc)
2004-04-28 17:26:49 19849 ERR 11599 BTRACE 7: 0x08587e75 PrepareAndFlushPageVector_10Log_WriterR21Log_RawDeviceIterator +C
2004-04-28 17:26:49 19849 ERR 11599 BTRACE /raid/sapdb/E30/db/pgm/kernel
2004-04-28 17:26:49 19849 ERR 11599 BTRACE Frameinfo [0x40fed6d4] (0x42afc938,0x5000000c,0x84a2c29)
2004-04-28 17:26:49 19849 ERR 11599 BTRACE 8: 0x08587806 Run_10Log_Writer +0x05d6
2004-04-28 17:26:49 19849 ERR 11599 BTRACE /raid/sapdb/E30/db/pgm/kernel
2004-04-28 17:26:49 19849 ERR 11599 BTRACE Frameinfo [0x40fed7a4] (0x42afc91c,0x2101,0x40fed830,0x30)
2004-04-28 17:26:49 19849 ERR 11599 BTRACE 9: 0x08327572 kb560LogWriter +0x002a
2004-04-28 17:26:49 19849 ERR 11599 BTRACE /raid/sapdb/E30/db/pgm/kernel
2004-04-28 17:26:49 19849 ERR 11599 BTRACE Frameinfo [0x40fed844] (0x42afc91c,0x2,0x40fed894,0x85sec594)
2004-04-28 17:26:49 19849 ERR 11599 BTRACE 10: 0x08104f47 ak91run_non_user_process +0x004f
2004-04-28 17:26:49 19849 ERR 11599 BTRACE /raid/sapdb/E30/db/pgm/kernel
2004-04-28 17:26:49 19849 ERR 11599 BTRACE Frameinfo [0x40fed864] (0x2,0x401dae98,0x40fed8f4,0x8104f3b)
2004-04-28 17:26:49 19849 ERR 11599 BTRACE 11: 0x0810558d a91mainprogram_with_allocator +0x0041
2004-04-28 17:26:49 19849 ERR 11599 BTRACE /raid/sapdb/E30/db/pgm/kernel
2004-04-28 17:26:49 19849 ERR 11599 BTRACE Frameinfo [0x40fed8f4] (0x40feda34,0x5,0x401db31c,0x401dae98)
2004-04-28 17:26:49 19849 ERR 11599 BTRACE 12: 0x084732d0 aa941CreateAllocatorAndCallMainprog +0x01c0
  
```

When the database crashes, support often needs to know at which point in the source code the database was when the crash occurred.

On Unix/Linux, this information is usually generated from a core dump with a debugger. On Windows, this information is found in the file drwtsn32.log, but only if Dr Watson is registered as the system debugger.

Core dumps can be very large. Writing a core dump delays the crash of the process.

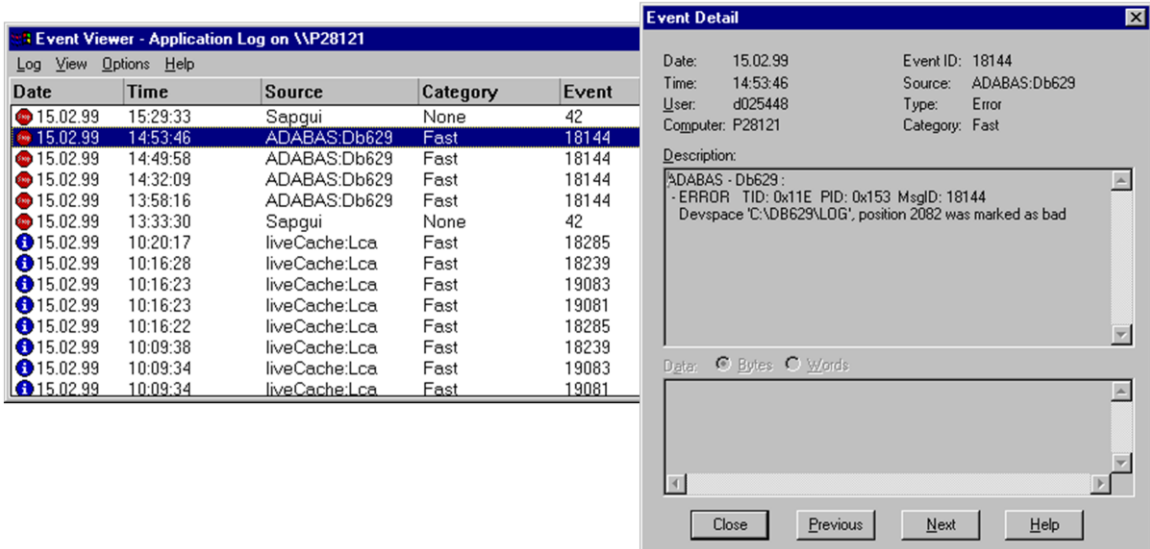
For that reason, when a crash occurs the MaxDB kernel automatically writes the backtrace stack and values of the CPU register to the **KnIMsg (knldiag)** file.

If the problem is due to an error in the database software, the cause can usually be found using this information.

In the present example we see a simulation of an I/O error during writing to a log volume. It is not a software error.

Under WINDOWS important messages are additionally written to the event log.

Example:



The screenshot shows the Windows Event Viewer interface. The main window displays a list of events from the 'Application Log' for computer '\P28121'. The selected event (ID 18144) is highlighted in blue. To the right, the 'Event Detail' window is open, showing the following information:

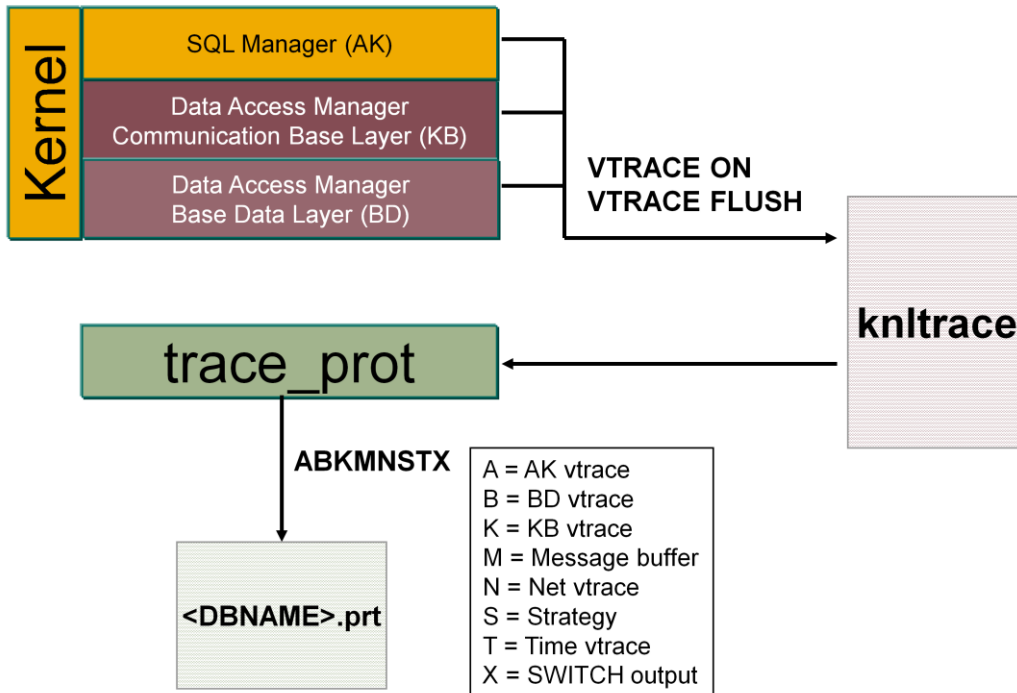
- Date: 15.02.99
- Time: 14:53:46
- User: d025448
- Computer: P28121
- Event ID: 18144
- Source: ADABAS:Db629
- Type: Error
- Category: Fast

The description of the event is as follows:

```
ADABAS - Db629 :
-ERROR TID: 0x11E PID: 0x153 MsgID: 18144
Devspace 'C:\DB629\LOG', position 2082 was marked as bad
```

At the bottom of the Event Detail window, there are radio buttons for 'Data', 'Bytes', and 'Words', with 'Data' currently selected. Below these are 'Close', 'Previous', 'Next', and 'Help' buttons.

If the file **KnIMsg** has already been overwritten, you may still find useful information here.



The kernel trace, or Vtrace, is used for analyzing executed SQL statements.

When you activate Vtrace, you specify which areas of the kernel the file **knltrace** is written from. Generally a default setting is taken.

The kernel trace is not active by default. The default trace has a minimal effect on system performance. Each UKT writes to its own main memory buffer, precluding collisions during trace writing. If you select other options, however, writing the trace can be resource intensive and should be done only where needed for problem analysis.

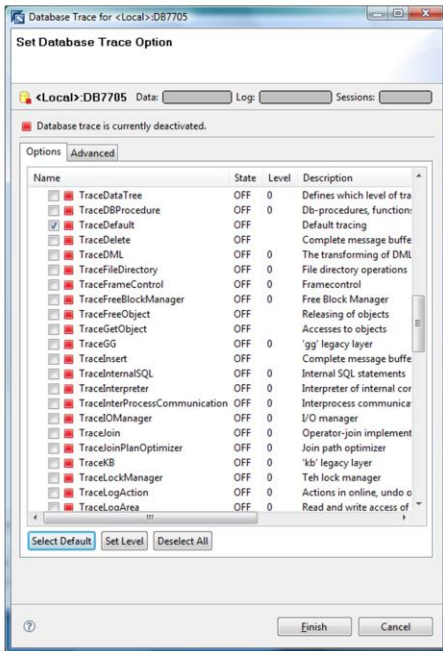
For the trace output, you specify the levels or modules of the kernel for which the logs are to be extracted.

Data concerning strategies and times is only output if the options OPTIMIZER or TIME, respectively, are active for the Vtrace.

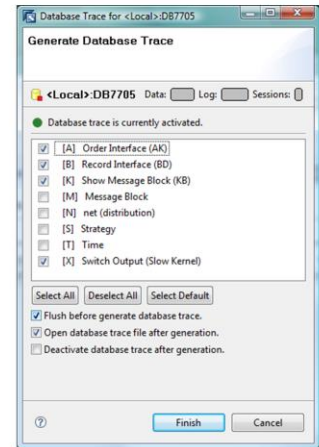
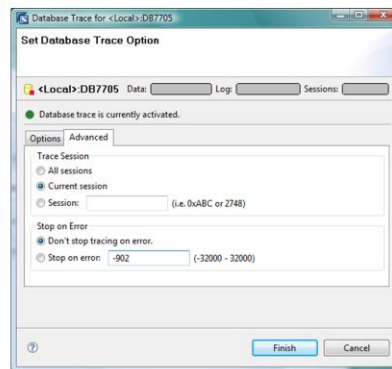
The SWITCH output contains data from the trace of a so-called slow kernel. A slow kernel is a special MaxDB debugging kernel. It is only used upon the special request of development or support.

The Vtrace can be activated for a single user session (FOR SESSION).

To prevent cyclical overwriting, the writing of the trace can be switched off automatically when a specified error code occurs (STOP ON ERROR).



1. Switch on the VTRACE (if required for a specified session)
2. Execute database action
3. VTRACE FLUSH
4. Switch off the VTRACE
5. Convert the file to ASCII format



Switching the Vtrace on or off as well as flushing it can be done with the dbmcli, the Database Studio or DBMGUI and with transaction db50. Flushing the Vtrace can also be done with the SQLSTUDIO.

Required dbmcli commands:

Activate:

```
dbmcli -d <SID> -u <dbm-user>,<password> trace_on default
```

Flush:

```
dbmcli -d <SID> -u <dbm-user>,<password> trace_flush
```

Deactivate:

```
dbmcli -d <SID> -u <dbm-user>,<password> trace_off
```

Evaluate:

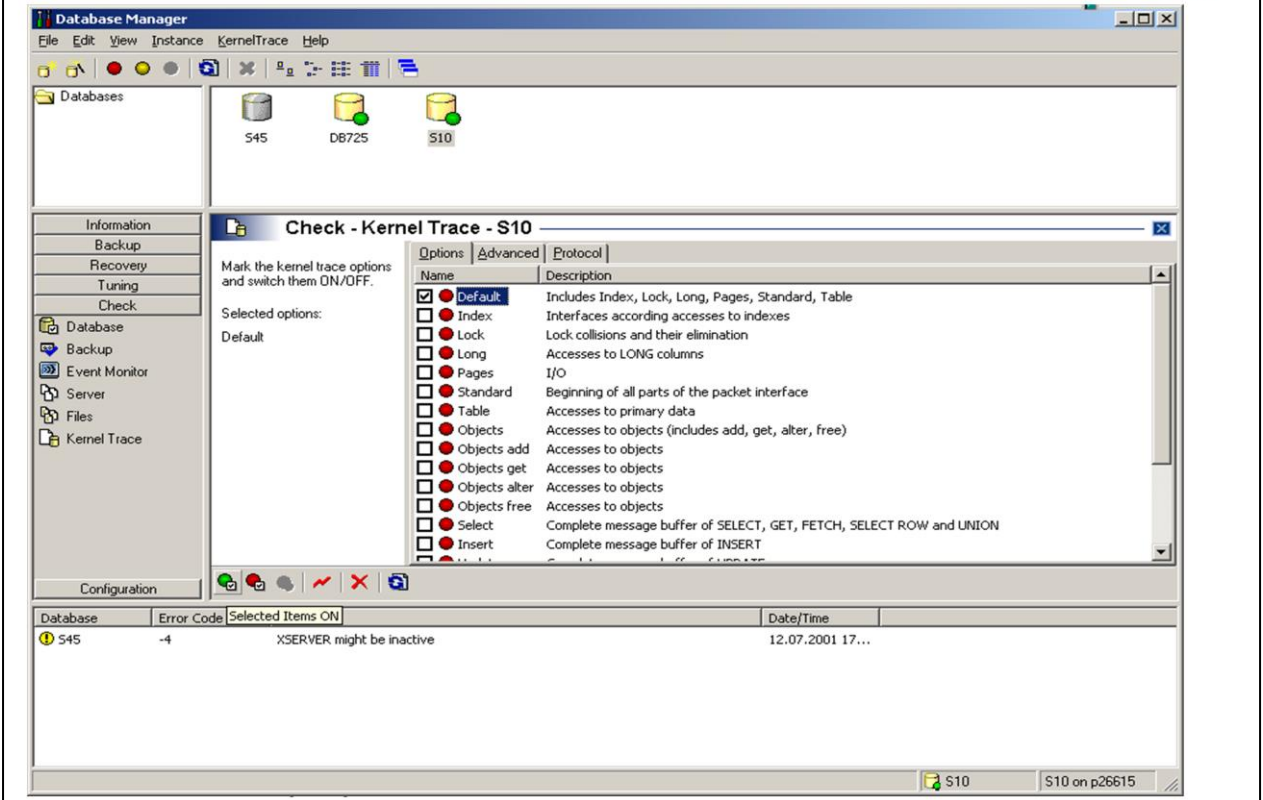
```
dbmcli -d <SID> -u <dbm-user>,<password> trace_prot <options>
```

In the context menu of the installed databases the trace can be switched on and generated as a readable file with.

Database Trace -> Options

Database Trace -> Generate

DBMGUI Kernel Trace (1)



You can administer the database kernel trace with the DBMGUI.

Unless otherwise specified by development or support, the default Vtrace is sufficient.

You can also activate information about DELETE, INSERT, UPDATE, SELECT and Optimizer operations.

The Vtrace can be activated and deactivated, flushed, initialized and displayed using the buttons. During initialization, all information in the trace buffer is deleted.

Kernel Trace (2)

SAP

The screenshot shows the SAP Database Manager interface. The main window is titled "Check - Kernel Trace - S10". It features a left-hand navigation pane with options like Information, Backup, Recovery, Tuning, Check, Database, Backup, Event Monitor, Server, Files, and Kernel Trace. The main area contains a configuration window for "Check - Kernel Trace - S10" with tabs for "Options", "Advanced", and "Protocol". The "Options" tab is active, showing a checkbox for "Trace session" set to "All Sessions" and a checkbox for "Stop on error" set to "(-32000 <= Error <= 32000)". Below the configuration window is a table with columns for Database, Error Code, Description, and Date/Time.

Database	Error Code	Description	Date/Time
S45	-4	XSERVER might be inactive	12.07.2001 17...

TRACE SESSION

The Vtrace can be activated for particular database sessions. To do so, however, the database session must be known.

The outputs of

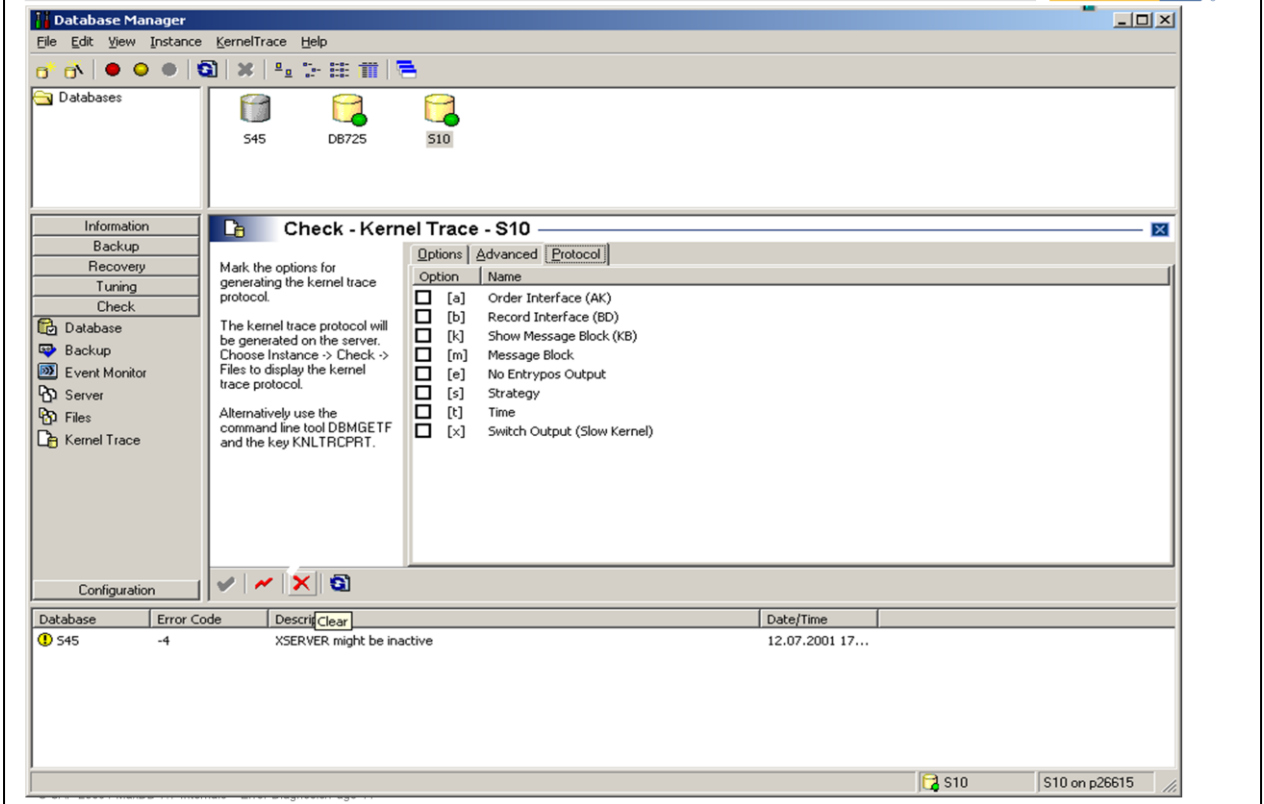
**x_cons <SID> show active and
SELECT * FROM TRANSACTIONS**

are helpful in this regard.

STOP ON ERROR

You can set the Vtrace so that it is automatically switched off when a certain error occurs. This is useful when you want to reproduce a particular problem and know which error will occur. This function prevents relevant information from being overwritten.

Kernel Trace (3)



On the **'Protocol'** tab, you can sort the information from the **knltrace** file and extract desired areas to an ASCII file.

You specify the layers or modules of the kernel for which you want to extract the trace outputs; DEFAULT: abkmx.

Data concerning strategies and times is only output if the options OPTIMIZER or TIME, respectively, are active for the Vtrace.

The SWITCH output contains data from the trace of a so-called slow kernel. A slow kernel is a special MaxDB debugging kernel. It is only used upon the special request of development or support.

Kernel Trace (4)



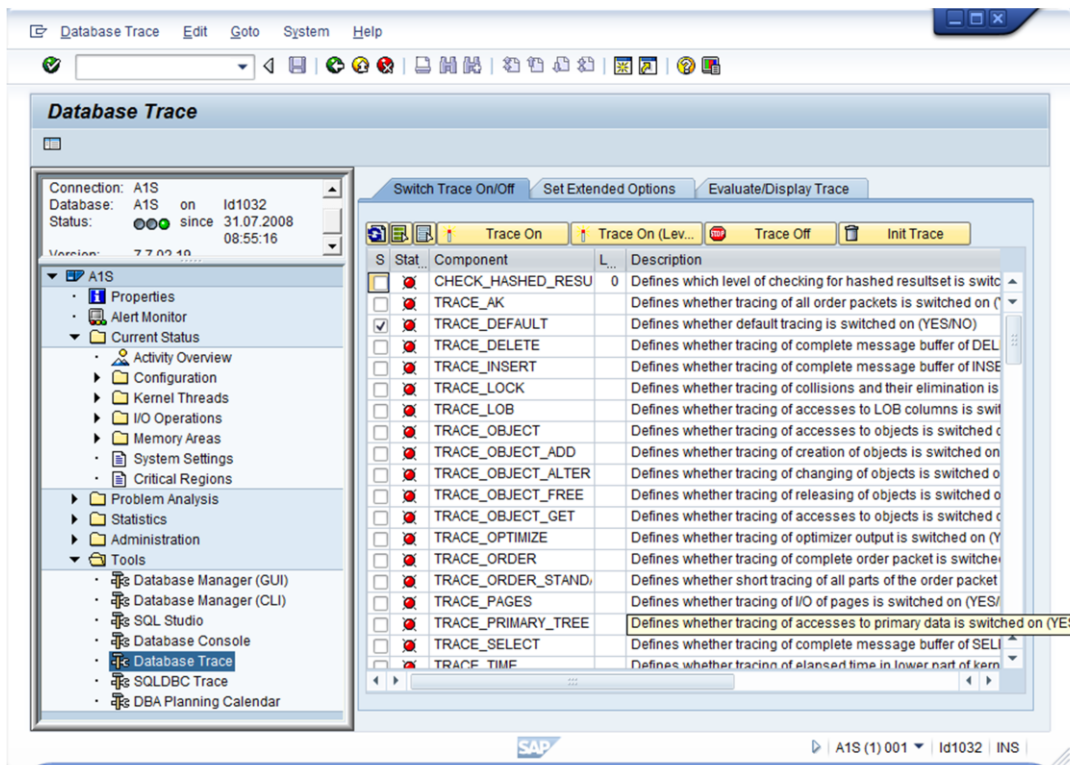
```
*** KERNPROT 7.2.5                2001-07-13 09:34:35 ***
-----
Input File: D:\Instroot_S10\wrk\S10\knitrace.dat
----- from entry 1.6 to 50.8197 (from page 1 to 50) -----
===== T5 ===== id14745728 =====1.6 page 1
RECEIVE: ascii, full_swap, 70205-DBM (1 segment, len: 40)
(1.6 page 1)
ok / RETURN SEGMENT 1 (0 parts, len: 40)
sqlstate: '00000'

REQUEST: ascii, full_swap, 70205-DBM (1 segment, len: 80)
(1.102 page 1)
dbs SEGMENT 1 (1 part, len: 80)
session_sqlmode, user_cmd
command PART (1 argument, size: 32232)
-----
```

Database	Error Code	Description	Date/Time
S45	-4	XSERVER might be inactive	12.07.2001 17...

You can display the contents of the Vtrace via the menu path ,**Check -> Files -> Kernel Trace Protocol**

Even if you can find the evaluated error using the search function, it is all but impossible for a customer to form an independent interpretation of this trace. Errors can be found here only with knowledge of the source code. Thus the trace file should be provided to development.



Administering the kernel trace (vtrace) can also be done with transaction db50.

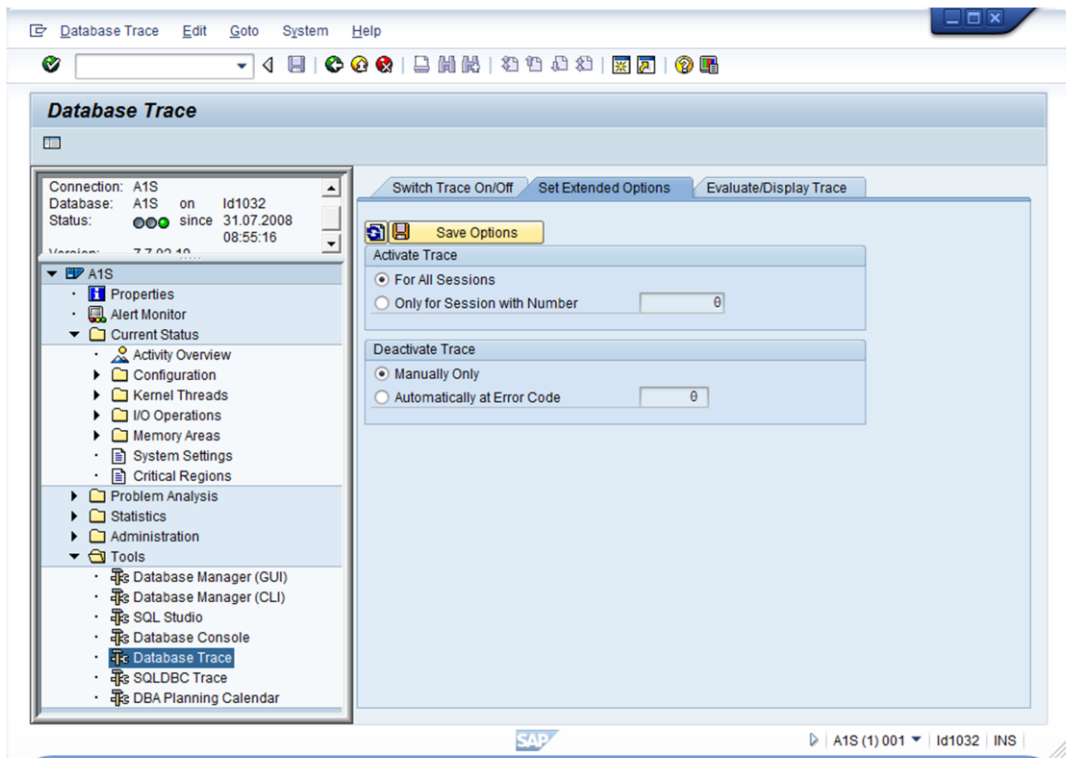
Initialize Trace: If you want to be sure that only subsequent database actions are logged, choose **'Init Trace'**.

Activate Trace: To activate the trace, first choose your trace options, (usually default options) and then **'Trace On'**.

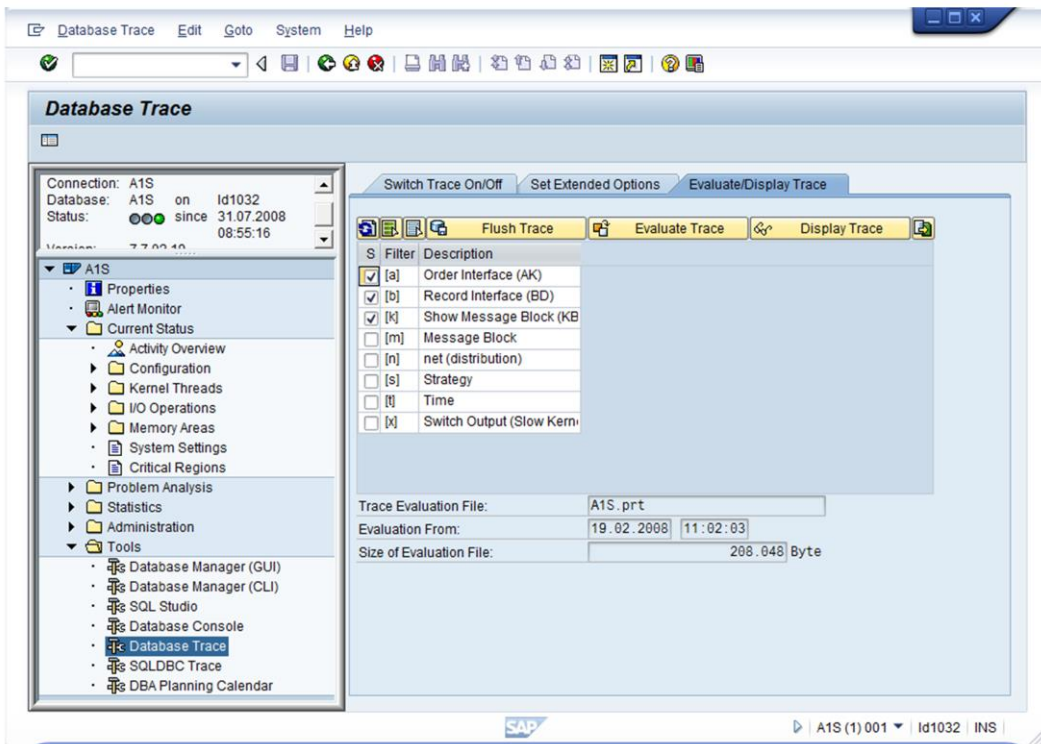
You can activate more trace options while the trace is running by selecting them and choosing **'Trace On'** again.

Then the program that received the short dump, for example, is restarted.

The "Status" column shows whether the trace is currently activated, and with which options. The activated options are displayed in green.



On the **'Set Extended Options'** tab, you can determine whether the kernel trace should be written only for a selected session and whether it should be stopped automatically in case of a selected error code in order to prevent overwriting.

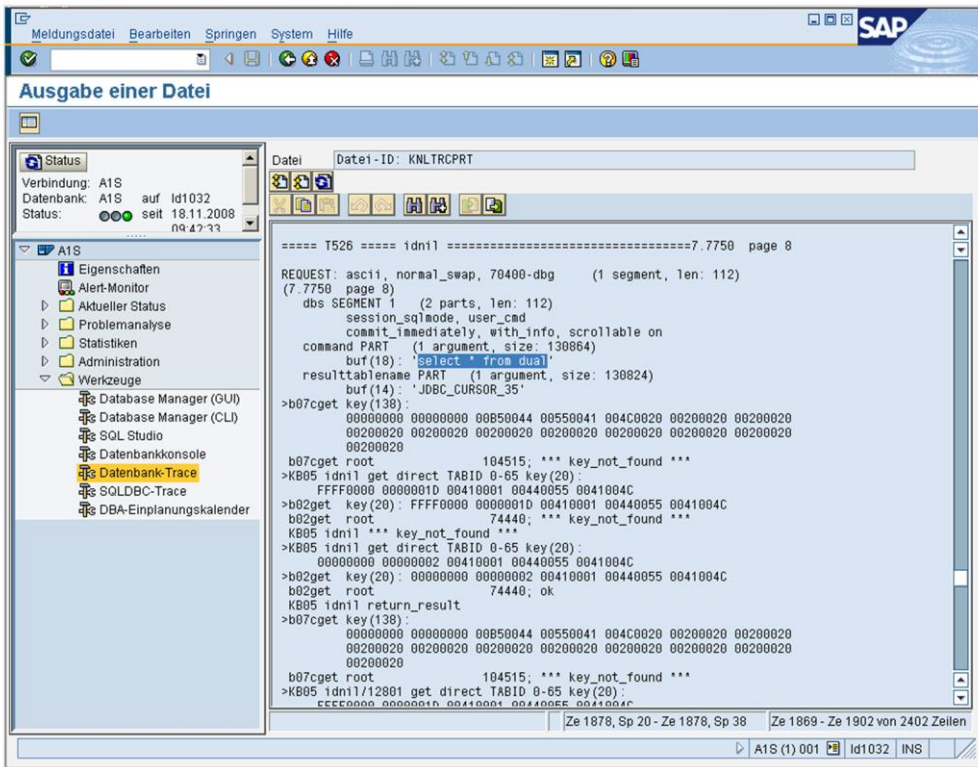


© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 48

When the program you want to check has been terminated, the Vtrace has to be flushed so that the information in the buffer is written to the disk.

Flush Trace Buffer : To analyze the trace, choose **‘Flush Trace’**.

Format Trace : To format the trace to a legible form, first select the desired layers and then **‘Evaluate Trace’**



Display Trace : Immediate display can be effectuated with **'Display Trace'**.

As the resulting file <SID>.prt can attain a considerable size, you can use the right-hand button to save to a local file.

The extracted trace is then read and analyzed by support and development. Knowledge about the source code is required for further interpretation.

Contains the global memory, e.g.:

- Lock lists, data cache, catalog cache, ...
- Administration structures of these caches

The file is created in the following ways:

- DIAGNOSE: by a user with DBA rights
- dbmcli: db_stop -dump
- when the database crashes

The file 'knldump' might become very huge. It contains binary data which can be transferred to a readable form with DIAGNOSE.

In case of crash or hanger situations due to manual interventions, the database generates a dump that contains the information from the global memory.

UNIX: No dump is written if the database crashes due to a UNIX signal.

The file **knldump** is stored in the run directory of the database ((default: <indepdatapath>/wrk/<SID>).

If there's not enough space in the filesystem here, for example, you can change the location and name of the file with the parameter KernelDumpFileName (_KERNELDUMPFIL).

As this is a binary file, displaying it with the Database Studio, DBMGUI or transaction db50 is not useful.

As default the parameter SUPPRESS_CORE is set to YES. This has the effect that MaxDB does not write a knldump in case of a crash; the output of a core file is also suppressed.



Status of the runtime environment in case of crash

- `x_cons <SID> show all`
- helps to identify previously active tasks
- detailed information about the individual tasks
- region statuses, suspend reasons, counter statistics

Additional analysis for KnIMsg output

No formatting necessary (legible file)

Storage in run directory

If a crash occurs, the status of the runtime environment is recorded in an **rtdump**.

The data corresponds to that in the output of `x_cons <SID> show all`

The file can be viewed directly in a system editor; no further formatting with a tool is required.

The file **rtdump** is stored in the run directory of the database ((default: <indepdatapath>/wrk/<SID>). Location and name of the file can be changed with the setting for the parameter `RTEDumpFileName` (`_RTEDUMPFIL`).

Access via Database Studio: **Diagnosis Files -> Runtime Environment Dump**

Access via DBMGUI: **Check -> Diagnosis Files -> Runtime Environment Dump (RTEDUMP)**

Access via DB50: **Properties -> Files -> RTEDUMP**

Example:

- Identification of the active task (status "Running")
- Detailed information about this task
- Identification of the affected application server
- If necessary, additional information in the system log as well as in the dev log tell you more about the triggering command.

```

rtdump2.txt - Notepad
File Edit Format View Help

[...]
T416 16 0xF28 User      7924* Running ←
[...]

Task Type      Stack used  Stack free
T416 User      92.4      931.6

[...]

      T416  ) user      ( pid = 7924      )
remote_node  : DDATH101      remote_pid   : 7924
dispatcher_cnt: 69788780    command_cnt  : 11549217
exclusive_cnt: 4049941230  self_susp_cnt: 141816
Resume count 0 total 425189 History [ T547 T548 T548 ]
self_read_io : 0           dev_read_io  : 6522
state_vwait  : 1291      state_vsleep : 337791  state_vsusp  : 425189
prio_total_cnt: 10443    from_oth_ukt: 10090

[...]

==> WP auf DDATH101 / pid 7924

N:\usr\sap\IP2\D00\work\dev_w16.o1d ←

DIA 16 660 JX98U6
      ZI2I BY4 Datenbankfehler 800 beim FET-Zugriff auf Tabelle
EQUI_ADDR aufgetreten

C Fri Feb 03 10:36:41 2006

C *** ERROR => SQL FETCH C_091 on connection DB_000, rc=800 (Implicit
SERVERDB restart (connection aborted))
[dbsladd.c 4831]

C SELECT * FROM "EQUI_ADDR" WHERE "CLIENT" = ? AND (
"POST_CODE1" LIKE ? OR "POST_CODE1" LIKE ? OR "POST_CODE1" LIKE ? )
AND ROWNUM <= ?
    
```

In addition to the information from **KnIMsg (knldiag)**, the output of **rtdump** can be of use in analyzing crashes. This can be the case for a variety of reasons as this dump contains a plethora of information from the runtime environment. But these special cases will not be discussed further here.

An example is shown on the slide. **rtdump** can help identify the command that caused a crash by determining the tasks that were active at the time. They are in the `x_cons <SID> show task` part of the output and marked "Running". In the detailed information for each individual task you'll find the application server under "remote_node". In the system log or the dev logs of this application server, commands are logged that led to some problem. Even if it cannot be guaranteed that the identified command was solely responsible for the crash, it is still worthwhile to try to reproduce the crash and (for instance with activated traces) determine the cause of the error.

Dump of corrupt pages

- Checksum error: *.bad
- Problem with page content identified: *.cor

Storage in run directory

Formatting done with x_diagnose

If corrupt pages are identified, they are written to the file system so they can be subjected to further analysis.

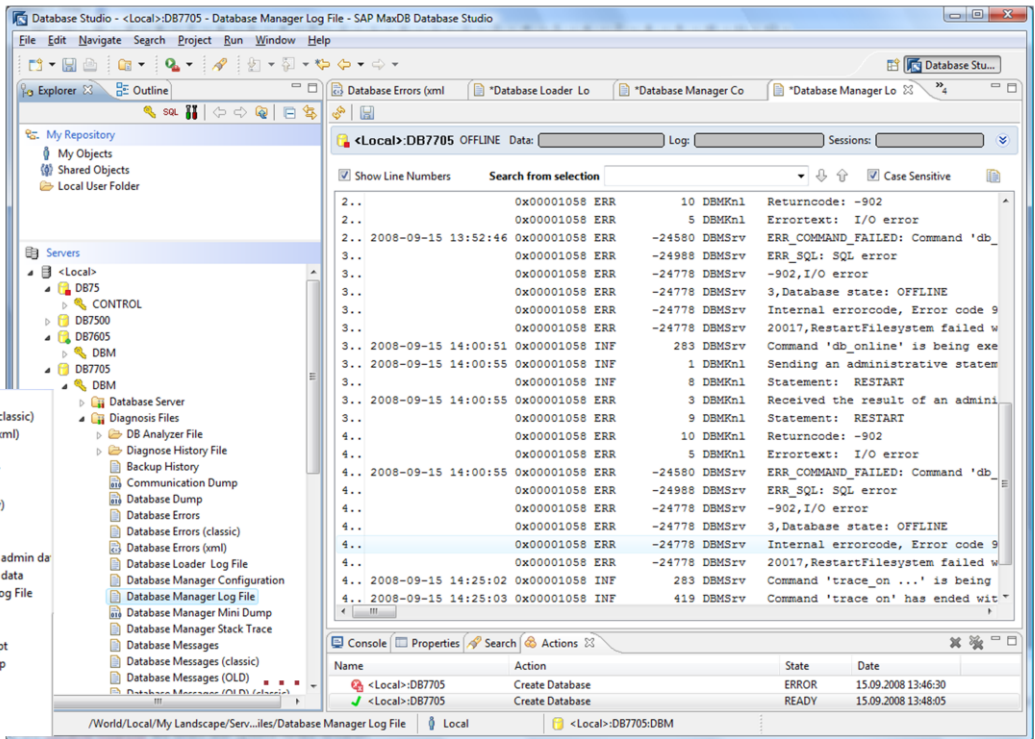
A corrupt page is dumped as a *.bad file if the I/O check found an error while importing a page (checksum error).

A *.cor file is generated if a content problem is identified with the available context knowledge while working with a page in the cache.

The files are generated in the run directory of the database ((default: <indepdatapath>/wrk/<SID>).

As these are binary files, display with Database Studio, DBMGUI or transaction db50 is not useful. Evaluation is done with the tool x_diagnose.

Database Studio – Access to Diagnosis Files



With Database Studio you can access via context menu for the selected database to the presentable diagnosis files. Binary files like Database Dump (knldump) or Database Trace (Raw/Binary) cannot be displayed without former evaluation. The Kernel Messages (KnIMsg) files stored in XML format are directly shown in a readable form.

DB50 – Access to Diagnosis Files



The screenshot shows the SAP DB50 Properties dialog box. The 'Files' tab is selected, displaying a 'File Overview' table. The table lists various diagnosis files with columns for File ID, File Name, Size, Date, Time, Description, and FileType. The file 'KNLMSG' is highlighted in yellow.

File ID	File Name	Size	Date	Time	Description	FileType
KNLMSG	KnMsg	8.447.051	16.09.2008	10:47:38	Database Messages	ASCII
KNLMSGARC	KnMsgArchive	1.470.464	12.09.2008	02:59:24	Database Errors	ASCII
KNLMSGOLD	KnMsg.old	8.450.269	26.07.2008	05:06:50	Database Messages (OLD)	ASCII
KNLTRC	knltrc	28.180.480	31.07.2008	08:57:39	Database Trace	BINARY
BACKHIST	dbm.knl	4.254	17.03.2008	15:53:53	Backup History	ASCII
BACKMDF	dbm.mdf	2.328	04.03.2008	11:04:37	Backup Media History	ASCII
DBMPRT	dbm.prt	667.019	15.09.2008	17:32:48	Database Manager Log File	ASCII
DBMMDF	dbm.mmm	179	04.02.2008	13:21:59	Database Manager Media	ASCII
DBMPAHI	A1S.pah	122.880	15.09.2008	12:36:56	Database Parameter History	ASCII
LCINIT	lcinit.log	16.581	24.01.2008	09:59:41	LiveCache Initialisation	ASCII
LCINITCMD	lcinit	21.568	19.12.2007	20:15:37	LiveCache Initialisation Script	ASCII
LCINITHIS	lcinit.his	16.581	24.01.2008	09:59:41	LiveCache Initialisation History	ASCII
INSTPRT	dbm.ins	774.717	03.03.2008	17:07:05	Installation Log File	ASCII
KNLTRCPRT	A1S.prt	208.048	19.02.2008	11:02:03	Kernel Trace Log File	ASCII
DBAHIST	dbahist.prt	2.958	12.09.2008	02:58:51	DBA Action Log	ASCII
DIAGDIR	File	4.096	31.07.2008	08:57:22	Diagnose History	DIRECTOR
ANALYZER	analyzer	4.096	16.09.2008	00:07:34	DB Analyzer File	DIRECTOR
LCTRC#init.log	lcinit.log	16.581	24.01.2008	09:59:41	LiveCache Trace (ASCII)	ASCII
LCTRC#init.his	lcinit.his	16.581	24.01.2008	09:59:41	LiveCache Trace (ASCII)	ASCII

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 55

With SAP transaction db50, error diagnosis can be performed for a running (online) database using the SAPGUI. Which tool you use is a matter of personal preference; however, this redundancy is often useful, for instance if only certain activity types or not all passwords for the various access types are available to you.

db50, then, also allows simple access to all diagnosis files of the database via the menu option *Properties* and the *Files* tab. Here you see an unarranged list; the actual contents of the most important diagnosis files are still located on the various menu paths.

Using transaction db59, you can administer multiple MaxDB and liveCache instances from a SAP WebAS.

DBACockpit – DBA Planning Calendar



The screenshot displays the SAP DBACockpit interface for the 'DBA Planning Calendar'. The main window is titled 'Jobs: DBA Planning Calendar' and shows a calendar for 'September 2008, Calendar Week 37'. The calendar grid is organized by day (Monday, 8 to Friday, 12) and time slots (00:00 to 20:00). An 'Action Pad' at the bottom lists various database maintenance tasks, including 'Complete data backup', 'Incremental data backup', 'Activate automatic log backup', 'Deactivate automatic log backup', 'Log backup', 'Mark tables requiring statistics update', 'Update statistics for marked tables', 'Update all optimizer statistics', and 'Check database structure'. The interface also includes a navigation tree on the left and a top menu bar with options like 'Administration', 'Edit', 'Goto', 'System', and 'Help'.

As of WebAS version 7.0 the transaction DBACockpit can be used as a central tool for database administration. In addition to several administrative tasks that are also provided by transaction db50, in the cockpit the planning calendar is maintained. Backup activities, update statistics and consistency checks can be scheduled here.

```

Command Prompt - telnet p34777
p34777:e30adm 87> dbmcli -help
usage: dbmcli [<options>] [[-c] <DBMServer-Command>]
  <options>:
    -u username,passwd (DBM user)
    -U [userkey] (default DBMUSR)
    -uUTL [username,passwd] (UTILITY session)
    -uSQL [username,passwd] (SQL session)
    -uSRV (Service session)
    -V (show version of server)
    -d dbname (set dbname)
    -R dbroot (set dbroot)
    -n node (name of servernode)
    -i inputfile (Default stdin)
    -ic inputfile (Default stdin)
    -o outputfile (Default stdout)
    -t protocolfile (writes subsequent protocol)
    -s (local mode, don't use with -n)
  <DBMServer-Command>:
    Everything after the options will be sent to DBMServer.
    For more information about the DBMServer-Commands use
    the DBMServer-Command help.

p34777:e30adm 88>
p34777:e30adm 88>
p34777:e30adm 88>

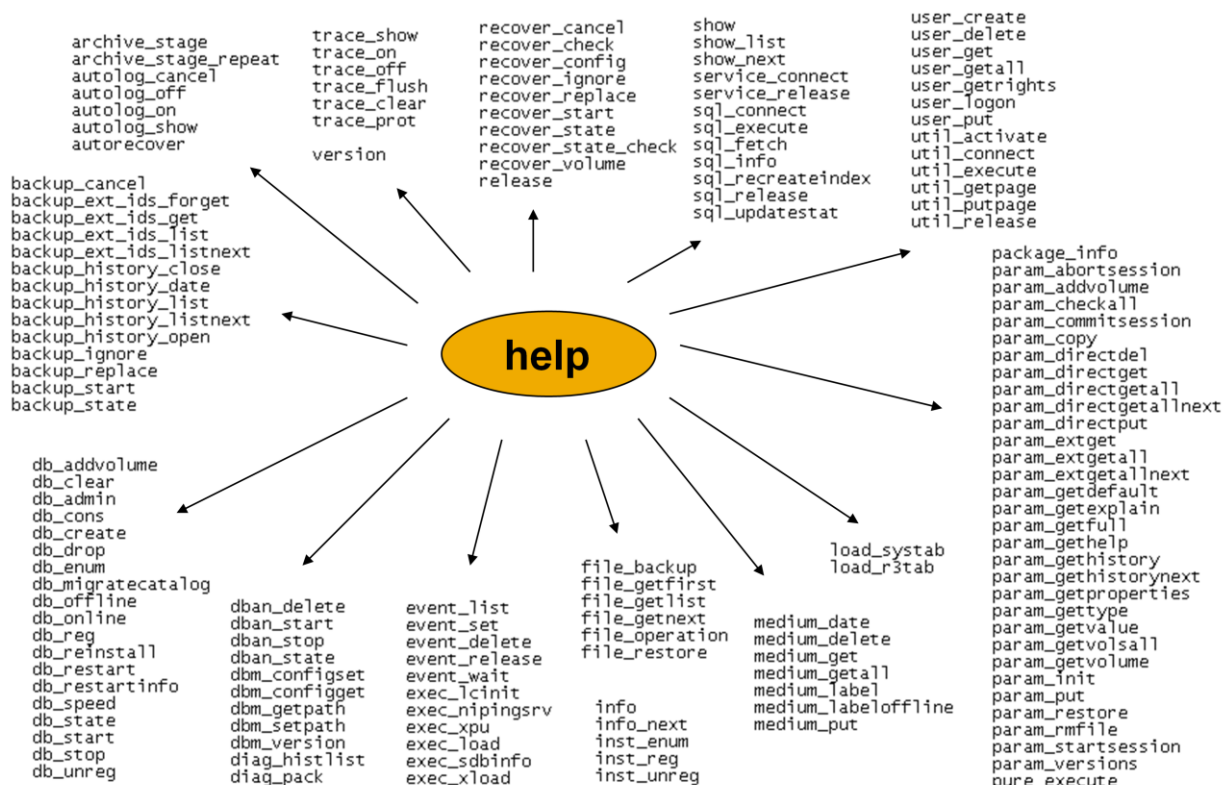
```

dbmcli is used for line-based database administration work; the name is an acronym for Database Manager Command Line Interface.

It can be useful for short ad hoc queries in a telnet session or for use in scripts. For more extensive administration tasks, the DBMGUI is preferable as it initiates the action and does not require precise knowledge of the command sequences, which can be very complex.

Commands are sent to the DBM server, which processes the requests; the commands that have been sent are logged in the file **dbm.prt**.

The **dbmcli** allows you to open a utility or an SQL session, which means that SQL queries can be sent to a database in the online operational state. The utility session is meaningless and only exists because of compatibility reasons.



The **dbmcli**, as the illustration makes clear, has an extensive range of functions. You can display the list of possible commands in a dbmcli session with **help**. The help information contains additional information about which parameters have to be entered and what type of logon is required.

Some commands cannot be used alone, but only make sense as part of a command sequence.

```
Select Command Prompt - telnet p34777
OK
State
ONLINE
p34777:e30adm 56> dbmcli -u control,control -d E30 dbm_getpath IndepDataPath
OK
/sapdb/data
p34777:e30adm 57> dbmcli -u control,control -d E30 param_directget RUNDIRECTORY
OK
RUNDIRECTORY      /sapdb/data/wrk/E30
p34777:e30adm 58> dbmcli -u control,control -d E30 version
OK
version,os,dbroot,logon,code,swap
"7.4.3","UNIX","/sapdb/E30/db",False,ASCII,2
p34777:e30adm 59> dbmcli -u control,control -d E30 param_gethelp LRU_FOR_SCAN
OK
Specification of scan performance in the data cache
p34777:e30adm 60> dbmcli -u control,control -d E30 -uSQL sape30,sap sql_execute
"select * from messages where msgno = -9026"
OK
END
-9026:'ENG': 'System error: BD Bad datapage'
p34777:e30adm 61> dbmcli -u control,control -d E30 medium_get Data_tmp
OK
Data_tmp          /tmp/E30.backup FILE      DATA      0          8          YES        NO
                20030612182306  20030612182306
p34777:e30adm 62> dbmcli -u control,control -d E30 db_speed
OK
Speed
FAST
p34777:e30adm 63> █
```

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 59

The examples show some commands that are useful for diagnosis; these are stand-alone commands that can provide an initial overview of the situation.

In command 60, in addition to logging on with the DBM operator, you must also specify a user authorized to access database objects.

```

Telnet Id1032
Id1032:alsadm 51> dbmgetf - help
usage: dbmgetf [<options>]
<options>:
  -d dbname      (name of database)
  -u user,pwd    (user for authorization)
  -U userkey     (userkey for authorization)
  -n node        (name of servernode)
  -e encryption (encryption method)
                (use only with -n)
                (valid methods: 'SSL')
  -k id          (id of database file)
  -f file        (name for local file - optional)
  -q            (quiet mode)
  -l            (list file id's)
  -p <param>    (DATE or LINE parameter)
  -op <opid>   (SHRINK or DELETE operation)

```

```

Telnet Id1032
Id1032:alsadm 71> dbmgetf -U c -d A1S -l
KNLMSG
KNLMSGARC
KNLMSGOLD
KNLTRC
BACKHIST
BACKMDF
DBMPRT
DBMMDF
DBMPAHI
LCINIT
LCINITCMD
LCINITHIS
      * * *
INSTPRT
KNLTRCPRT
DBAHIST
DIAGDIR
ANALYZER
LCTRC#init.log
LCTRC#init.his
EDCFGI
      * * *

```

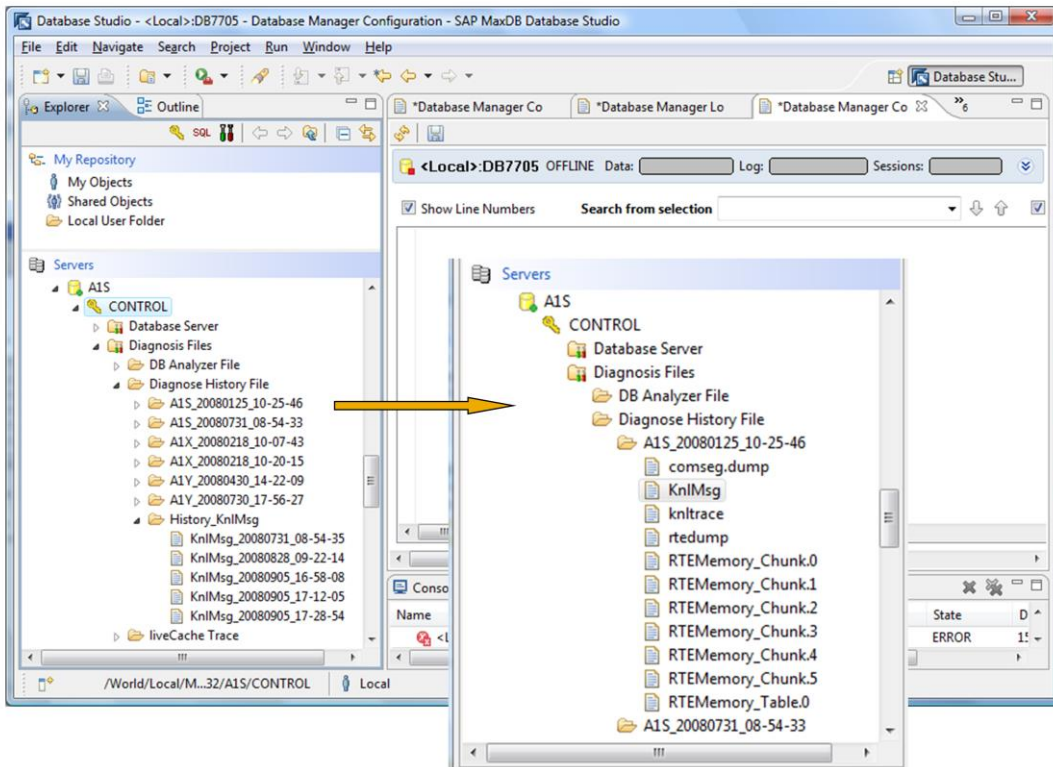
dbmgetf -d A1S -u control,control -k KNLMSG
dbmgetf -n p34777 -d E30 -u control,control -k DBMPRT

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 60

dbmgetf is a tool that enables quick access to log files, for instance in a telnet session. It is mainly used internally since, in general, the GUI-supported display options are more convenient. The KnlMsg files are automatically transformed to a readable format.

With the **-n** option, you can specify a computer on which you want to enable remote access.

The log files are not addressed by the names stored in the operating system, but rather by abbreviations, which can be displayed using the **-l** option.



© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 61

There is an automatic procedure for receiving important information about crash situations.

The following files do not have to be explicitly backed up after a crash since they are automatically copied to a backup directory:

KnIMsg (knldiag), knltrace, knldump, rtdump, *.dmp, *.buf, *.stm

If the database recognizes that it is being restarted after a crash, then the necessary files are backed up to a directory with the following naming convention:

- `<DB-NAME>_<DATUM>_<ZEIT>`, e.g.: `S10_20001114_12-09-45`

The backed up diagnosis files are deleted from the original directory.

The backup directory is under the directory **DiagnoseHistoryPath (DIAG_HISTORY_PATH)** (which must be configured) and is referred to as the history in the following.

You can also configure the number of histories with parameter **DiagnoseHistoryCount (DIAG_HISTORY_NUM)**. If you exceed this number of histories, then the oldest history is deleted when a new backup is made.

The database can still be restarted if a backup cannot be made correctly.



CHECK DATA [Options]

- Checks structural consistency of the whole database. If no errors are found, „bad flags“ in the so-called filedirectory and the root page are reset.

CHECK TABLE <OWNER>.<TABLENAME> [Options]

- Checks all pointers within the specified table tree.

Mirroring

- If data volumes are mirrored by means of the operating system or by hardware, the database cannot influence which disk is used for reading pages. CHECK TABLE may not find any errors.

Errors

- If CHECK TABLE delivers an error, hardware problems must be solved and a backup must be restored.

Check Data (previously Verify) checks the structural consistency of the entire database. It considers tables as well as indexes and LOB columns.

The semantics of the data model is not taken into account. Logical errors are not found, but only errors caused by hardware defects.

Every page contains a check number. This is calculated with each read-I/O and compared with the value stored on the page. If the values are different, there is an error.

One typical error that may be detected is BAD DATA PAGE.

Check Table checks all dependencies and links within the specified table tree. Indexes are not taken into account.



EXTENDED

- Extended check of the key sequence

WITH LONG CHECK (former WITH SHARE LOCK)

- Additional check of LONG columns
- Share lock is set

EXCEPT INDEX

- Indexes are not checked

WITH UPDATE

- Execution in DB mode ADMIN
- Additional maintenance of the converter: page numbers with no references are removed.

Another diagnosis option is calling

CHECK DATA EXTENDED.

This performs a more precise check of the key lengths and checks the sequence of the primary keys on all levels of the B* tree. Because this option is CPU-intensive, execution was not standard in older versions. As of version 7.6.01 it is standard behaviour for CHECK DATA and CHECK TABLE, because CPU load can now be neglected due to the performance of modern CPUs.

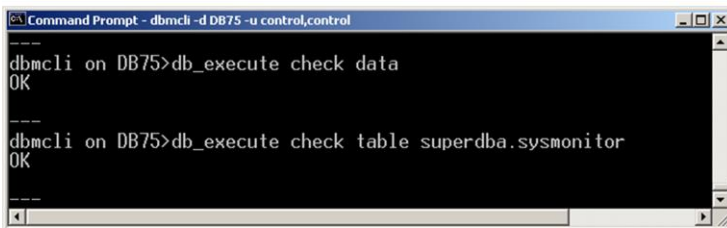
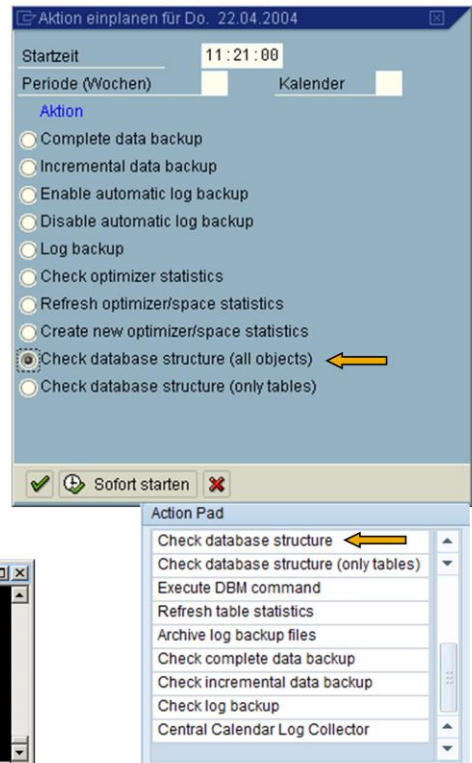
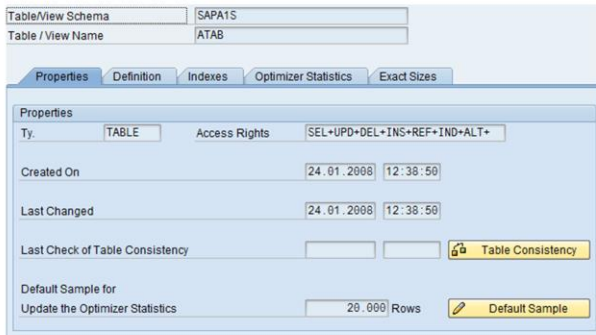
The option **WITH LONG CHECK** makes an additional check of BLOBs. As the name in older releases suggests, a lock is set on tables while the command is executed.

To save time when checking the database, you can use the option **EXCEPT INDEX**. Secondary indexes are not checked in that case.

Consistency Checks (1)



- Transactions DBACockpit, db13
- dbmcli: db_execute check data
- Check Table (f.e. via transaction db50 -> Tables/Indexes)



© SAP 2009 / MaxDB 7.7 Internals - Error Diagnosis/Page 64

The structural consistency of the database can be checked in different ways.

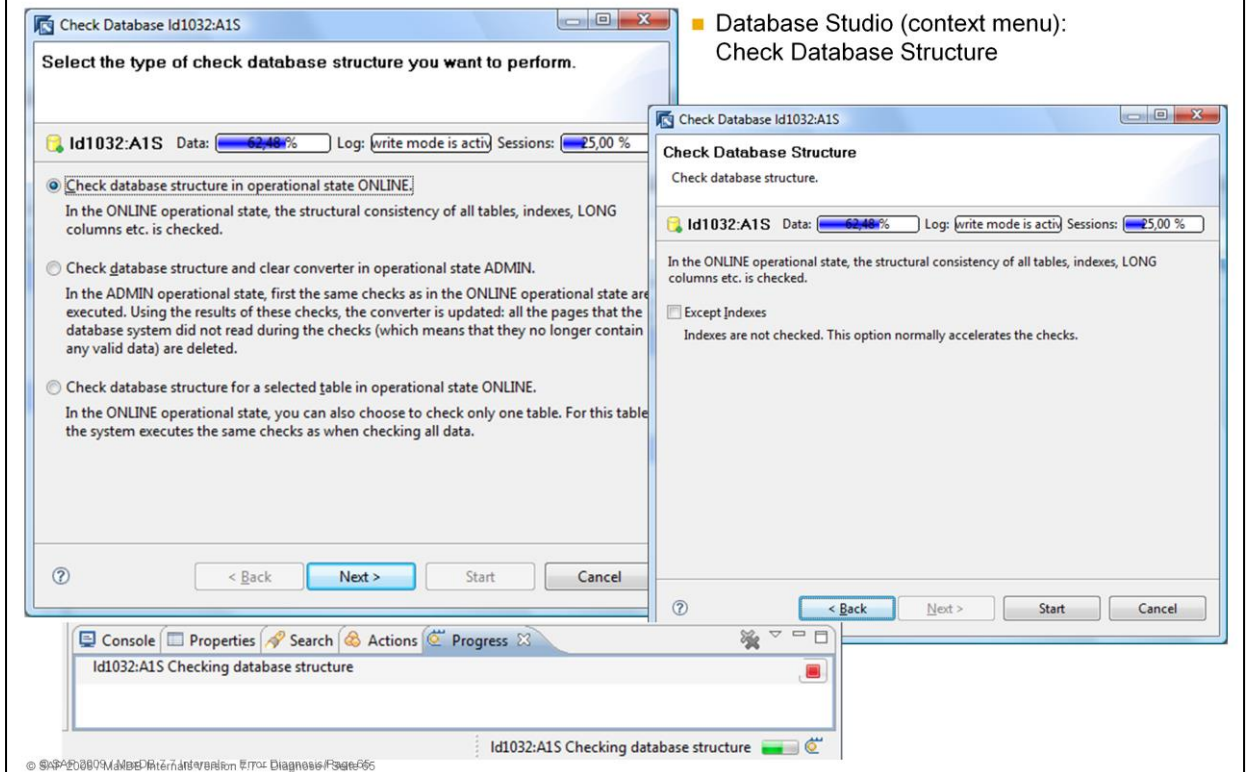
If you choose '**Check database structure (all objects)**' transaction db13, all B* trees, including indexes, are checked. '**Check database structure (only tables)**' checks only the tables.

You can also start consistency checks with the dbmcli:

- dbmcli > db_execute check data (checks all tables and indexes)
- dbmcli > db_execute check table <owner>.<tablename> (selection of a table)

Transaction db50 enables you to select a table for which '**Check Table**' (see next slide) is then initiated.

Consistency Checks (2)



In Database Studio choose **‘Check Database Structure’** in the context menu of the database. There are different choices.

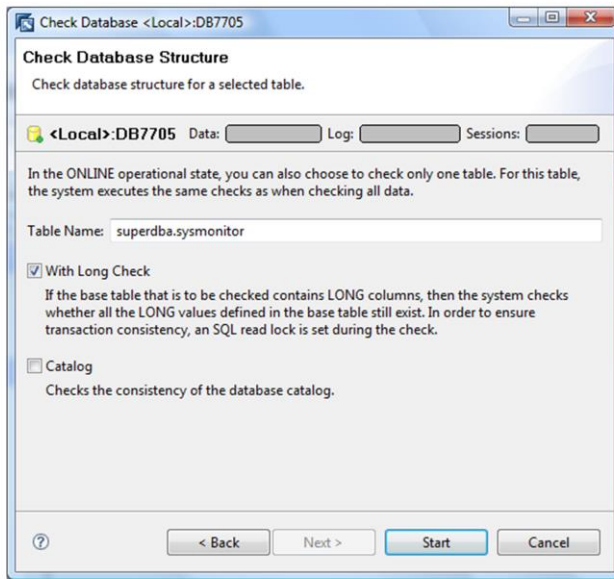
A consistency check can be executed in different operational states of the database. In ONLINE state the structural consistency of all tables, indexes and LOB columns is checked. In ADMIN mode additionally the converter is updated; pages with no more references are deleted.

The check can be restricted to one table.

The amount of data to be checked can be restricted with the option **‘Except Indexes’**. As of version 7.6.01 an EXTENDED check is automatically done; so there is no more need for the database studio to provide this as an option.

Database Studio shows in the status information (Progress) that a CHECK DATA is executed right now. There is no feedback given if the execution was successful. In case of errors a popup is shown describing the first error.

A check of the database structure is time-consuming and CPU-intensive. For a productive system the check should be planned for times of low workload (f.e. on weekends) or, if possible, the check should be done on a separate system copy.



© SAP AG 2009. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or by any information storage or retrieval system, without prior written permission from SAP AG.

The check can be restricted to a specified table.

With **CHECK CATALOG** the catalog information of a chosen table can be checked.

Consistency Checks (4) – Check Success



■ dbm.prt

2004-11-30 11:11:31 0x000009c0 0 DBM command db_execute CHECK DATA
2004-11-30 11:17:37 0x0000079c 0 DBM command db_execute CHECK TABLE

```
dbm.prt - Notepad
File Edit Format View Help
2008-09-17 10:10:41 0x00001058 INF 419 DBMSrv Command 'db_online' has ended with return code 0.
2008-09-17 10:15:00 0x00001058 INF 1 DBMKnI Sending an administrative statement to the database
Statement: CHECK DATA
2008-09-17 10:15:00 0x00001058 INF 8 DBMKnI Received the result of an administrative statement from the database
Statement: CHECK DATA
0x00001058 INF 9 DBMKnI Returncode: 0
```

■ KnIMsg

```
KnIMsg.prt - Notepad
File Edit Format View Help
2008-09-17 10:15:00 RTE 10815: T27 Calling Kernel_Main
2008-09-17 10:15:00 CONNECT 19633: Connect req. (DB7705, T27, connection obj. 0x7FF81C18, Node:'BERI
2008-09-17 10:15:00 Data 82: Start check database
2008-09-17 10:15:00 Data 108: Check database progress report: 27 of about 274 pages checked
2008-09-17 10:15:00 Data 108: Check database progress report: 54 of about 274 pages checked
2008-09-17 10:15:00 Data 108: Check database progress report: 81 of about 274 pages checked
2008-09-17 10:15:00 Data 108: Check database progress report: 108 of about 274 pages checked
2008-09-17 10:15:00 Data 108: Check database progress report: 135 of about 274 pages checked
2008-09-17 10:15:00 Data 108: Check database progress report: 162 of about 274 pages checked
2008-09-17 10:15:00 Data 85: Start LOB checking
2008-09-17 10:15:00 Data 108: Check database progress report: 189 of about 274 pages checked
2008-09-17 10:15:00 Data 108: Check database progress report: 216 of about 274 pages checked
2008-09-17 10:15:00 Data 108: Check database progress report: 243 of about 274 pages checked
2008-09-17 10:15:00 Data 108: Check database progress report: 263 of about 263 pages checked
2008-09-17 10:15:00 Data 79: Check database finished Successfullly
2008-09-17 10:15:00 CONNECT 19651: Connection released (DB7705, T27, connection obj. 7FF81C18)
2008-09-17 10:15:00 RTE 10815: T27 kernel_Main returned
2008-09-17 10:21:45 Savepoint 1: Savepoint (Time) started by T1
```

© SAP AG 2009. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or by any information storage or retrieval system, without prior written permission from SAP AG.

The successful end of CHECK DATA can be checked in **dbm.prt** or in file **KnIMsg (knldiag)**.

If in **dbm.prt** a returncode 0 is delivered the CHECK DATA was successful. In the **KnIMsg** at the end of the progress report a success message is written.

Consistency Checks (5) – Error Situation



```
KnIMsg.prt - Notepad
File Edit Format View Help
11:30:07 RTE 10815: T30 Calling Kernel_Main
11:30:07 CONNECT 19633: Connect req. (DB7705, T30, connection obj. 0x7FF84D38, Node: 'BERD00222089A.dhcp.ber.s
11:30:07 Data 82: Start check database
11:30:07 Data 108: Check database progress report: 27 of about 278 pages checked
11:30:07 Data 108: Check database progress report: 54 of about 278 pages checked
11:30:07 Data 47: Check data on database object failed,KNL_BASE_ERROR=index_not_accessible,ROOT=74502
11:30:07 Data 108: Check database progress report: 81 of about 278 pages checked
11:30:07 Data 47: Check data on database object failed,KNL_BASE_ERROR=index_not_accessible,ROOT=44729
11:30:07 Data 108: Check database progress report: 108 of about 278 pages checked
11:30:07 Data 108: Check database progress report: 135 of about 278 pages checked
11:30:07 Data 108: Check database progress report: 162 of about 278 pages checked
11:30:07 Data 108: Check database progress report: 189 of about 278 pages checked
11:30:07 Data 108: Check database progress report: 216 of about 278 pages checked
11:30:07 Data 108: Check database progress report: 243 of about 278 pages checked
11:30:07 Data 85: Start LOB checking
11:30:07 Data 100: Check database finished unsuccessfully
11:30:07 Index 4: Found not accessible index,FILE_STATE=FileNotAccessible,INTERNAL_FILENAME=0701000000C
11:30:07 Index 4: Found not accessible index,FILE_STATE=FileNotAccessible,INTERNAL_FILENAME=0701000000C
```

Problem Occurred

<Local>-DB7705 Checking database structure (Time of error: 17. September 2008 11:17:40 MESZ)

Reason:
SQL error
-9407,0000000010000000001
100,Check database finished unsuccessfully
4,Found not accessible index

OK Details >>

```
KnIMsg
Popup
dbm.prt

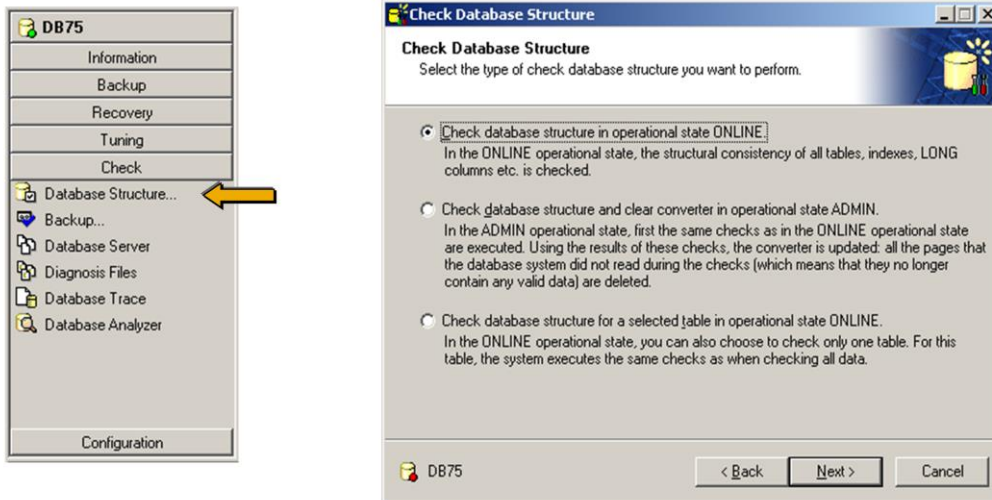
ending an administrative statement to the database
statement: CHECK DATA
ceived the result of an administrative statement from the database
statement: CHECK DATA
Returncode: -9407
Errortext: 0000000020000000002
```

© SAP AG 2007. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or by any information storage or retrieval system, without prior written permission from SAP AG.

If in **dbm.prt** a returncode unequal to 0 is logged, there is an error situation and the defective data object has to be found out. The roots of the defective B* trees are listed in **KnIMsg**.

At the end of CHECK DATA Database Studio opens a popup showing the first error that occurred. Information about further errors has to be gathered from the diagnosis files.

■ DBMGUI: Check -> Database Structure



In the DBMGUI, choose **Check -> Database Structure**. There are several options.

A consistency check can be done in various operational states. In the ONLINE operational state, the structural consistency of all tables, indexes, and LOB columns is checked. In the ADMIN operational state, the converter is also updated; pages that are no longer referenced are deleted.

The check can be restricted to a single table.

Consistency Checks - DBMGUI (2)



Check Database Structure
Make settings.

In the ONLINE operational state, the structural consistency of all tables, indexes, LONG columns etc. is checked.

Extended
Additional checks, for example the ascending order of keys.

Except Index
Indexes are not checked. This option normally accelerates the checks.

DB75 < Back Start

Check Database Structure
Check database structure
Checking of database structure successfully finished.

Checking of database structure successfully finished.

Check Database Structure
Check database structure for a selected table
Specify table and make settings.

In the ONLINE operational state, you can also choose to check only one table. For this table, the system executes the same checks as when checking all data.

Owner: superdba
Table name: sysmonitor

Extended
Additional checks, for example the ascending order of keys.

With Long Check
If the base table that is to be checked contains LONG columns, then the system checks whether all the LONG values defined in the base table still exist. In order to ensure transaction consistency, an SQL read lock is set during the check.

Catalog
Checks the consistency of the database catalog.

DB75 < Back Start Cancel

Different choices:

- Extended
- Except Index
- With Long Check
- Catalog

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 70

The selection options EXTENDED, EXCEPT INDEX, WITH LONG CHECK have already been explained.

CHECK CATALOG enables you to check the catalog information of a selected table.



Backups are checked using a service database.

- No data is written to the disks.
- Service database merely occupies disk space.
- Check of parallel backups is possible.
- DBM command: `recover_check`

Check if the backup

- is complete
- has valid contents

Before you overwrite the backups of one generation, you should make sure that you have an intact backup.

Since the check of a backup is executed on a special service database which merely uses disk space for log files, no resource bottleneck occurs.

The service database is automatically registered when a database instance is created and is stored under the name `.M<version>` (e.g.: `.M750019`).

(In older releases, the Name `_SAPDB<SID>` was used, though the name was shortened to 8 characters, so part of `<SID>` was lost.)

For a restore, the processes are logged in ***KnIMsg (knldiag)*** and the I/O can be monitored with `x_cons`.

Check Backup (2)



Right Click: Check Backup

OR: Check Backup in the context menu of the database

Label	Backup Type	Action	Start	Result	Medium
LOG_000000001	LOG	SAVE WARM	17.09.2008 13:17:11	OK	DB7705.log
DAT_000000001	COMPLETE DATA	SAVE WARM	17.09.2008 13:03:56	OK	DB7705.dat
	HISTLOST				
	HISTLOST				

Start Backup Check

Confirm your selection and start backup check.

<Local>:DB7705 Data: 7,48 % Log: 0,03 % Sessions: 20,00 %

Items: 4 of 4 Backup history

Summary

- Database: <Local>:DB7705
- Version: 7.7.05.07
- Rundirectory: C:\sdb\data\wrk\DB7705
- Recovery Type: SELECTED BACKUP
- Complete Data Backup: DAT_000000001 (374)
- First Log Backup: LOG_000000001 (0)
- Last Log Backup: LOG_000000001 (374)

You have completed the steps required to perform a backup check. Make the specified medium available and choose 'Start'.

Label: DAT_000000001 Medium: DB7705.dat Device/File: c:\backup\DB7705.dat

Start Cancel

In the Database Studio, you can execute a check of a backup by choosing '**Check Backup**'. Then you have to select the appropriate backup medium.

Check Backup (3)



Check Backup of Database <Local>:DB7705

Check Backup Next

<Local>:DB7705 Data: 7,48 % Log: 0,03 % Sessions: 20,00 %

Label	Medium	Device/File
✓ DAT_000000001	DB7705.dat	c:\backu...
➔ LOG_000000001	DB7705.log	DB7705.l...

To continue the check backup make the specified m
➔ Label: LOG_000000001 Medium: DB7705.log Devi

Continue Cancel

Check Backup of Database <Local>:DB7705

Backup Check Successfully Completed

<Local>:DB7705 Data: 7,48 % Log: 0,03 % Sessions: 20,00 %

Label	Date	Result	Transferred	Left	Medium	Device/File	Size	Ne
✓ DAT_000000001	17.09.2008 13:27:49	OK	2.368 KB	0 KB	DB7705.dat	c:\backup\DB7705.dat	278	37
✓ RESTART	17.09.2008 13:32:24	OK						

The backup check was successfully completed.

Close

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 73

A successful execution of a **Check Backup** is marked with a green check mark. After the check of a data backup you will automatically be guided to the check of the corresponding log backups.



- Connect problems
- Log full
- DB full
- Crash, Emergency Shutdown
- System hanger
- Restart problem
- Backup-/Restore problems
- System copy
- I/O problems
- System errors -9026, -9028

```

x5448 on p: /sapmnt/home1/d025448
p34777:e30adm 64> setenv SQLOPT "-X"
p34777:e30adm 65> R3trans -d
This is R3trans version 6.07 (release 620 - 23.04.02 - 10:12:00).
2EETW169 no connect possible: "DBMS = ADABAS D          --- DBNAM
E = ""
R3trans finished (0012).
p34777:e30adm 66> more SAPDB.21710.pct
<html><head><meta http-equiv=Content-type content="text/html; charset=utf-8"><
/head><body ><PRE><PLAINTEXT>
PRODUCT : SAP DB C-PreComp Runtime
DRIVER   : /sapdb/programs/runtime/7301/lib/libpccr
VERSION  : 7.3.1
BUILD    : 015-000-095-214

version :P_1, P_2
SQL STATEMENT : FROM MODULE : dbslada          AT LINE : 5326
Statement Name : :0x000016
OUTPUT : LZU : X321LINUX 7.3.1      Build 015-000-095-106
OUTPUT : PCR : C-PreComp 7.3.1     Build 015-000-095-214
START  : DATE : 2004-04-20      TIME : 0017:45:40
END    : DATE : 2004-04-20      TIME : 0017:45:40

SESSION   : 1;
DATABASE  : DB_000
USERKEY   : DEFAULT
SQLMODE   : SAPR3
SERVERDB  : E30
SERVERNODE: p34777
CONNECT "SAPXX          " IDENTIFIED BY :A SQLMODE SAPR3 ISOLATION LEVEL 0
TIMEOUT 0
SQL STATEMENT : FROM MODULE : dbslada          AT LINE : 7928
Statement Name : :0x000018
SQLCODE: -4008      Unknown user name/password combination
SQLERRD(INDEX_5) : 1      SEC. FOR STATEMENT
START  : DATE : 2004-04-20      TIME : 0017:45:40
END    : DATE : 2004-04-20      TIME : 0017:45:41
    
```

"Connect" problems can usually be reproduced quite easily with R3trans. Call R3trans with option `-d` or `-X`.

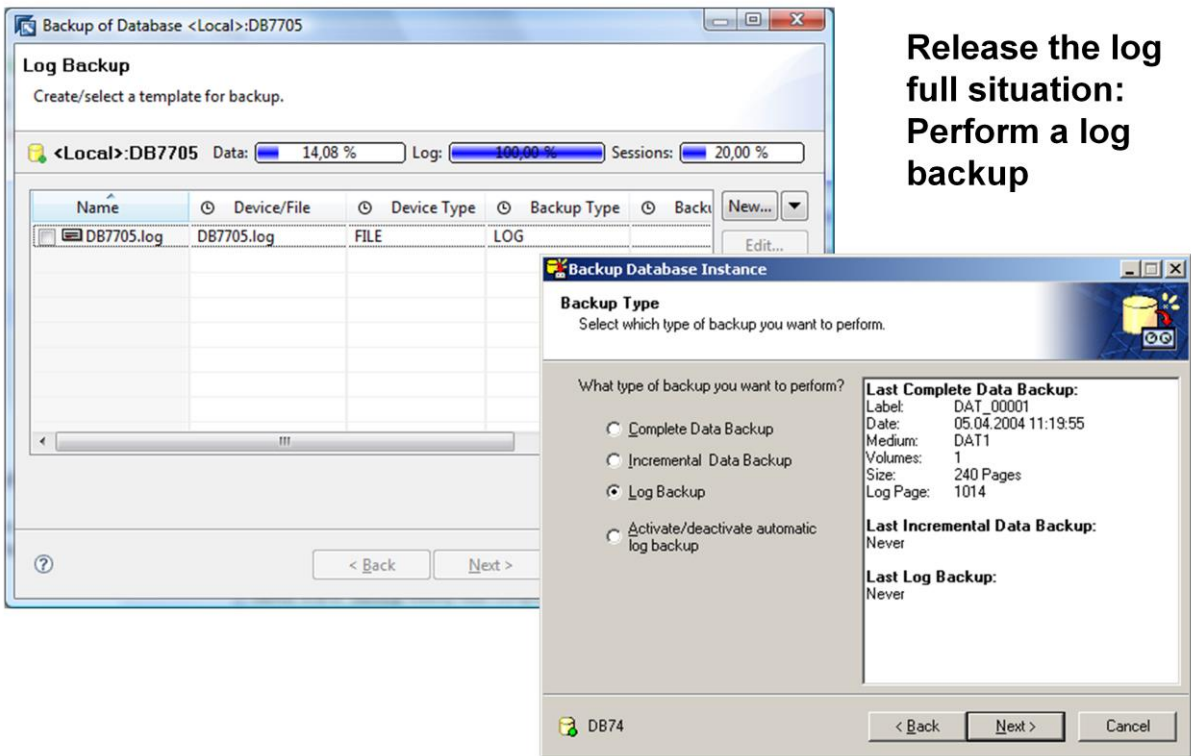
The Precompiler Runtime of the database creates a trace if the variable `SQLOPT` contains the value `"-X"`. The trace is written to the file `SAPDB.<PID of the client process>.pct`.

In this example either the user name or the password is incorrect. The user `SAPXX` is probably not correct.

Check the user specifications with the command `"xuser list"`. Maintain the user data as described in note 39439.

With "Connect" problems, it is often helpful to have a look in the dev logs (the `dev_w*` files from the work directory in the SAP system).

**Release the log full situation:
Perform a log backup**



A Log Full situation can **ONLY** be resolved by executing a log backup.

The Backup Wizard guides you through the required steps.

Adding a new log volume is NOT a possible way of solving the problem. As log volumes are cyclically overwritten, the pointer is usually 'somewhere in the middle' of the device and cannot jump to a new volume.

DB Full - Identification



```
D:\sapdb\indep_data\wrk\DB74>x_cons DB74 sh act
```

```
SERVERDB: DB74
```

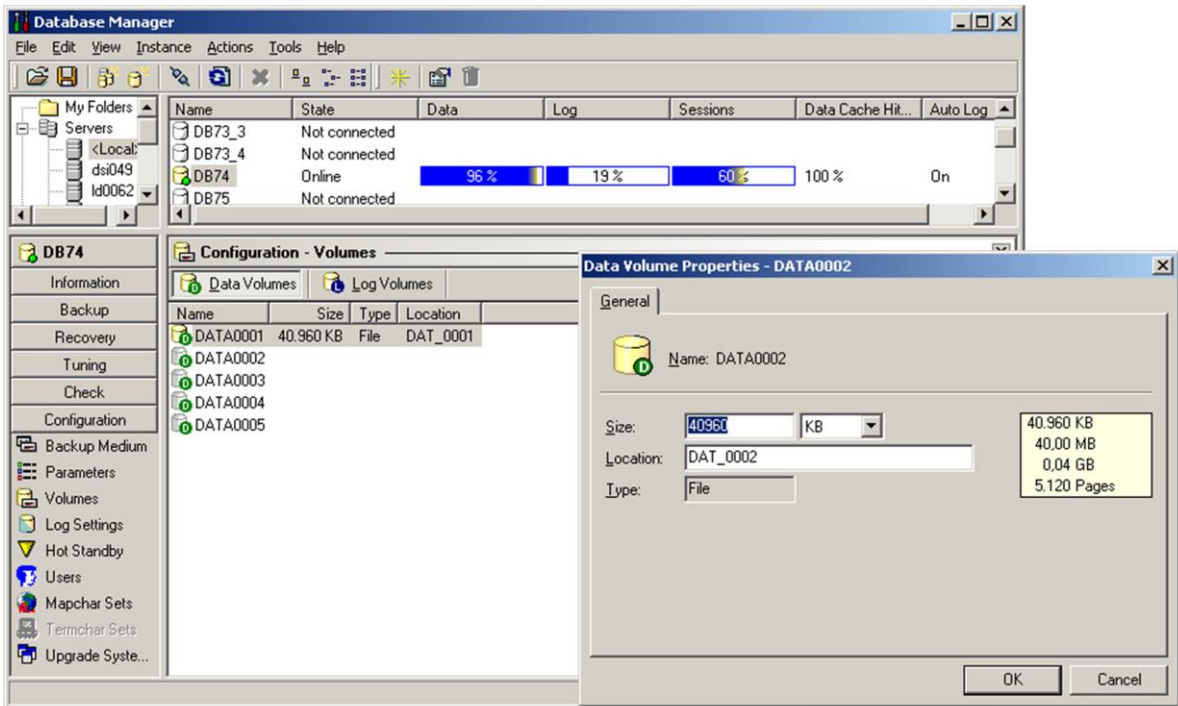
ID	UKT	Win tid	TASK type	APPL pid	Current state	Timeout	Region cnt	Wait item
T14	7	0x890	User	2104	db-full (197)		0 0	43554(s)

```
2004-04-05 12:51:33 0xA24 8 Pager SVP(3) Start write Data
2004-04-05 12:51:33 0xA24 9 Pager SVP(3) Stop Data IO, Pages: 1 IO: 1
2004-04-05 12:51:33 0xA24 10 Pager SVP(3) Start Write Converter
2004-04-05 12:51:33 0x890 19633 CONNECT Connect req. (T15, Node='', PID:2532)
2004-04-05 12:51:33 0xA24 11 Pager SVP(3) Stop Converter IO, Pages: 8 IO: 8
2004-04-05 12:51:33 0xA24 53071 SAVPOINT B20SVP_COMPLETED: 18
2004-04-05 12:51:33 0x890 WRN 39 Convert DB FULL(FBM): Task 14 suspended (45 pages changed)
2004-04-05 12:51:33 0x890 WRN 39 Convert DB FULL(FBM): Task 14 suspended (45 pages changed)
2004-04-05 12:51:33 0xA24 52024 AUTOSAVE 200 pages -> "d:\backup\db74.log.012"
2004-04-05 12:51:33 0x890 WRN 39 Convert DB FULL(FBM): Task 14 suspended (45 pages changed)
2004-04-05 12:51:33 0x890 WRN 39 Convert DB FULL(FBM): Task 14 suspended (45 pages changed)
2004-04-05 12:51:33 0x890 WRN 39 Convert DB FULL(FBM): Task 14 suspended (45 pages changed)
2004-04-05 12:51:34 0x890 WRN 39 Convert DB FULL(FBM): Task 14 suspended (45 pages changed)
2004-04-05 12:51:34 0xA24 52024 AUTOSAVE 696 pages -> "d:\backup\db74.log.012"
2004-04-05 12:51:34 0x948 19626 IO Async I/O thread stopped, 'd:\backup\db74.log.012'
2004-04-05 12:51:34 0xA24 52031 AUTOSAVE ready
2004-04-05 12:51:34 0x890 WRN 39 Convert DB FULL(FBM): Task 14 suspended (45 pages changed)
2004-04-05 12:51:34 0x890 WRN 39 Convert DB FULL(FBM): Task 14 suspended (45 pages changed)
2004-04-05 12:51:34 0x890 WRN 39 Convert DB FULL(FBM): Task 14 suspended (45 pages changed)
2004-04-05 12:51:34 0x890 WRN 39 Convert DB FULL(FBM): Task 14 suspended (45 pages changed)
2004-04-05 12:51:34 0x890 WRN 19651 CONNECT Connection released, T15
2004-04-05 12:51:34 0x890 WRN 39 Convert DB FULL(FBM): Task 14 suspended (45 pages changed)
2004-04-05 12:51:49 0x890 WRN 19633 CONNECT Connect req. (T15, Node='', PID:2532)
2004-04-05 12:51:49 0x890 WRN 39 Convert DB FULL(FBM): Task 14 suspended (45 pages changed)
2004-04-05 12:51:49 0x890 WRN 39 Convert DB FULL(FBM): Task 14 suspended (45 pages changed)
2004-04-05 12:51:49 0x890 WRN 39 Convert DB FULL(FBM): Task 14 suspended (45 pages changed)
2004-04-05 12:51:49 0x890 WRN 39 Convert DB FULL(FBM): Task 14 suspended (45 pages changed)
2004-04-05 12:51:49 0x890 WRN 39 Convert DB FULL(FBM): Task 14 suspended (45 pages changed)
2004-04-05 12:51:49 0x890 WRN 39 Convert DB FULL(FBM): Task 14 suspended (45 pages changed)
2004-04-05 12:51:50 0x890 WRN 39 Convert DB FULL(FBM): Task 14 suspended (45 pages changed)
2004-04-05 12:51:50 0x890 WRN 39 Convert DB FULL(FBM): Task 14 suspended (45 pages changed)
2004-04-05 12:51:50 0x890 WRN 19651 CONNECT Connection released, T15
2004-04-05 12:51:50 0x890 WRN 39 Convert DB FULL(FBM): Task 14 suspended (45 pages changed)
----- current write position -----
```

© SAP 2009 / MaxDB 7.7 Internals - Error Diagnosis/Page 78

A DB Full situation first manifests itself to the user exactly as it does with a Log Full. The user tasks are suspended and no further actions are possible.

Here too, the DBMGUI (without picture), **KnIMsg/knldiag** and x_cons provide information about the hang situation.



© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 79

To resolve a DB Full situation, you have to add another data volume.

To do this with DBMGUI, choose **Configuration -> Volumes**.

The DBMGUI generates default values for the new volume and directs the rest of the process.

Choose **Administration -> Data Area -> New** in Database Studio to create a new volume.

This problem can be prevented by using the AUTOEXTEND functionality. If a defined filling grade is reached automatically a new volume is added.

knldiag

Signal 9: A thread of the database has been killed with „kill -9“.

```

Check - Diagnosis Files - Database Messages
-----
2004-04-19 16:46:02 15897 11561 COMMUNIC Connected T38 local 15808
2004-04-19 16:46:03 15858 11561 COMMUNIC Connecting T39 local 15953
2004-04-19 16:46:03 15897 11561 COMMUNIC Connected T39 local 15953
2004-04-19 16:46:03 15898 11560 COMMUNIC Releasing T39
2004-04-19 16:46:03 15898 12929 TASKING Task T39 started
2004-04-19 16:46:03 15898 11007 COMMUNIC wait for connection T39
2004-04-19 16:46:03 15897 11560 COMMUNIC Releasing T38
2004-04-19 16:46:03 15897 12929 TASKING Task T38 started
2004-04-19 16:46:03 15897 11007 COMMUNIC wait for connection T38
----- current write position -----
++++ Kernel Exit +++++
2004-04-19 16:46:25 0 12847 DBSTATE Kernel exited without core and exit status 0x9
2004-04-19 16:46:25 0 12850 DBSTATE Kernel exited due to signal 9(SIGKILL)
2004-04-19 16:46:25 0 12808 DBSTATE Flushing knltrace pages
2004-04-19 16:46:25 0 11987 dump_rte rtedump written to file 'rtedump'
2004-04-19 16:46:25 0 12696 DBSTATE Change DbState to 'OFFLINE '(28)
    
```

The first place to look after a database crash is **KnIMsg/knldiag**. In this example, the database process on Unix/Linux received signal 9. Signal 9 comes from "outside" and is not caused by the database. On Unix you can find a short description of the signals in the file `/usr/include/sys/signal.h`. Linux stores these definitions in `/usr/include/bits/signum.h`.

Interesting signals:

- SIGILL 4 /* Illegal instruction (ANSI). */
This signal comes from outside and implies a hardware problem.
- SIGABRT 6 /* Abort (ANSI). */
Termination without further information.
- SIGKILL 9 /* Kill, unblockable (POSIX). */
Process/thread was terminated with kill.
- SIGBUS 10 /* bus error */
Error predominantly in the bus system; usually an error in the database software.
- SIGSEGV 11 /* Segmentation violation (ANSI). */
Memory overwrite; usually an error in the database software.

knldiag

Error during write to a log volume

```
Check - Diagnosis Files - Database Messages
17:23:45 16076 11561 COMMUNIC Connected T38 local 16220
17:23:45 16115 11561 COMMUNIC Connected T38 local 16220
17:23:45 16079 ERR 11000 singleio User requested I/O error writing to devno 2 page 638
17:23:45 16079 ERR 34 IOMan Log volume 1 is set to bad
17:23:45 16079 11000 vdetach 'LOG_001' devno 2 T2
17:23:45 16077 12822 TASKING Thread 16168 joining
17:23:45 16168 11566 stop DEVi stopped
17:23:45 16077 12822 TASKING Thread 16169 joining
17:23:45 16169 11566 stop DEVi stopped
17:23:45 16079 ERR 14 IOMan Access error on Log volume 1 blockno 638: I/O error request
17:23:45 16079 ERR 16 IOMan EMERGENCY SHUTDOWN
17:23:45 16079 ERR 11196 DBCRASH vabort:Emergency Shutdown, IOMan_Volume.cpp: 680
17:23:45 16079 ERR 11599 BTRACE ----> Emergency Stack Back Trace <----
17:23:45 16079 ERR 11599 BTRACE (0):0x8491782 [0x41f8d260] (0x0,0x0,0x5000000c,0x1)
17:23:45 16079 ERR 11599 BTRACE (2):0x8495709 [0x41f8d280] (0x402015d8,0x0,0x875a700,0x4002c
17:23:45 16079 ERR 11599 BTRACE (4):0x85d5977 [0x41f8d2b0] (0x40200001,0x2a8,0x41f8d2e0,0x85
17:23:45 16079 ERR 11599 BTRACE (6):0x84e0ca2 [0x41f8d2e0] (0x41f8d320,0x0,0x0,0x86c4382)
17:23:45 16079 ERR 11599 BTRACE (8):0x86174b3 [0x41f8d550] (0x430d110,0x2,0x430cah78,0x41f5
```

Errors while writing to the database log are very critical, in particular if the database is not being mirrored.

Determine the cause of the I/O error. For this example, the error has been simulated.

If the log is mirrored on the database side, then

- provide a new disk for the log volume,
- transfer the database to the ADMIN operational state and execute a restore for the volume using:
dbmcli > db_execute restore log volume '<name of the volume>' ,
- start the database ONLINE.

If the log is mirrored in the system, check whether the error can be corrected in the system. If that is the case, start the database in the ONLINE operational state after the correction has been made.

If the log is irreparable, proceed as follows:

- Create a data backup. The backup is consistent on the basis of the last savepoint.
- Back up the current log area. If the log area cannot be backed up, you can use the generated data backup.
- Initialize the instance when a functional disk has been provided for the log volume. Import the data backup and the generated log backup.
You use Database Studio (**Initialize Database** or **Create Database**) or the Installation Wizard of the DBMGUI to initialize it.

Identification of a System Hanger



The screenshot shows the SAP transaction `sm50` (Prozessübersicht) with a list of processes. A callout box states: "A lot of updates on table ZZTELE occupy the dialog processes". Below the process list, a database console window is open, showing the output of the `x_cons` command. The console output shows several tasks in a `Vwait` state, indicating they are waiting for an SQL lock.

Nr	Typ	Pid	Status	Grund	Start	Err	Sem	CPU	Zeit	Report	Man	Benutzer	Aktion
0	DIA	15449	läuft	ja	1				9	ZFLOCK	000	E30	Update ZZTELE
1	DIA	1410	läuft	ja					20	ZFLOCK	000	E30	Update ZZTELE
2	DIA	1411	läuft	ja					16	ZFLOCK	000	E30	Update ZZTELE
3	DIA	1412	läuft	ja					18	ZFLOCK	000	E30	Update ZZTELE
4	DIA	1413	läuft	ja					4	ZFLOCK	000	E30	Update ZZTELE
5	DIA	1414	läuft	ja						SAPLTHFB	000	E30	
6	UPD	1415	wartet										
7	UPD	1416	wartet										
8	ENQ	1417	wartet										
9	BTC	1418	wartet										
10	BTC	1419	wartet										
11	SPD	1421	wartet										
12	UP2	1423	wartet										

ID	UKT	UNIX	TASK	APPL	Current	Timeout	Region	Wait
		tid	type		pid	priority	cnt	item
T37	6	1232	User		1411	Vwait	0 0	34 0(s)
T39	6	1234	User		1413	Vwait	0 0	20 0(s)
T40	6	1235	User		1410	Vwait	0 0	37 0(s)
T60	7	1255	User		15449	Vwait	0 0	27 0(s)
T65	7	1260	User		1412	Vwait	0 0	36 0(s)

Active tasks have the status **Vwait** (SQL lock)

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 82

This example shows a system hanger situation.

Transaction `sm50` or `sm66` show numerous dialog processes that are executing updates on table `ZZTELE`. If transactions `sm50` and `sm66` are no longer usable because all dialog processes are occupied, call the program `dpmon` on the operating system level. In the 'Menu' there you'll see a comparable output.

The database console shows the respective tasks in the `Vwait` status. The tasks are waiting for the release of an SQL lock.

At present no other task is active in the database; that is, the lock holder is active in the application or waiting for user input.

System Hanger: Who is the Lock Holder?



The screenshot shows the SAP System Hanger interface. The top bar includes 'Wartesituationen Bearbeiten Springen System Hilfe' and the SAP logo. Below the title 'Exklusive SQL-Wartesituationen', there is a table titled 'Exklusive Wartesituationen'. The table has columns: Task-ID, Appl.-ID, Appl.-Server, Spe..., Sperrart, Tabellename, Task-ID, Appl.-ID, Appl.-Server, and Warte. The first row is highlighted in yellow and contains the following data: Task-ID: 48, Appl.-ID: 9008, Appl.-Server: dewdfm189.wdf.sap.corp, Spe...: [lock icon], Sperrart: row_exclusive, Tabellename: ZZTELE, Task-ID: 40, Appl.-ID: 1410, Appl.-Server: p34777, Warte: [lock icon]. A callout box points to this row with the text: 'Task 48 keeps the lock and belongs to application process 9008. It is not known as work process.'

Task-ID	Appl.-ID	Appl.-Server	Spe...	Sperrart	Tabellename	Task-ID	Appl.-ID	Appl.-Server	Warte
48	9008	dewdfm189.wdf.sap.corp	[lock icon]	row_exclusive	ZZTELE	40	1410	p34777	[lock icon]
48	9008	dewdfm189.wdf.sap.corp	[lock icon]	row_exclusive	ZZTELE	65	1412	p34777	[lock icon]
48	9008	dewdfm189.wdf.sap.corp	[lock icon]	row_exclusive	ZZTELE	37	1411	p34777	[lock icon]
48	9008	dewdfm189.wdf.sap.corp	[lock icon]	row_exclusive	ZZTELE	60	15449	p34777	[lock icon]
48	9008	dewdfm189.wdf.sap.corp	[lock icon]	row_exclusive	ZZTELE	39	1413	p34777	[lock icon]

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 83

Transaction db50 provides more information under 'SQL Locks -> Wait Situations'. All waiting tasks are waiting for task 48. This task belongs to application process 9008 on the server dewdfm189. The server is not a SAP application server.

User operations generally have priority. Task 48 should therefore be forced to release the lock.

System Hanger: What is Task 48 doing?



The screenshot shows the SAP Task Manager interface. The 'Benutzer-Tasks' tab is selected, displaying a table of active tasks. The table has columns for ID, Thread-ID, Task-Typ, A., Task-Zustand, Zustandsbeschreibung, Wart..., Warten..., and Applikation... The following table represents the data shown in the screenshot:

ID	Thread-ID	Task-Typ	A.	Task-Zustand	Zustandsbeschreibung	Wart...	Warten ...	Applikation...	A
37	1230	User		vwait		48		1411	p
39	1230	User		vwait		48		1413	p
40	1230	User		vwait		48		1410	p
60	1255	User		vwait		48		15449	p
64	1255	User		Running				1414	p
65	1255	User		vwait		48		1412	p

A callout box at the bottom of the screenshot contains the text: "Task 48 is not active in the database. Task 64 works for transaction db50."

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 84

Under '**Current Status -> Kernel Threads -> Task Manager**', transaction db50 displays the task activities. Task 48 is not active. The running task 64 formats the information for db50 itself.

To terminate 48, display all user tasks. Select task 48 and choose 'End Session'.

It is not possible to terminate a command for task 48 if task 48 is not active. The lock can only be released by terminating the transaction. If the locking transaction is not active in the database, its transaction can be terminated by closing the session.

If a work process which is holding locks is active on the database, the termination of the command leads to the termination of the transaction. When it receives return code -102, the SAP system rolls back the transaction and writes a short dump.

Standstill of the System: Eliminate Cause



```

d025448 on p: /sapmnt/home1/d025448
p34777:e30adm 87> x_cons E30 sh act
SERVERDB: E30
ID   UKT UNIX   TASK      APPL Current      Timeout Region      Wait
   tid type      pid state          priority cnt try    item
T37  6  1232 User      1411 Vwait            0 0   48   154 0(s)
T39  6  1234 User      1413 Vwait            0 0   48   151 0(s)
T40  6  1235 User      1410 Vwait            0 0   48   157 0(s)
T60  7  1255 User     15449 Vwait            0 0   48   152 0(s)
T65  7  1260 User      1412 Vwait            0 0   48   155 0(s)
p34777:e30adm 88> x_cons E30 cancel t48
p34777:e30adm 89> x_cons E30 sh act
SERVERDB: E30
ID   UKT UNIX   TASK      APPL Current      Timeout Region      Wait
   tid type      pid state          priority cnt try    item
T37  6  1232 User      1411 Vwait            0 0   48   166 0(s)
T39  6  1234 User      1413 Vwait            0 0   48   163 0(s)
T40  6  1235 User      1410 Vwait            0 0   48   169 0(s)
T60  7  1255 User     15449 Vwait            0 0   48   164 0(s)
T65  7  1260 User      1412 Vwait            0 0   48   167 0(s)
p34777:e30adm 90> x_cons E30 kill t48
p34777:e30adm 91> x_cons E30 sh act
SERVERDB: E30
ID   UKT UNIX   TASK      APPL Current      Timeout Region      Wait
   tid type      pid state          priority cnt try    item
p34777:e30adm 92>
    
```

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 85

The action '**Terminate Command**' in transaction db50 corresponds to the console command '*cancel <task>*'. You terminate user sessions with '*kill <task>*'.

Terminating the locking transaction can take some time. MaxDB works with cooperative multitasking. The tasks are not managed through a dispatcher instance. Some actions only check whether the termination flag is set every 30 seconds.

In the console output, if the termination flag was set, this is indicated by an exclamation mark. If the task remains active (in particular in the Running and I/OWait statuses), it executes a rollback of the changes that have already been made.

Restart Failed: Lack of Memory



knldiag

Check - Diagnosis Files - Database Messages

```

10:25:00 19324 12931 INFO maximum cpu time: unlimited
10:25:00 19324 12932 INFO maximum number of processes: 0 MB
10:25:00 19324 12891 INFO maximum core size: 0 MB
10:25:00 19324 12892 INFO maximum number of open files: 1024
10:25:00 19324 12933 INFO maximum file size: unlimited
10:25:00 19324 12895 INFO maximum number of threads: 1024
10:25:00 19324 12934 INFO maximum stack size: 1 MB
10:25:00 19324 12935 INFO lockable memory size: unlimited
10:25:00 19324 12893 INFO maximum virtual memory size: 488 MB
10:25:00 19324 12936 INFO resident set size size: 488 MB
10:25:00 19324 12894 INFO maximum heap size: 488 MB
10:25:00 19324 WNG 12448 MEMORY memory size needed exceeds virtual address space!
10:25:00 19324 WNG 12447 MEMORY memory size needed exceeds heap size!
10:25:00 19324 12898 ENVIRON Environment dump start
    
```

```

10:25:04 19326 ERR 11872 MEMORY Not enough memory for I/O cache wanted : 800000 kB!
+++++ Kernel Exit +++++
    
```

d025448 on p:/sapmnt/home1/d025448

```

2004-04-20 10:24:50 19318 12898 ENVIRON Current user id 3488 effective id
3488
2004-04-20 10:24:50 19318 12898 ENVIRON Current group id 1008 effective id
1008
2004-04-20 10:24:50 19318 12898 ENVIRON cpu time unlimited
2004-04-20 10:24:50 19318 12898 ENVIRON number of processes 4096
2004-04-20 10:24:50 19318 12898 ENVIRON number of open files 1024
2004-04-20 10:24:50 19318 12898 ENVIRON core size 0 KBytes
2004-04-20 10:24:50 19318 12898 ENVIRON file size unlimited
2004-04-20 10:24:50 19318 12898 ENVIRON heap memory size 500000 KBytes
2004-04-20 10:24:50 19318 12898 ENVIRON stack memory size unlimited
2004-04-20 10:24:50 19318 12898 ENVIRON lockable memory size unlimited
2004-04-20 10:24:50 19318 12898 ENVIRON virtual memory size 500000 KBytes
2004-04-20 10:24:50 19318 12898 ENVIRON resident set size 500000 KBytes
2004-04-20 10:24:50 19318 12898 ENVIRON Resource limit dump completed
2004-04-20 10:24:50 19318 12898 ENVIRON Environment dump start
2004-04-20 10:24:50 19318 12898 ENVIRON LESSKEY=/etc/lesskey.bin
2004-04-20 10:24:50 19318 12898 ENVIRON MANPATH=/usr/local/man:/usr/share
/man:/u
/sapdb/data/wrk/xserver_p34777.prt" [readonly] 822L, 65536C 10,1 1%
    
```

xserver*.prt

In this example the database cannot transfer to the ADMIN operational state because the operating system cannot allocate enough memory.

The file *knldiag* shows an excerpt of the limitations for the user. These limitations are inherited from the owner of the *x_server* process.

When you start the *x_server*, make sure that the user has set sufficient limitations. On Unix/Linux, you set limitations either with *limit* or *ulimit*, depending on the shell.

Check the limitations of the *x_server* process in the file <indepdatapath>/wrk/xserver*.prt.

This case can be resolved by setting the limitations correctly and restarting the *x_server*.



Overview of backup activities: dbm.knl

Formatted display
in DBMGUI:
Information -> Backup
History

or

in Database Studio:
Administration ->
Backup

Label	Action	Beginning	Result	Medium	Size (Pages)	Volumes	Next Log Page	From Page	To Page
DAT_00005	SAVE WARM	29.07.2003 17:12:43	OK	Data_tmp	468288	1	287309		
DAT_00004	SAVE WARM	12.06.2003 18:23:12	OK	Data_tmp	467536	1	1394		
LOG_00000	SAVE WARM	12.06.2003 17:41:11	Error: (-123)						
LOG_00000	SAVE WARM	12.06.2003 17:38:34	Error: (-123)						
HISTLOST		12.06.2003 17:25:58	OK						
LOG_00000	SAVE WARM	12.06.2003 17:22:40	Error: (-123)						
HISTLOST		12.06.2003 17:22:30	OK						
DAT_00003	SAVE COLD	12.06.2003 17:06:28	OK	Data	467200	1	21596		
DAT_00002	SAVE COLD	12.06.2003 16:56:57	OK	Data	467192	1	21057		
HISTLOST		12.06.2003 16:55:54	OK						
DAT_00001	RESTORE	12.06.2003 16:42:28	OK	Data_tmp	467184	1	20981		
DAT_00001	SAVE WARM	10.06.2003 17:34:21	OK	Data	467184	1	20981		
HISTLOST		26.05.2003 15:43:57	OK						

Label	Backup Type	Action	Start	Result	Medium	Size	Devic...	Next ...	From...	To P...
LOG_000000003	LOG	SAVE WARM	04.03.2008 11:03:48	OK	LOG	204.864 KB	0		51245	76847
LOG_000000002	LOG	SAVE WARM	04.03.2008 11:03:40	OK	LOG	204.864 KB	0		25645	51244
LOG_000000001	LOG	SAVE WARM	04.03.2008 11:03:30	OK	LOG	204.864 KB	0		0	25644
DAT_000000003	COMPLETE DATA	SAVE WARM	04.03.2008 10:17:43	OK	PIPE	54.634.368 KB	1	93024		
HISTLOST										
DAT_000000002	COMPLETE DATA	SAVE WARM	18.02.2008 14:17:12	OK	PIPE	54.048.896 KB	1	1180135		
HISTLOST										
DAT_000000001	COMPLETE DATA	SAVE WARM	04.02.2008 13:22:18	OK	PIPE	53.791.808 KB	1	488815		

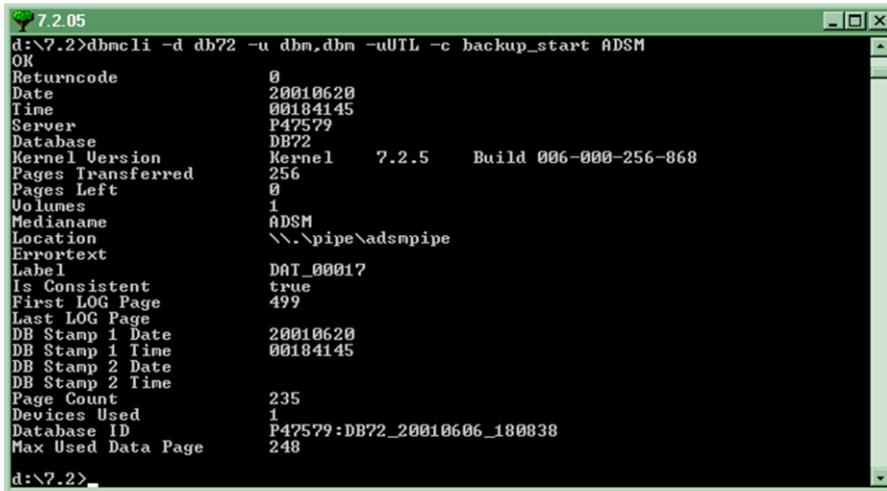
© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 87

The file **dbm.knl** presents a first overview of which backups and which restore activities were successfully executed; or for a more orderly display in the form of a backup history in the Database Studio choose **Administration -> Backup** or **Information->Backup History** in the DBMGUI.

If any errors have occurred, the causes are noted in brief. More precise information can be found in **KnIMsg/knldiag**.

Up to version 7.6 the file **dbm.utl** provides information about backups. As of version 7.7 the information from **dbm.utl** is integrated into other diagnosis files (**KnIMsg/dbm.prt**).

If external backup tools (Networker, ADSM, Omniback, etc.) and the backint interface are being used, you should also check their logs, which are described in the following.



```
d:\7.2>dbmcli -d db72 -u dbm.dbm -uUTL -c backup_start ADSM
OK
Returncode          0
Date                20010620
Time                00184145
Server              P47579
Database            DB72
Kernel Version      Kernel    7.2.5    Build 006-000-256-868
Pages Transferred   256
Pages Left          0
Volumes            1
Medianame           ADSM
Location            \\.\pipe\adsnpipe
ErrorText
Label               DAT_00017
Is Consistent       true
First LOG Page      499
Last LOG Page
DB Stamp 1 Date     20010620
DB Stamp 1 Time     00184145
DB Stamp 2 Date
DB Stamp 2 Time
Page Count          235
Devices Used        1
Database ID         P47579:DB72_20010606_180038
Max Used Data Page  248
d:\7.2>
```

One example shows the execution of a backup via ADSM using the dbmcli.

You execute a backup of the database with the backup_start command.

As the DBM server can derive the desired backup tool from the names of the medium, there is, in this case, no difference to a backup without a tool.

```
7.2.05 - dbmcli -d db72 -u dbm,dbm
dbmcli on db72>recover_start ADSM DATA EBID "P47579_DB72_2001.06.15_13.56.30_SAUEDTNCHK_ADSM"
OK
Returncode          0
Date                20010620
Time                00192422
Server              P47579
Database            DB72
Kernel Version      Kernel    7.2.5    Build 006-000-256-868
Pages Transferred   256
Pages Left          0
Volumes            1
Medianame           ADSM
Location            \\.\pipe\adsmpipe
Errortext
Label               DAT_00011
Is Consistent       true
First LOG Page      476
Last LOG Page
DB Stamp 1 Date     20010615
DB Stamp 1 Time     00135630
DB Stamp 2 Date
DB Stamp 2 Time
Page Count          235
Devices Used        1
Database ID         P47579:DB72_20010606_180838
Max Used Data Page  248
-----
dbmcli on db72>
```

A restore is executed using the commands **recover_start** and **recover_replace** (for restoring multiple backups).

The keyword EBID (or ExternalBackupID) is followed by a list (separated by commas) of external backup IDs (only one in the present example). If the lists contains blanks, it must be written in quotation marks.

```
7.2.05 - dbmcli -d db72 -u dbm,dbm
d:\7.2>dbmcli -d db72 -u dbm,dbm
dbmcli on db72>backup_ext_ids_get ADSM
OK

-----
dbmcli on db72>backup_ext_ids_list
OK
END
AVAILABLE!P47579_DB72_2001.06.15_13.56.30_SAVEDTINCHK_ADSM!DATA RECOVERY !2001-06-15 13:56:30!
AVAILABLE!P47579_DB72_2001.06.20_10.52.56_SAVEDATA_ADSM!DATA MIGRATION!2001-06-20 10:52:56!
AVAILABLE!P47579_DB72_2001.06.20_18.41.45_SAVEDTINCHK_ADSM!DATA RECOVERY !2001-06-20 18:41:45!

-----
dbmcli on db72>_
```

As more than one DBM server command is required for displaying the external backup ID, you have to use an interactive dbmcli session.

The columns of the displayed list are separated by the pipe (|) character.

The list has the format:

<Availability>|<External Backup ID>|<backup typ>|<date_time>|

If an output of the **backup_ext_ids_list** command contains a line with the keyword CONTINUE followed by OK, the next part can be queried with the **backup_ext_ids_listnext** command.

Example: Backup / Restore (1)



The restore from the system <SID> with backup from 15.01.01 was killed with:

```
2004-01-16 20:25:53      13 ERR 54001 I/O          page 00C29008010D0200...008
2004-01-16 20:25:53      13 ERR 52015 RESTORE      write/check count mismatch
2004-01-16 20:25:53      12 ERR 52012 RESTORE      error occurred, basis_err 30
2004-01-16 20:25:53      12 ERR 51080 SYSERROR     -9026 Message not available
```

The restore with the backup from 16.01 was killed with:

```
2004-01-16 17:11:31      13 ERR 54001 I/O          page 00AA5684010D0200...203
2004-01-16 17:11:31      13 ERR 52015 RESTORE      write/check count mismatch
2004-01-16 17:11:31      12 ERR 52012 RESTORE      error occurred, basis_err 30
2004-01-16 17:11:31      12 ERR 51080 SYSERROR     -9026 Message not available
```

In the present example, the restoring of a backup terminated with a system error.

First, the system attempts to repeat the procedure with various backups. It turns out that several backups have already been affected and that a restore returns error -9026.

At this stage, the user should check the logs to see what they say about the backups, e.g. whether they were successful, etc.

Example: Backup / Restore (2)



UTLPRT:

=====

```
2004-01-19 01:00:11 400B1E0B0006 0000 SDB SAVE DATA QUICK TO '/nsr/sap
2004-01-19 03:48:08 400B1E0B0006 0001 RET RETURNCODE 0 ←
[...]
```

DBMPRT:

=====

```
2004-01-19 01:00:11 0x00007227 0 DBM
  command backup_start nsr_full RECOVERY DATA AUTOIGNORE
2004-01-19 03:49:45 0x00007227 ERR -24920 DBM
  ERR_BACKUPOP: backup operation was unsuccessful
                0x00007227 ERR -24920 DBM
  The backup tool failed with 0 as sum of exit codes.
```

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 92

Looking at **dbm.utl** is not sufficient! The return code 0 here shows only that the backup was successful from the database's point of view. In other words, it correctly delivered all database pages to the pipe of the external database tool.

In **dbm.prt** we see that the backup could not be completed successfully. The cause is not immediately visible in this file, but the tool has signaled that the backup failed from its point of view.

Backups that have the return code 0 in **dbm.utl** (that is, on the kernel side) but failed according to **dbm.prt** are identified as having failed in the backup history (DBMCLI -> backup_history_list as well as in DBMGUI). The error code, then, is the error code of the backup/restore (generally -24920).

Example: Backup / Restore (3)



BACKEBL:

=====

2004-01-19 03:49:44

Analyzing output of NetWorker.

Have found a completion message in the NetWorker output file

'/var/tmp/temp1074470411-1' in line:

save: S6F level=full, 130 GB 02:43:44 5 files

Have found a savetime message in the NetWorker output file

'/var/tmp/temp1074470411-1' in line:

completed savetime=1074470655

NetWorker has saved only 140123308032 bytes ←

(the database kernel has saved 159754027008 bytes).

The backup size of database and NetWorker are different.

2004-01-19 03:49:44

Filling reply buffer.

Have encountered error -24920:

The backup tool failed with 0 as sum of exit codes.

Constructed the following reply:

ERR

-24920,ERR_BACKUPOP: backup operation was unsuccessful

The backup tool failed with 0 as sum of exit codes.

Reply buffer filled.

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 93

The log of the backup tool provides information as to why the backup was not considered successful.

There is a big discrepancy between the number of bytes backed up by the database kernel and the number of bytes given by the NetWorker which cannot be explained by rounding errors.

Here we cannot identify the cause, which could only be determined in cooperation with the NetWorker manufacturer Legato. While a file system backup was in progress, the NetWorker failed to end the data backup correctly when it accessed the data backup pipe.

MaxDB now ensures that file system and database backups remain separate.

Example: Faulty Tape Devices



1st attempt: RESTORE DATA was successful RESTORE LOG fails

The recovery of the system <SID> breaks with

```
2004-01-16 20:11:11 13 ERR 52015 RESTORE bad log page 98897294
2004-01-16 20:11:25 13 ERR 52608 RESTART LOCAL: failed
2004-01-16 20:11:29 12 ERR 52012 RESTORE error occurred, basis_err 90
2004-01-16 20:11:29 12 ERR 51080 SYSERROR -9030 Message not available
```

2nd attempt: RESTORE DATA fails, but is later successful with the identical backup

Excerpt from the Backup History:

```
4007CDD70001|DAT_00210|RESTORE |2004-01-13 23:00:20|
2004-01-13 23:00:20|2004-01-16 12:46:10|2004-01-16 12:51:56|
98576129| |YES|nsr_comp
10624| 0| -9026|
```

```
4007D1BF0002|DAT_00210|RESTORE |2004-01-13 23:00:20|
2004-01-13 23:00:20|2004-01-16 13:01:55|2004-01-16 19:39:16|
98576129| |YES|nsr_comp
20023232| 1| 0|
```

In this example, the restore of a log backup terminates with system error -9030 (bad log page).

During the attempt to repeat the restore process completely, there was already a problem with the data backup. Subsequently, however, the data backup (with the same label) was successfully recovered.

This non-deterministic behavior suggests a problem with the tape peripherals. In such cases, checking the tape drives and the controller and changing the defective device will solve the problem.

With luck, the problem will have been merely a read error; in the worst case scenario, the tapes will already have been incorrectly filled.

Error during System Copy



The screenshot shows the 'Initialize Database Instance' dialog box in the SAP DBMGUI. The 'Create and start instance' option is selected, with 'superdba' as the user name. The 'Create instance for recovery' option is unselected. Below the dialog, a command window shows the following commands being executed:

```

BM ERR_SQL: sql error
BM 100,Row not found
BM command trace_show
BM command trace_flush
BM command trace_prot
BM command db_online -f
BM command backup_save "data" DATA RECOVERY
BM command db_offline
BM command db_admin -f
BM command db_activate
BM command load_systab
BM command db_offline
BM command db_admin
BM command recover_start data
BM command db_online
    
```

Yellow arrows point from the 'db_activate' and 'db_admin' commands to the error log in the Notepad window below. The Notepad window shows the following log entries:

```

2005-01-27 15:31:02 0xDB4 54003 dynpool LOCK SupplyPoolSegments: 75
2005-01-27 15:31:02 0xDB4 54003 dynpool LOCK SupplyPoolSize : 541504
2005-01-27 15:31:02 0xDB4 54003 dynpool LOCK objColl_list size : 2816
2005-01-27 15:31:02 0xDB4 54003 dynpool LOCK objReq_list size : 1408
2005-01-27 15:31:02 0xDB4 ERR 29 Log DBIdentifier of DataVolume (P124984.wdf.sap.corp:HOTELDB_20041223_105938)
and LogVolume (P124984.wdf.sap.corp:HOTELDB_20050127_152929)
2005-01-27 15:31:02 0xDB4 19616 VOLUMEIO Detaching volume DISKL001
2005-01-27 15:31:02 0x140 19618 VOLUMEIO Single I/O detach, 'DISKL001', UKT:3
2005-01-27 15:31:02 0x2E8 19614 DBSTATE I/O thread for 'DISKL001' stopped
2005-01-27 15:31:02 0xDB4 WRN 23 Admin Initialization of log for 'restart' failed with 'LogAndDataIncompatible'
2005-01-27 15:31:02 0xDB4 37 Admin kernel state: 'OFFLINE' reached
2005-01-27 15:31:02 0xDB4 19620 DBSTATE Shutdown normal requested
2005-01-27 15:31:02 0x13C 19617 VOLUMEIO Single I/O attach, 'knltrace', UKT:1
2005-01-27 15:31:02 0xDD4 19684 TASKING Tracewriter termination timeout: 1200 sec
2005-01-27 15:31:02 0x13C 19619 TASKING Releasing tracewriter
2005-01-27 15:31:02 0xDD4 19600 VERSION 'kernel 7.5.0 Build 018-123-079-776'
2005-01-27 15:31:02 0xDD4 19654 DBSTATE SERVERDB 'HOTELDB' has stopped
-----
----- current write position -----
    
```

One typical user error with system copies is to first completely install a database (including restart and loading the system tables) and only then import a backup. This often leads to confusion when it then turns out that it is no longer an "empty DB".

The user chose '**Create and start instance**' instead of '**Create instance for recovery**' in the DBMGUI.

The individual steps can be viewed in *dbm.prt*. We can see that db_activate was carried out, which represents the first restart of a DB, and that only afterwards a backup was imported.

The system's reaction to this error is somewhat different than in earlier versions (<= 7.3). The system no longer issues return code -8003 „Log and Data must be compatible“ because the database is immediately transferred to the OFFLINE operational state and the DBMGUI no longer receives a message about the exact cause of the error. This ensures that the memory areas can be completely cleared.

KnIMsg reports that the cause of the shutdown here again was LogAndDataIncompatible, albeit without the familiar return code.

It is also noted that the DBIdentifiers of the data and log volumes do not match.

I/O Error: Verification of Checksum in Index Page



The screenshot displays two windows from the SAP Database Manager. The top window, titled 'Check - Diagnosis Files - Database Messages', shows a log of database events. A red box highlights several error messages related to checksum mismatches and index corruption:

```
2004-04-21 17:21:18 25783 11561 COMMUNIC Connected T38 local 25859
2004-04-21 17:21:18 25783 ERR 4 Data Checksum mismatch; calculated: 618008976 found: 618000000
2004-04-21 17:21:18 25783 ERR 12 IOMan Bad page on Data volume 1 blockno 1205
2004-04-21 17:21:20 25783 ERR 4 Data Checksum mismatch; calculated: 618008976 found: 618000000
2004-04-21 17:21:20 25783 ERR 12 IOMan Bad page on Data volume 1 blockno 1205
2004-04-21 17:21:22 25783 ERR 4 Data Checksum mismatch; calculated: 618008976 found: 618000000
2004-04-21 17:21:22 25783 ERR 12 IOMan Bad page on Data volume 1 blockno 1205
2004-04-21 17:21:24 25783 ERR 24 IOMan Bad data page 60863
2004-04-21 17:21:24 25783 ERR 53000 B*TREE 070100000000000002A7000000000000
2004-04-21 17:21:24 25783 ERR 53000 B*TREE INDEX ROOT 105515
2004-04-21 17:21:24 25783 ERR 53250 B*TREE BAD INDEX 105515 (ROOT)
2004-04-21 17:21:24 25783 ERR 53250 B*TREE BAD INDEX 105515 (ROOT)
2004-04-21 17:21:24 25783 ERR 51080 SYSERROR -9041 BD Index not accessible
```

The bottom window, titled 'Database Manager', shows the 'Recovery - Indexes' dialog. The 'Index' tab is selected, and the index 'TEST_1' is highlighted in the list. The dialog includes fields for 'Owner', 'Table Name', and 'Index Name', and a 'Choose the Action' section.

© SAP 2009 / MaxDB 7.7 Internals - Error Diagnosis/Page 96

After a data page was read from a data volume, checksum 618008976 was calculated. Before writing the block, checksum 618000000 was calculated and written to the block. Apparently the block is not situated correctly on the disk.

This read I/O is repeated twice. If the error occurs every time, the database assumes that the block is defective. This is a block for an index (secondary key tree). The index is marked as BAD.

Check the I/O system. If the damage to the I/O system can be repaired, you can delete the index and then regenerate it.

Under '**Recovery -> Index**', the DBMGUI displays the indexes marked as BAD. You can select the index and recreate it.

Regardless of whether or not you are able to identify errors in the I/O system, it is a good idea to run a **CHECK DATA** in such a case.

DB50: Recreate Index



Table / view information

Table/View Schema: MONA
Table / View Name: BKPF

Properties Definition Indexes Optimizer Statistics Exact Sizes

Inactive Indexes Unused Indexes Bad Indexes

Table / Index / Column	P...	Sort	Type	C...	A..	U...	Accesses	Reset Date	Time	Creation D...	Time
MONA.BKPF											
· BKPF-1							0	26.07.2006	09:06:17	26.07.2006	09:06:17
· MANDT	1	ASC									
· BUKRS	2	ASC									
· BSTAT	3	ASC									
· XBLNR	4	ASC									
· BKPF-2							10	26.07.2006	09:06:17	26.07.2006	09:06:17
· MANDT	1	ASC									
· BUKRS	2	ASC									
· BSTAT	3	ASC									
· BUDAT	4	ASC									
· BKPF-3							0	26.07.2006	09:06:17	26.07.2006	09:06:17
· MANDT	1	ASC									
· BUKRS	2	ASC									

Restore Bad Indexes

Restoring indexes can take a long time, depending on the table size.
While an index is being restored, the associated table cannot be accessed.

Restoring indexes should therefore be performed at a time of minimal workload.
You should restore indexes of large tables in the background.

Do you want to restore the selected bad index now?

In Dialog In the Background X

Corrupted indexes (different example) can directly be recreated by using transaction db50. Mark the corrupted index and choose **Restore Index**.

Attention: Up to version 7.7.4 during index creation the corresponding table is locked for write transactions.

I/O Error: Verification of Checksum in Data Page



The screenshot displays two SAP SQL dialog windows. The top window, titled 'Check - Diagnosis Files - Database Messages', shows a list of error messages. The bottom window, titled 'SQL Dialog 1', shows the execution of a SQL query and its results.

Database Messages:

Time	Code	Msg Class	Msg Type	Text
2004-04-26 10:35:48	8865	ERR	4 Data	Checksum mismatch; calculated: 89809970 found: 8980000
2004-04-26 10:35:48	8865	ERR	12 IOMan	Bad page on Data volume 1 blockno 551
2004-04-26 10:35:50	8865	ERR	4 Data	Checksum mismatch; calculated: 89809970 found: 8980000
2004-04-26 10:35:50	8865	ERR	12 IOMan	Bad page on Data volume 1 blockno 551
2004-04-26 10:35:52	8865	ERR	4 Data	Checksum mismatch; calculated: 89809970 found: 8980000
2004-04-26 10:35:52	8865	ERR	12 IOMan	Bad page on Data volume 1 blockno 551
2004-04-26 10:35:54	8865	ERR	24 IOMan	Bad data page 60793
2004-04-26 10:35:54	8865	ERR	53000 B*TREE	0D0000000000000002A7000000000000
2004-04-26 10:35:54	8865	ERR	53000 B*TREE	TABLE ROOT 75569
2004-04-26 10:35:54	8865	ERR	53021 B*TREE	BAD FILE: 75569 (ROOT)
2004-04-26 10:35:54	8865	ERR	51080 SYSERROR	-9026 BD Bad datapage

SQL Dialog 1:

```
select tablename, indexname, type, root
from roots where root = 75569
```

TABLENAME	INDEXNAME	TYPE	ROOT
TEST	?	TABLE	75569

Rows in Result: 1

Auto Commit: On Internal Not Committed At least one error occurred

Statement success

----- Error -----
Auto Commit: On, SQL Mode: Internal, Isolation Level: Not Committed
General error:-9026 POS(1) System error: BD Bad datapage
check table test extended

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 98

In this case, too, a block was read whose checksum did not match with the calculated value. According to the ROOTS view, this tree belongs to the TEST table.

In such a case, check the table with the **CHECK TABLE EXTENDED** statement (default as of 7.6.01). With the option EXTENDED, the sequence of the primary keys is checked on all B* tree levels.

If CHECK TABLE does not return any errors, the table is intact. Note that in disk mirroring, depending on the disk used for the I/O, a correct block and then an incorrect block may be returned.

If CHECK TABLE continues to return the error, you have the following options:

- Restore the database
- Delete the tables and load the data from a sister system. This can lead to data loss. When tables are deleted, blocks that are no longer accessible remain occupied. In the ADMIN operational state, these blocks are transferred to free space administration with a **CHECK DATA WITH UPDATE**.
- Download the table without reading the records of the defective blocks, delete the table and load the downloaded records. The table data can be read in primary key sequence. The primary key values of the records in the defective block are not specified in the selection. This method is only possible if no index page of the B* tree has been affected. Data loss occurs.



Diagnosis of severe errors
(-10000 < Error number <= -9000)
Sometimes the database crashes.

- 1. After a crash, the start procedure copies the diagnosis files to a directory. In the standard, the parameter DiagnoseHistoryPath is set to <RUNDIRECTORY>/DIAGHISTORY. Only two versions of these files are kept (parameter DiagnoseHistoryCount)
- 2. After the crash:
 - Try to restart the database (with vtrace switched on)
- 3. Check, if the error can be reproduced (with vtrace switched on)
- 4. Inform the support group, if the cause could not be identified – open a customer message.

Diagnosis files only have to be explicitly saved if they are not automatically copied to the DIAGHISTORY.

Settings for the collection of historical crash information can be done via the parameters DiagnoseHistoryPath (DIAG_HISTORY_PATH) and DiagnoseHistoryCount (DIAG_HISTORY_NUM).

Message -602 in Syslog

Zeit	Typ	Nr	Man	Benutzer	Tcod	MNr	Text	Datum: 17.02.05
10:48:07	DIA	1	000	E30	SE38	BYL	Datenbankfehler -602 erfordert Intervention durch Datenbankadministrator	
10:48:07	DIA	1	000	E30	SE38	BYL	Datenbankfehler -602 beim SEL-Zugriff auf Tabelle ZZTELE_902 aufgetreten	
10:48:20	DIA	1	000	E30	SE38	AB0	Laufzeitfehler "DBIF_RSOL_SQL_ERROR" aufgetreten.	
10:48:20	DIA	1	000	E30	SE38	AB1	> Kurzdump "050217 104820 p34777 E30 " erstellt.	

How can database problems be analysed using db50?

- Example 1: -9026 Bad Data Page
- Example 2: -9028 Bad File

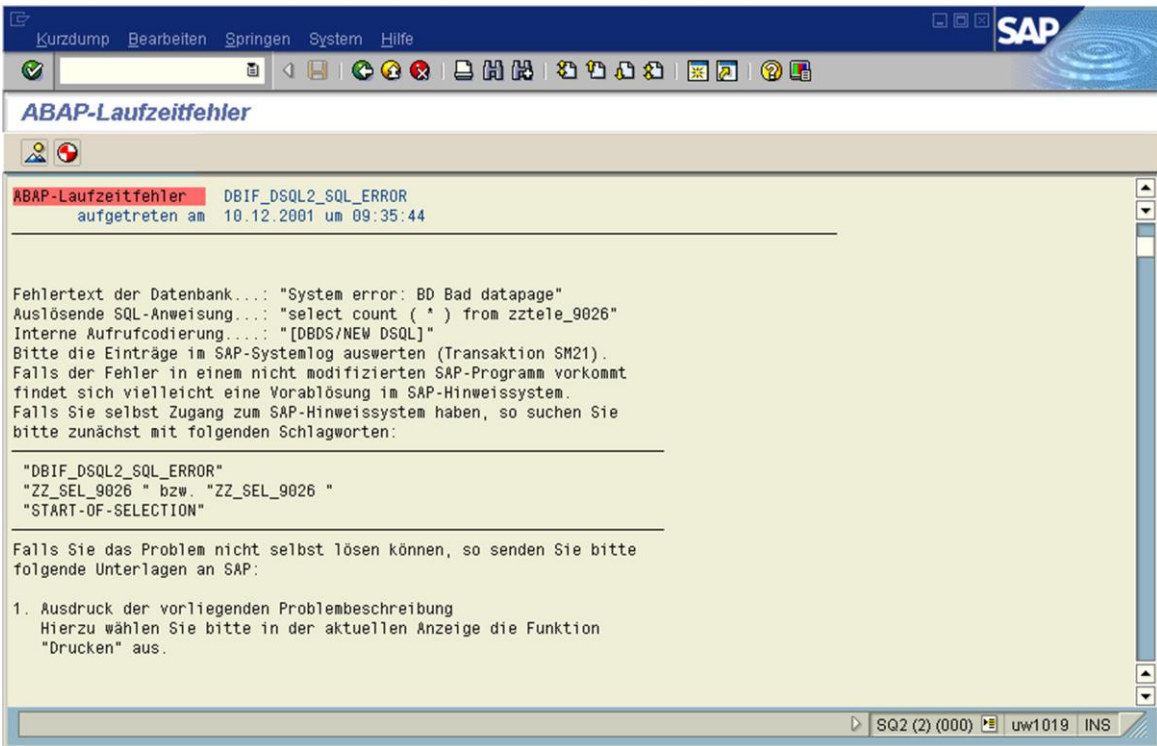
How does MaxDB support do an analysis if the database can no longer be transferred to the Online operational state (DB state Admin)?

- Analyze pages
- Restart record

MaxDB system errors are "mapped" to the general error -602 in the WebAS System. So this error number does not tell you much.

If the database is still in the ONLINE operational state or has restarted it following a crash, the analysis can be continued with transaction db50.

If restarting the database is no longer possible, other measures are required.



The screenshot shows the SAP ABAP runtime error message for error -9026. The title bar reads "ABAP-Laufzeitfehler". The error details are as follows:

ABAP-Laufzeitfehler DBIF_DSQ2_SQL_ERROR
aufgetreten am 10.12.2001 um 09:35:44

Fehlertext der Datenbank...: "System error: BD Bad datapage"
Auslösende SQL-Anweisung...: "select count (*) from zzteile_9026"
Interne Aufrufcodierung...: "[DBDS/NEW DSQL]"
Bitte die Einträge im SAP-Systemlog auswerten (Transaktion SM21).
Falls der Fehler in einem nicht modifizierten SAP-Programm vorkommt
findet sich vielleicht eine Vorablösung im SAP-Hinweissystem.
Falls Sie selbst Zugang zum SAP-Hinweissystem haben, so suchen Sie
bitte zunächst mit folgenden Schlagworten:

"DBIF_DSQ2_SQL_ERROR"
"ZZ_SEL_9026 " bzw. "ZZ_SEL_9026 "
"START-OF-SELECTION"

Falls Sie das Problem nicht selbst lösen können, so senden Sie bitte
folgende Unterlagen an SAP:

1. Ausdruck der vorliegenden Problembeschreibung
Hierzu wählen Sie bitte in der aktuellen Anzeige die Funktion
"Drucken" aus.

At the bottom right of the window, the status bar shows: SQ2 (2) (000) uw1019 INS

A short dump with error -602 'BD Bad Data Page' occurred during execution of the ABAP report ZZ_SEL_9026.

The short dump thus provides more detailed information about the error than the system log and returns the corresponding text from the database.

DB50: Error Texts and Codes



The screenshot shows the SAP DB50 interface. On the left, a tree view shows the 'Eigenschaften' (Properties) section for 'Alert-Monitor' and 'Aktueller Status'. Below this, the 'Fehlertext bestimmen' (Determine error text) dialog is open, showing 'Fehlernummer' (Error number) as '-602' and 'Fehlertext' (Error text) as 'System error'. On the right, the 'Tabelle: MESSAGES' (Table: MESSAGES) is displayed, showing a list of error messages with columns for 'MSGNO', 'LANGUAGE', and 'MSGTEXT'. The table contains the following data:

MSGNO	LANGUAGE	MSGTEXT
9.044	ENG	System error: BD Inconsistent nodetype
9.043	ENG	System error: BD Fair access not allowed
9.042	ENG	System error: BD Append not allowed
9.041	ENG	System error: BD Index not accessible
9.040	ENG	System error: BD Dropping of fair not allowed
9.033	ENG	System error: Bad converter bitmap page
9.032	ENG	System error: BD Bad converterpage
9.030	ENG	System error: BD Bad logpage
9.029	ENG	System error: BD Bad invfile
9.028	ENG	System error: BD Bad file
9.027	ENG	System error: BD Bad fair
9.026	ENG	System error: BD Bad datapage
9.024	ENG	System error: BD invalid entrypos
9.023	ENG	System error: BD illegal entrypos
9.020	ENG	System error: BD Init missing
9.019	ENG	System error: BD Buffer limit

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 102

In the initial menu for transaction db50, you can find the text that corresponds to a particular number via **Utilities -> Determine error text**. The text for error -602, however, is not terribly helpful as it is too general.

Utilities-> Error Codes provides information about (system) error numbers as well as their texts by displaying the Messages table.

In the present example, **KnImsg/knldiag** must be utilized for further analysis.

DB50: Problem Analysis - Messages



The screenshot shows the SAP DB50 Problem Analysis - Messages window. The title bar reads "Ausgabe einer Meldungsdatei". The window is divided into several sections:

- Left Panel:** A tree view showing the system hierarchy. The "Problemanalyse" folder is expanded, and "Meldungen" (Messages) is selected. Under "Meldungen", "Kern" (Kernel) is expanded, and "Aktuell" (Current) is selected.
- Top Panel:** Tabs for "Aktuelle Meldungen" (Current Messages), "Alte Meldungen" (Old Messages), and "Fehlermeldungen" (Error Messages). The "Aktuelle Meldungen" tab is active.
- Main Panel:** A table titled "Aktuelle Datenbankmeldungen (800 Kilobyte)". The table has columns for "Zeit" (Time), "PID", "Typ" (Type), "MsgID", "Label", and "Meldungstext" (Message Text). The messages are listed as follows:

Zeit	PID	Typ	MsgID	Label	Meldungstext
09:35:40	15	ERR	54001	I/O	page 00159A47010D0200...00000000020D0200
09:35:40	15	ERR	54001	I/O	BAD DATA PAGE 1415751
09:35:40	15	ERR	54001	I/O	on DEVNO 2 DEV_OFFSET 22177
09:35:40	15	ERR	53016	I/O	/sapdb/SQ2/sapdata/DISKD0002
09:35:42	15	ERR	54001	I/O	page 00159A47010D0200...00000000020D0200
09:35:42	15	ERR	54001	I/O	BAD DATA PAGE 1415751
09:35:42	15	ERR	54001	I/O	on DEVNO 2 DEV_OFFSET 22177
09:35:42	15	ERR	53016	I/O	/sapdb/SQ2/sapdata/DISKD0002
09:35:44	15	ERR	54001	I/O	page 00159A47010D0200...00000000020D0200
09:35:44	15	ERR	54001	I/O	BAD DATA PAGE 1415751
09:35:44	15	ERR	54001	I/O	on DEVNO 2 DEV_OFFSET 22177

The status bar at the bottom right shows "SQ2 (2) (000) uw1019 INS".

© SAP 2009 / MaxDB 7.7 Internals - Error Diagnosis/Page 103

To display the messages of the database system (*KnMsg/knldiag*), choose **Problem Analysis-> Messages**.

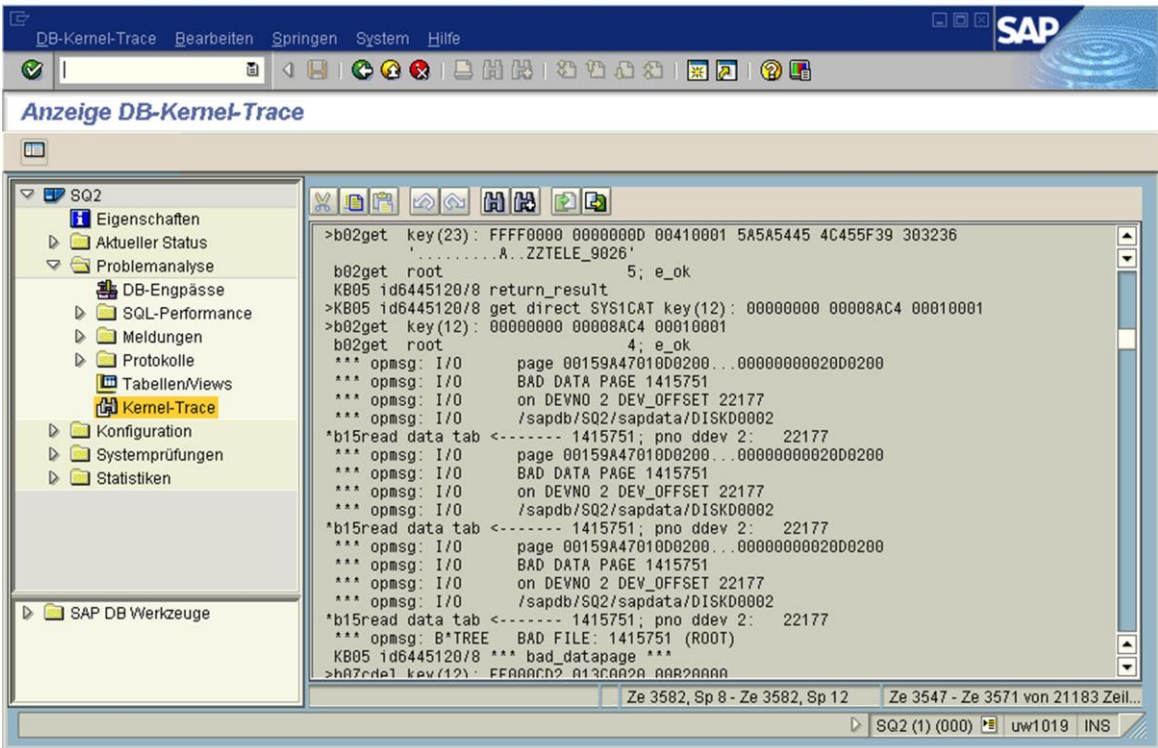
The error 'Bad Data Page' with error number -9026 is logged.

The root page of the affected object 1415751 is also recorded.

The position of the object is also logged. The defective object is located in Data Volume Number 2 at position 22177.

For a more precise analysis as to what is wrong with this object, the kernel trace (Vtrace) can be useful.

DB50: Problem Analysis – Kernel Trace



© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 104

Using db50, a Vtrace has been created.

The root page of the affected object 141575, the volume (DevNo 2) and the position in the volume (22177) are logged.

In addition, an important section of the affected page is logged in the Vtrace which allows you to identify the cause of error -9026.

Each page has one so-called header and one trailer entry, consisting of 8 bytes each. Both entries are checked when the page is accessed.

Header entry: Page: 00 15 9A 47 01 0D 02 00

Trailer entry: Page: 00 00 00 00 02 0D 02 00

If inconsistencies appear when the header and trailer are compared, the 'Bad data page' error is sent to the application.

In this example we see that the first 5 bytes in the trailer differ from the header.

If the affected object is a database table, the database must be restored.

Example -9026: Solution



- Check, if the affected object is a table.
- Examine volume 2, if there are hardware problems.
- Remove hardware problem.
- Restore of the database
- CHECK DATA with transaction db13

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 105

If the affected object is an index, error -9026 could be remedied by simply deleting and recreating the index; the cause of the problem, however, would not be solved.

The hardware must be examined in any event as such cases (-9026) can be due to hardware errors.

DB50: Problem Analysis -9028 BD BAD FILE



The screenshot shows the SAP ABAP Debugger interface. The title bar reads 'ABAP-Laufzeitfehler'. The main window displays the following error details:

ABAP-Laufzeitfehler DBIF_DSQ2_SQL_ERROR
aufgetreten am 07.12.2001 um 11:55:12

Fehlertext der Datenbank...: "System error: BD Bad file"
Auslösende SQL-Anweisung...: "insert INTO zzteile_bad values (? , ? , ? , ? , ? , ? , ? , ? , ?)"
Interne Aufrufcodierung...: "[DBDS/NEW DSQL]"

Bitte die Einträge im SAP-Systemlog auswerten (Transaktion SM21).
Falls der Fehler in einem nicht modifizierten SAP-Programm vorkommt
findet sich vielleicht eine Vorablösung im SAP-Hinweissystem.
Falls Sie selbst Zugang zum SAP-Hinweissystem haben, so suchen Sie
bitte zunächst mit folgenden Schlagworten:

"DBIF_DSQ2_SQL_ERROR"
"ZZ_INS_9028 " bzw. "ZZ_INS_9028 "
"START-OF-SELECTION"

Falls Sie das Problem nicht selbst lösen können, so senden Sie bitte
folgende Unterlagen an SAP:

1. Ausdruck der vorliegenden Problembeschreibung
Hierzu wählen Sie bitte in der aktuellen Anzeige die Funktion
"Drucken" aus.

At the bottom right of the debugger window, the status bar shows 'SQ2 (1) (000) uw1019 INS'.

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 106

Report zz_ins_9028 terminated with a short dump.

The cause of the error, -602 'BD Bad File', can be seen in the short dump.

Ausgabe einer Meldungsdatei

Kernel-Threads
I/O-Operationen
Kritische Abschnitte
SQL-Sperren
Speicherbereiche
Systemeinstellungen
Transaktionen
Problemanalyse
DB-Engpässe
SQL-Performance
Meldungen
Kern
Aktuell
Alt
Fehler
Database Manager
SAP DB Werkzeuge

Aktuelle Meldungen Alte Meldungen Fehlermeldungen

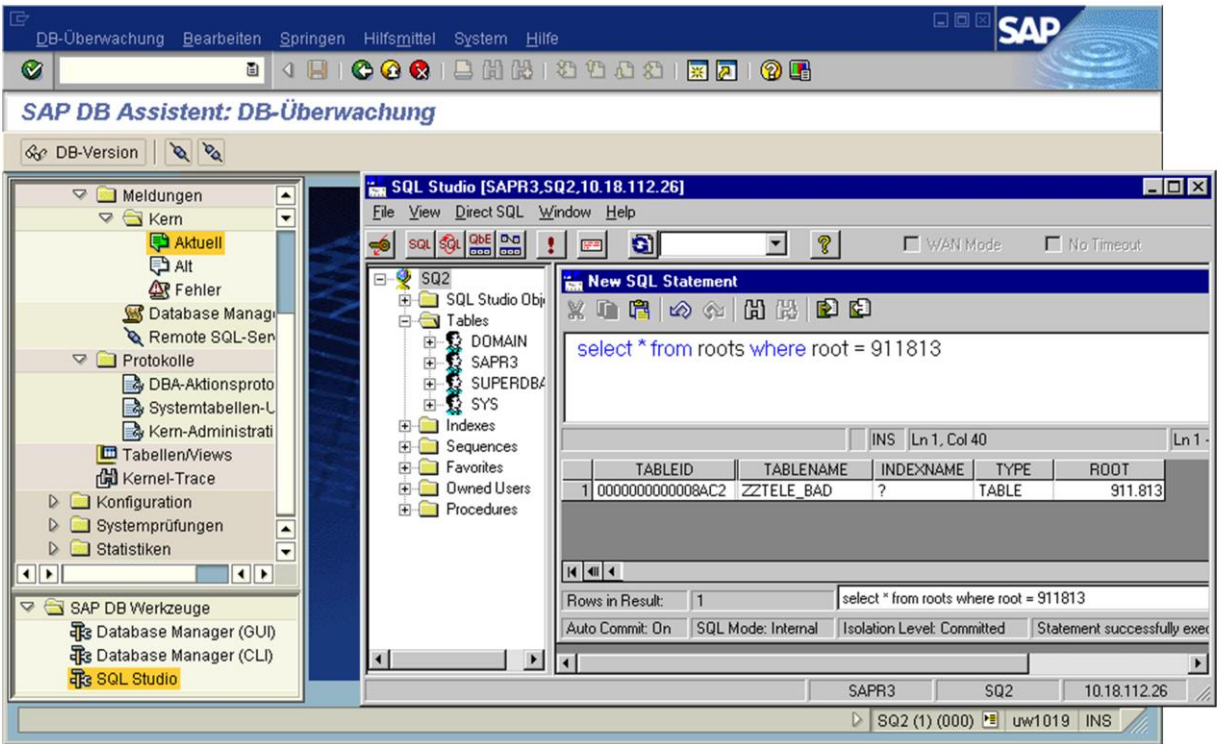
Aktuelle Datenbankmeldungen (800 Kilobyte)

Datum	Zeit	PID	Typ	MsgID	Label	Meldungstext
12-07	11:47:32	15		11007	COMMUNIC	wait for connection T44
12-07	11:50:37	15		12600	VERSION	*Kernel 7.2.5 Build 014-000-274-
12-07	11:50:37	15		12600	VERSION	*Kernel 7.2.5 Build 014-000-274-
12-07	11:50:40	15		12600	VERSION	*Kernel 7.2.5 Build 014-000-274-
12-07	11:55:11	15	ERR	53000	B*TREE	Bad file for Root: 911813
12-07	11:55:11	15	ERR	51080	SYSERROR	-9028 BD Bad file
12-07	11:55:34	15		12600	VERSION	*Kernel 7.2.5 Build 014-000-274-
12-07	11:55:34	15		12600	VERSION	*Kernel 7.2.5 Build 014-000-274-
12-07	12:00:36	15		12600	VERSION	*Kernel 7.2.5 Build 014-000-274-
12-07	12:00:36	15		12600	VERSION	*Kernel 7.2.5 Build 014-000-274-
12-07	12:00:36	15		12600	VERSION	*Kernel 7.2.5 Build 014-000-274-
12-07	12:05:34	15		12600	VERSION	*Kernel 7.2.5 Build 014-000-274-

SQ2 (1) (000) uw1019 INS

The current example has the error "-9028 Bad File": access to the table has been blocked because a serious error (e.g. -9026) occurred.

The root page number is recorded in ***KnIMsg/knldiag***. You can find out the table with the root page number.



© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 108

We have already seen how commands can be sent interactively to the database using the SQL Studio.

The SQL Studio can be started directly from transaction db50.

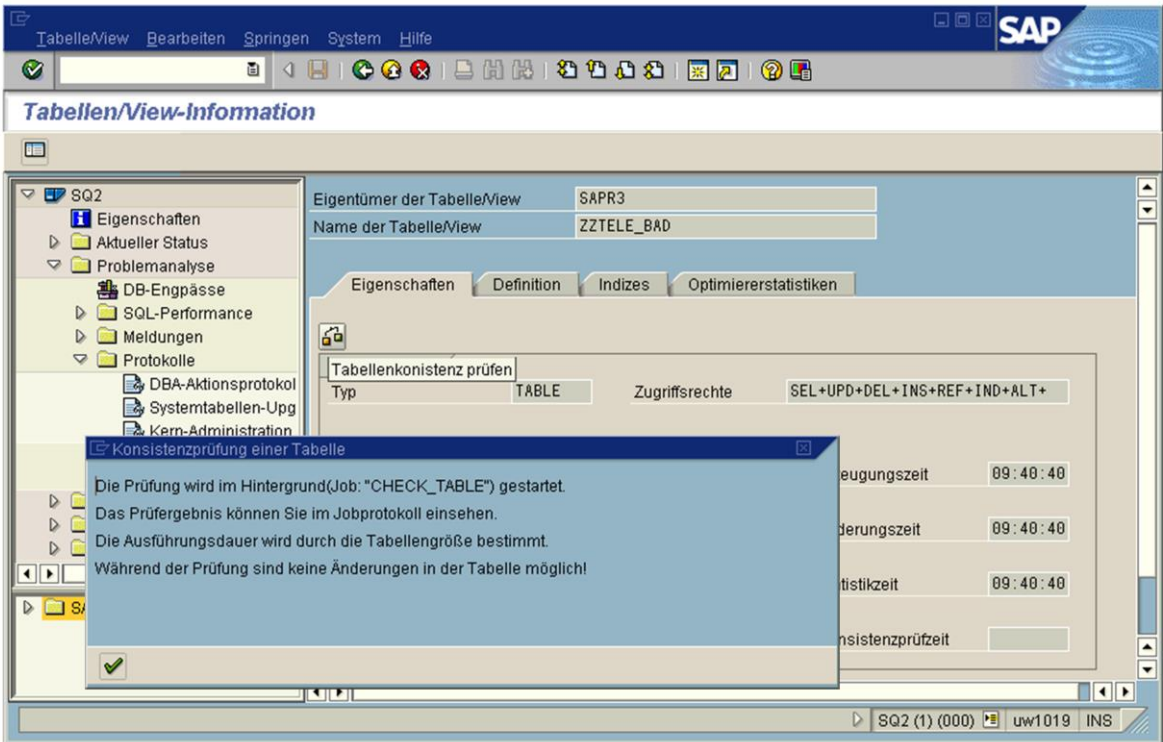
The Roots table contains the root page number, the object type and the name of the database object for all database objects.

Example – Field Types in the view roots



- SYS System table (not accessible)
- NAMED INDEX named index
- TABLE Table
- SHORT STRING FILE contains the short LOB COLUMNS
(exists for each table with LOB COLUMNS)
- LONG COLUMN contains the long LOBs (OWNER, TABLENAME and
INDEXNAME not specified)
- TEMP temporary table

DB50: CHECK TABLE



© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 110

In the present example, we know that table `zztele_bad` has caused a problem; a consistency check is triggered.

A Check Table is executed on the table.

Check Table checks the tree structure of the B* tree, header-trailer and so on.

If no inconsistencies are found, the BAD flag is retracted and access to the table is enabled.

This can happen if, for example, a Raid system reports an error but then corrects it immediately. Then the table is consistent, but has nevertheless been set to BAD.

SM37: Job Log Check_table



Job-Log Uebersicht für Job: CHECK_TABLE

Datum	Uhrzeit	Nachrichtentext	N-Klasse	N-Nummer	N
07.12.2001	13:05:00	Job wurde gestartet	00	516	
07.12.2001	13:05:01	Step 001 gestartet (Programm RSADACHT, Variante &0000000000001, Benutzername SQ2)	00	550	
07.12.2001	13:05:03	Tabellenprüfung für Tabelle "ZZTELE_BAD" erfolgreich beendet	SADA	115	
07.12.2001	13:05:04	Job wurde beendet	00	517	

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 111

The 'Check Table' executed without problems and reported no errors.

The program that had terminated with -9028 or -602 System Error can now be restarted.

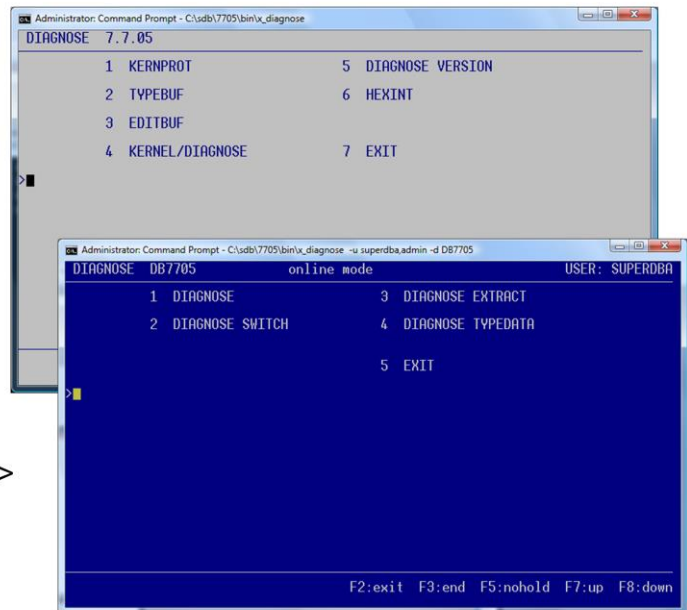
- Direct access to database pages (data, converter, log)
- Extraction of B* trees
- Analysis of knldump

■ Command:

<instroot>/bin/x_diagnose



< instroot>/bin/x_diagnose
-u <dbm-user>,<dbm-pw> -d <SID>



The tool **x_diagnose** allows you to access log pages and data pages in the database directly. With **x_diagnose**, you can can export configuration or restart information from the pages.

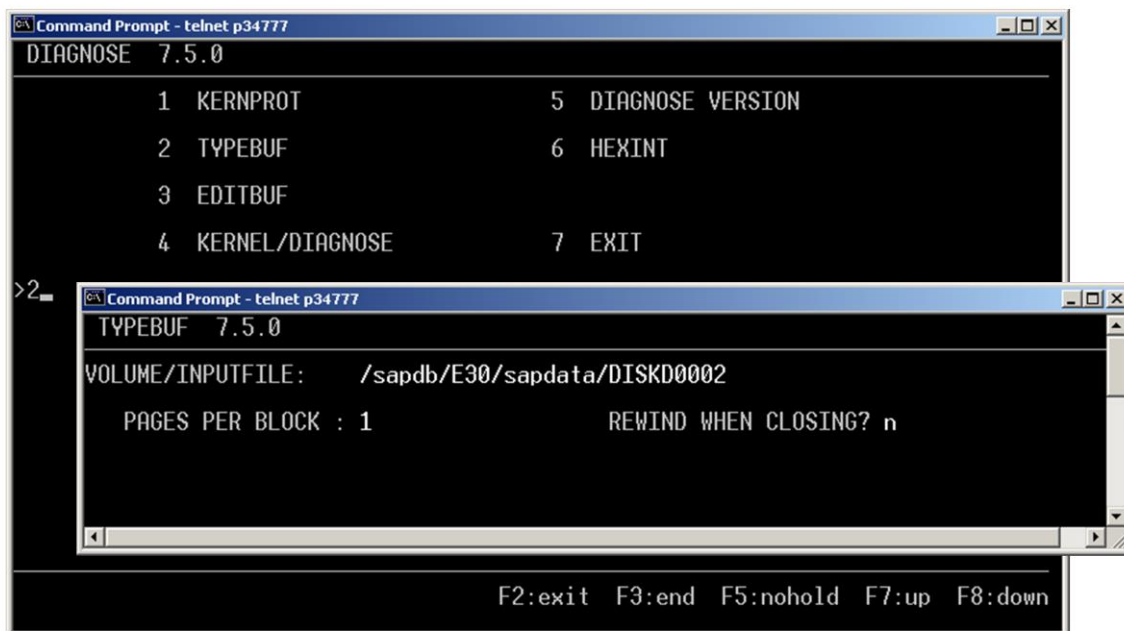
If necessary, you can extract an entire table tree.

x_diagnose is also used to evaluate **knldumps**. Cache contents, converter information, lock list entries, etc., can be analyzed at a later time.

Because improper use of the tool can be dangerous, **x_diagnose** should only be used by development.

In exceptional cases, pages can be repaired directly using an editing function.

Depending on the call of **x_diagnose** (with or without user/password combination for the database administrator) different menus were offered.



```
Command Prompt - telnet p34777
DIAGNOSE 7.5.0

1 KERNPROT          5 DIAGNOSE VERSION
2 TYPEBUF           6 HEXINT
3 EDITBUF           7 EXIT
4 KERNEL/DIAGNOSE

>2_

Command Prompt - telnet p34777
TYPEBUF 7.5.0

VOLUME/INPUTFILE:  /sapdb/E30/sapdata/DISKD0002
PAGES PER BLOCK :  1          REWIND WHEN CLOSING? n

F2:exit F3:end F5:nohold F7:up F8:down
```

The following pages show how to extract a data page with **Diagnose**.

First you choose TYPEBUF.

Then you enter the volume name.

The following slides show the x_diagnose tool of version 7.5; the menus are still identical in 7.7.



```
Command Prompt - telnet p34777
TYPEBUF 7.5.0                SELECT FUNCTION                /sapdb/E30/sapdata/D
1 ALL                        10 MINBUF
2 FROM/TO                    11 -
3 GET BLOCK                  12 BUFLNGTH
4 -                          13 LOWER/UPPER BOUND
5 SEARCH DUMP                14 SEARCH PAGE
6 SCAN                       15 SEARCH ON LOG VOLUME
7 NOSCAN                     16 HEXINT
8 -                          17 -
9 NODISPLAY                  18 NEXT INPUTFILE
                             19 RETURN
>6

F2:exit F3:end F5:nohold F7:up F8:down
```



```
Command Prompt - telnet p34777
TYPEBUF 7.5.0 /sapdb/E30/sapdata/D

node:
  F = node header
  K = key          S = key (short info: first/last key of node)
                  M = key (minbuf/maxbuf)
  R = record/object

log:
  H = logentry (header only)
  L = logentry
  G = logentry (minbuf/maxbuf)

dump:
  D = dump label
  F = full info

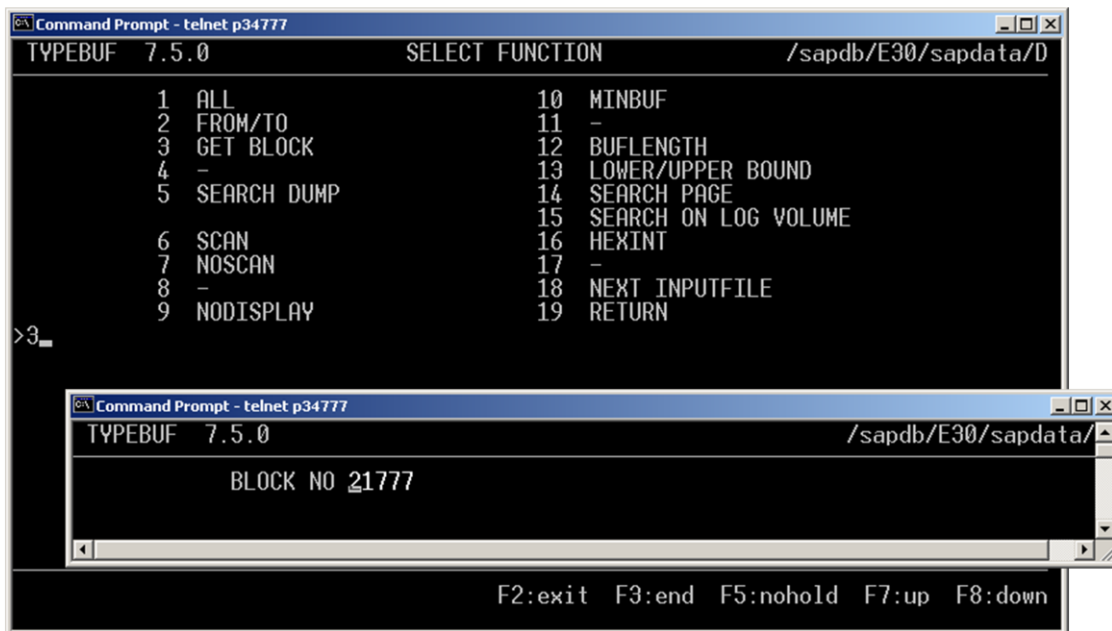
BD pages (converter):
  B = BD full info

other pages (restartrec, log info, hostfile info, etc.):
  F = full info

SELECT CHAR: f_

F2:exit F3:end F5:nohold F7:up F8:down
```

Using the SCAN menu, you can then specify what information you desire.



By specifying a block address - taken, for example, from *KnIMsg (knldiag)* - you come to the desired page.



```

Command Prompt - telnet p34777
TYPEBUF 7.5.0 /sapdb/E30/sapdata/D

LEAF 66043 perm      entries : 83      [block 21777]
      bottom : 7331  root   : 60723   convvers: 421
                        right  : 66044   writecnt: 1

00001      nodepage.pno: 66043      nodepage.pt : data
00006      nodepage.pt2: tab       nodepage.chk: ChecksumData
00008      nodepage.mde: empty
08181      nd_checksum : 131338522  nodepage2.pno: 66043
08189      nodepage2.pt : data      nodepage2.pt2: tab
08191      nodepage2.chk: ChecksumData
08192      nodepage2.mde: empty
00009      nd_bottom   : 7331      nd_rec_cnt  : 83
00017      nd_level    : 0
00019      nd_filestate: empty
00020      nd_sorted   : false      nd_root     : 60723/33ED0000
00025      nd_right    : 66044/FC010100 nd_left     : nil_pno
00033      nd_last     : nil_pno    nd_leaf_no  : nil_pno
00041      nd_conv_vers: 421        nd_str_vers : nil_pno
00049      nd_file_vers: dummy      nd_inv_usage: 0
00057      nd_leaf_cnt : 1          nd_treeleavs: nil

HOLDING      F1:hex/int F2:exit F3:end F5:nohold F7:up F8:down
    
```

Diagnosis: Restart Record (1)



```
Command Prompt - telnet p34777
DIAGNOSE  E30                online mode                USER: CONTROL
1  DIAGNOSE                    3  DIAGNOSE EXTRACT
2  DIAGNOSE SWITCH             4  DIAGNOSE TYPEDATA
                                   5  EXIT
>4_

Command Prompt - telnet p34777
DIAGNOSE  E30                SELECT FUNCTION        USER: CONTROL
1  GET DATA PERM              10  MINBUF
2  GET DATA STATIC            11  -
3  -                           12  BUFLLENGTH
4  GET LOG INFO                13  LOWER/UPPER BOUND
5  GET RESTART RECORD          14  -
                                   15  -
6  SCAN                        16  HEXINT
7  NOSCAN                      17  -
8  -                           18  -
9  NODISPLAY                   19  RETURN
>5_
```

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 118

You want to check the restart record.

You can access the restart record in various ways. One way is to choose KERNEL/DIAGNOSE, which brings you to the menus displayed here.

Restart Record (2)



```
Command Prompt - telnet p34777
DIAGNOSE E30 USER: CONTROL

RESTARTREC 21599 Savept: at 2004-09-21 09:38:45 0
00001 i4 1 restartr.pno: 21599 restartr.pt : data
00006 1 1 restartr.pt2: checkpt restartr.chk: ChecksumLogInfo
00008 1 restartr.mde: - - -
08181 i4i4 checksum : 372708 restart2.pno: 21599
08189 1 1 restart2.pt : data restart2.pt2: checkpt
08191 1 restart2.chk: ChecksumLogInfo
08192 1 restart2.mde: - - -
00017 bli1 rstIsConsist: false rstConfigPha: 0
00019 b1 rstSetEndRd0: false
00025 i4i4 rstConvVrs : 4952 rstPrevConvV: 4951
00033 i4i4 rstCurrBupVs: 4936 rstPrevBupVs: 4936
00049 i4i4 crSnapShotBA: 514 crConvRootBA: 39812609
00057 i4i4 crFdirRoot : 1 crLngDirRoot: 2
00065 i4i4 crMaxDynPno : 586214 crMaxStatPno: 1860
00073 b1 crRecovIncom: false
00089 crReleaseVrs: Kernel 7.3.0 Build 020-000-084-663
00201 i4i1 svpId : 568 svpReason : 0
00207 i2i4 svpOpenTrans: 0 svpIOseqNo : 2081936
00213 i4i2 svpStrtDevOf: 38682 svpStrtEntOf: 3152

HOLDING F1:NOHOLD F3:end F5:nohold
```

© SAP 2009 / MaxDB 7.7 Internals – Error Diagnosis/Page 119

The last Savepoint was written on 21.09.2004.

The database is in an inconsistent state (rstIsConsist: false).

Restart Record (3)



```
Command Prompt - telnet p34777
DIAGNOSE E30 USER: CONTROL
00219 i1  svpStrtEntTp: 30
00225 i4i4 svpOpnTrnsRo: 579679          svpHistoryRo: nil_pno
00233      svpDate      : 2004-09-21      svpTime       : 09:38:45
00241      svpEntryDate: 2004-09-20      svpEntryTime  : 07:05:31
00369 4    srgSession  : 226991
00377      srgSurrogate: 0-66C0
00385      srgSyskey   : 0-22B6C9
00393 6    srgTrans    : 1879216
00613 i4i4 rstLotLogSiz: 0                rstDatBupCnt: 16
00621 i4   rstLastDataB: 2044497
00625      rstReleaseVe: Kernel 7.5.0    Build 018-121-079-776
00665      rstDbIdent  : p34777:E30_20030612_182312

HOLDING F1:NOHOLD F3:end F5:nohold
```

```
Command Prompt - telnet p34777
DIAGNOSE E30 USER: CONTROL
LOGINFO 2 at 2004-09-22 11:00:14 MAX USED OFFSET 131070
FlushMode: MinimizeSpace DeviceState: Okay
00001 i4 1 id : 2 pagetype : LogInfoPage
00009 date : 2004-09-22 time : 11:00:14
00017 i1i1 flushmode : 0 devicestate : 0
00019 i2i4 queuecount : 1 cycleRelaOFF: 131070
00025 i4i4 lastknownSEQ: 2086118 lastknownOFF: 39307
00033 i4i4 oldNotSavOFF: 12820 oldNotSavSEQ: 1967890
00041 b1b1 devspaceEnab: true autoOverwrit: false
00045 i4 logBackupCnt: 13 redoUntilDat: 0000-00-00
00053 i4 redoUntilTim: 00:00:00 clearLowOFF : nil
00061 i4i4 clearUppOFF : nil seqAftRedoUn: nil
00069 i4 oldstKwnIOS: 1967890
00241 dbVersion : Kernel 7.5.0 Build 018-121-079-776
00281 dbident : p34777:E30_20030612_182312
00669 master node : P34777

HOLDING F1:NOHOLD F3:end F5:nohold
```

You access Loginfo Page via the same menu (GET LOG INFO).

The DBIdent, among other things, can be determined here.

Any questions?



THE BEST-RUN BUSINESSES RUN SAP™

