SAP® MaxDB™ Expert Session

SAP® MaxDB™: Kernel Parameter Handling

Christiane Hienger June 11, 2013

Public

SAP® MaxDB™ Expert Session

SAP® MaxDB™ Kernel Parameter Handling

Christiane Hienger
Heike Gursch
IMS MaxDB/liveCache Development Support
June 11, 2013

Agenda

- MaxDB Kernel Parameter File
- Tools to list/change Kernel Parameters
- Categories and Properties of Parameters
- Dependencies of Parameters (cserv.pcf)
- Parameter History
- Parameter Check

Nice to Know

- RunDirectoryPath and KernelDumpFileName
- Volume Parameters and Configuration Information
- MaxUserTasks, MaxCPUs and UseableCPUs

General Information on Parameters

MaxDB kernel parameters are used to configure a SAP MaxDB/liveCache database.

Parameters are available for

- Data and log volume configuration
- Caches and various memory structures
- Communication, I/O
- Process structure, CPU-Usage
- Log files and traces
- Optimizer
- ...

The Kernel Parameter File (1)

- Location: <PRIVATEDATAPATH>/config
- Name: <DBNAME>

e.g. -rw-r--r-- 1 sdb sdba 29669 12. Jun 15:50 WB5

- Format: Binary
- Tools: DBMCLI, Database Studio (DBMGUI for MaxDB Versions < 7.8 only)
- SAVE DATA and SAVE PAGES store the content of the current parameter file to backup media

The system stores the kernel parameters in a parameter file.

The system stores this parameter file in the file system in binary format in the directory <PRIVATEDATAPATH>/config.

The name of the parameter file corresponds to the database name.

You can only retrieve the parameter file with a database tool (DBMCLI or DatabaseStudio, DBMGUI (MaxDB Version < 7.8)).

When you carry out backups of the type SAVE DATA or SAVE PAGES, the system writes the content of the parameter file to the backup media.

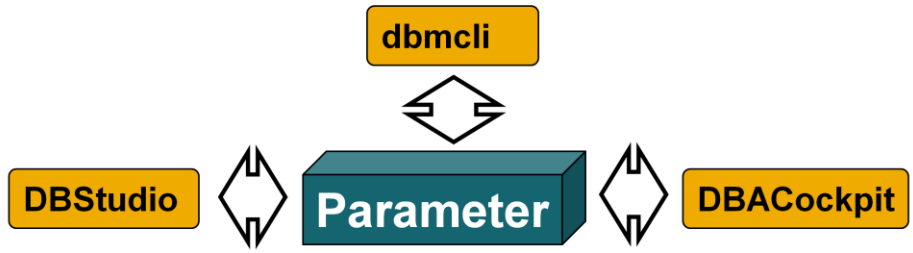
The Kernel Parameter File (2)

- Parameter History files (up to 10 versions)
  <PRIVATEDATAPATH>/config/<instance>.<number>

```
-r--r----- 1 sdb sdba  29669 15. Feb 15:44 WB5.10
-r--r----- 1 sdb sdba  29669 22. Feb 16:20 WB5.09
-r--r----- 1 sdb sdba  29669 22. Feb 16:25 WB5.08
-r--r----- 1 sdb sdba  29669 14. Mar 16:44 WB5.07
-r--r----- 1 sdb sdba  29669 25. Mar 10:11 WB5.06
-r--r----- 1 sdb sdba  29669 27. Mar 09:32 WB5.05
-r--r----- 1 sdb sdba  29669 12. Apr 15:00 WB5.04
-r--r----- 1 sdb sdba  29704 12. Apr 15:04 WB5.03
-r--r----- 1 sdb sdba  29704 12. Apr 15:51 WB5.02
-r--r----- 1 sdb sdba  29704 12. Apr 15:52 WB5.01

-rw-rw---- 1 sdb sdba  29866 12. Apr 15:52 WB5.pah
```

SAP MaxDB tools to display and change kernel parameters?



Example:

Call:	dbmcli –d MYDB –u control,pass
Display all:	param_directgetall
Display:	param_getvalue CacheMemorySize
Assign value:	param_put CacheMemorySize 100000
Calculate:	param_checkall

The parameter file is stored in the file system in binary format.

The Database Manager's client tools enable you to read and change parameters.

With dbmcli, parameters can be changed directly or in a parameter session.

Above you see a few examples of commands for making changes directly.

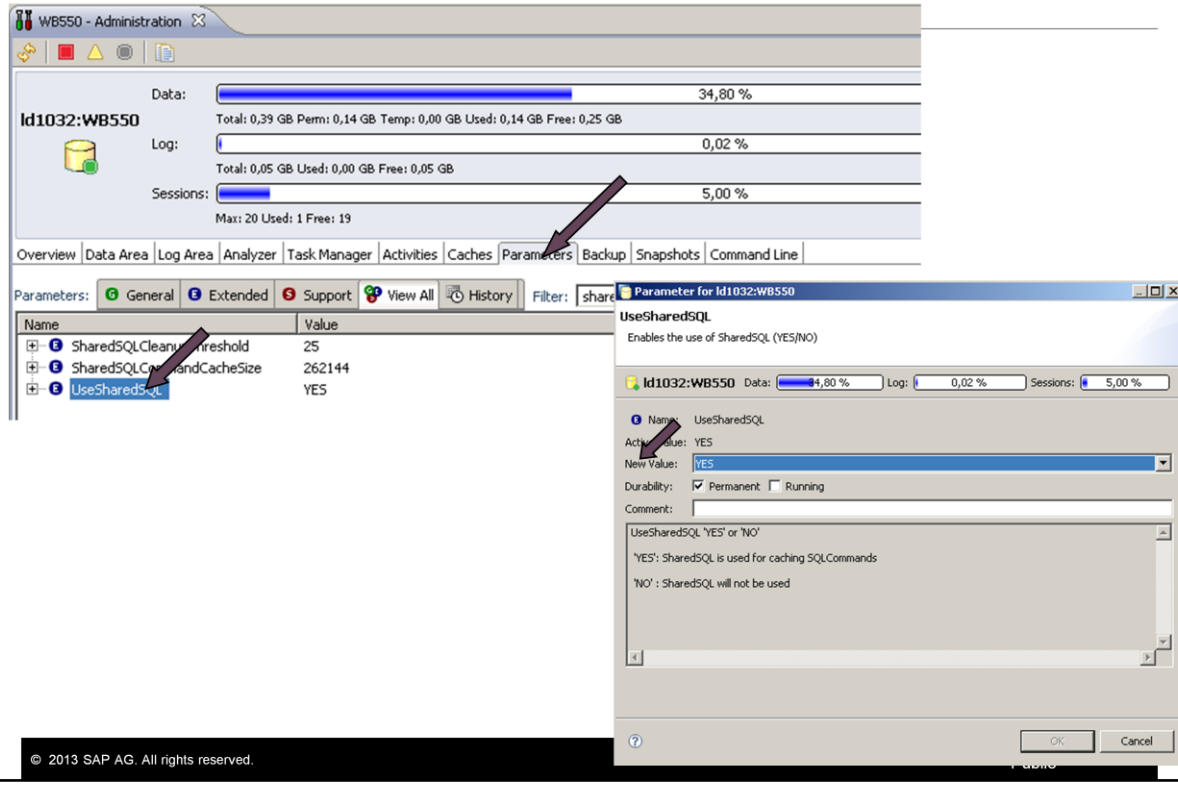
If parameters are changed within a session, a Commit makes all the changes valid while an Abort makes them all invalid.

Use the commands:

- param_startsession starts a parameter session
- param_commitsession ends the parameter session and saves the values
- param_abortsession ends the parameter session without saving it
- param_getvalue displays a parameter value
- param_put changes the value of a parameter
- param_restore activates old parameter version
- recover_config restores parameter settings from a data backup

The dbmcli command “help param” displays more commands.

Categories of Parameters



Kernel parameters are divided into three classes:

- **General**

  These parameters are set by database administrators.

- **Extended**
  These parameters are set in consultation with MaxDB Support or by implementing notes from the database administrator.

- **Support**

  These parameters are set by MaxDB Support or the developers.

Before a MaxDB version is delivered, it is programmed to calculate the optimal values for the respective operating system platform.

Note the following:

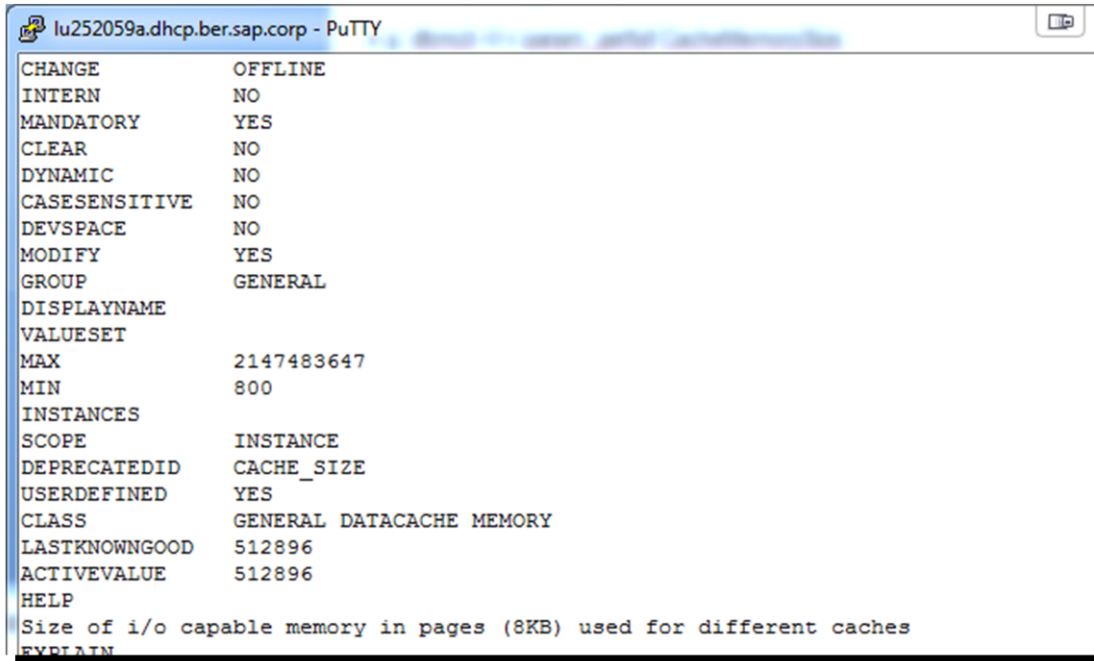
If you change extended parameters or support parameters, specify the note number or the customer message number in the comment field so that you can reproduce why a parameter was changed.

param_put [-running] [-permanent] <keyname> <value> [<comment>]

If the comment contains blank characters, set the comment in double quotes.

Properties of Parameters (1)

dbmcli –U c param_getfull <parameter name> e.g. CacheMemorySize



You can change kernel parameters in online mode of the database, when these parameter changes will get active is defined in the ‘properties’ of a parameter.

The properties of kernel parameters also determine, for example:

- whether you must assign a value to the parameter,
- whether a parameter has dependencies on other parameters, and so on.

You can use the dbmcli command 'param_getfull <parameter name>' to display the properties of a parameter.

CHANGE: If the property CHANGE has the value RUNNING, you can change the parameter while the system is running. The value OFFLINE means that the parameter change becomes active only after you restart the database offline.

INTERN: The value of this parameter is not contained in the parameter file (YES).

MANDATORY: The parameter value must be assigned (YES /NO)

CLEAR: During a database copy, the parameter is not copied (YES /NO)

DYNAMIC: Automatic numbering (for example, DATAVOL_?) YES /NO

CASESENSITIVE: upper/lower case (contents) YES | NO

DEVSPACE: Volume Parameter YES | NO

MODIFY: The parameter may be changed after generation of the instance.YES/NO

GROUP: Classification General, Extended, ...

DISPLAYNAME: Parameter name displayed in Database Studio

MAX: Maximum parameter value (numeric) <value>

MIN: Minimum parameter value (numeric) <value>

DEPRECATEDID: The parameter has an alias name that is not displayed in the MaxDB tools. After you change from a version lower than Version 7.7.03 to Version 7.7.03 or higher, the old parameter name is recorded here.

VALUESET: Permitted values

Properties of Parameters (2)

dbmcli –U c param_getfull <parameter name> e.g. CacheMemorySize

lu252059a.dhcp.ber.sap.corp - PuTTY

```
ACTIVEVALUE     512896
HELP
Size of i/o capable memory in pages (8KB) used for different caches
EXPLAIN

 The value specifies the i/o capable memory used by different
 MaxDB components in particular the data cache, converter and
 the shared catalog cache.

 Use the database monitoring to obtain information about the data cache
 hit rate and the performance of the system.

 The lower and upper limits are:
     800 <= CacheMemorySize < total RAM size

 (4 bytes integer)
```

Parameter Dependencies

File: <installationpath>/env/cserv.pcf

```
ID MaxCPUs
  TYPE int
  DEFAULT 1
  MANDATORY YES
  CLASS GENERAL TASKING
' SCOPE UNDECIDED
  DEPRECATEDID MAXCPU
  CODE
    CONSTRAINT  \
                1 MaxCPUs <=  \
                MaxUserTasks MaxCPUs >=  \
                AND
  ENDCODE
EXPLAIN
 The value of MaxCPUs has a great influence on the distribution of
 the database kernel tasks to operating system threads (UKTs).

 The parameter defines the maximum of CPU-Cores that are occupied by
 the operating system threads (UKTs) that are generating the main load.

 If the computer is used as database server exclusively, MaxCPUs should
 correspond to the actual number of CPU-Cores of the machine.
 Otherwise reduce the value by the number of CPU-Cores occupied by other applications.

 The lower and upper limits are:
     1 <= MaxCPUs <= MaxUserTasks
```


Calculation formulas, short texts and help texts for the parameters are found in the file <installationpath>/env/cserv.pcf

e.g. /sapdb/WB5/db/env

Here are the dependencies of the parameters defined too.

Formulas are defined as reverse polish notation in between CODE and ENDCODE

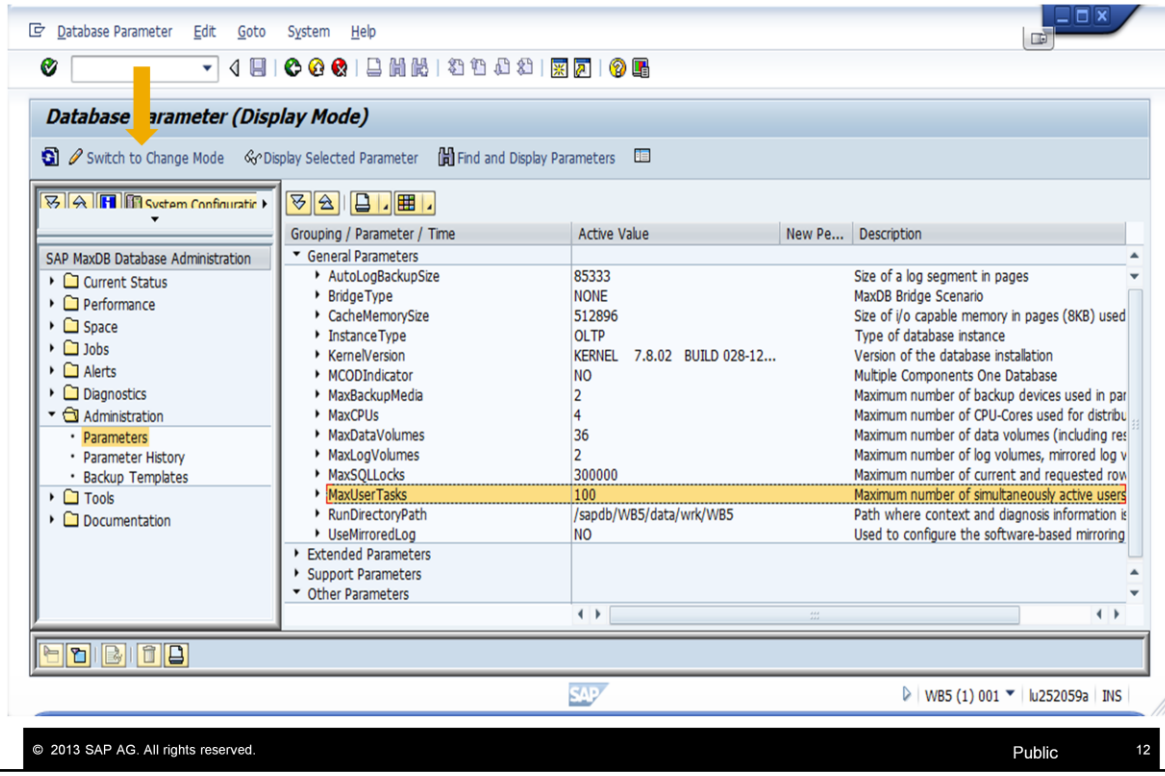
The cserv.pcf is created for each MaxDB version for each platform.

Note: It is not possible to use the same cserv.pcf file of one version for other platforms than the original one.

Note: The kernel executable and the parameter config file cserv.pcf have an internal version information which must fit to each other. It is not possible to start a MaxDB kernel successfully if the cserv.pcf file has a different version information.

Do not change the file cserv.pcf under any circumstances unless instructed to do so by SAP MaxDB Development Support.

Current Parameter Values in DBACOCKPIT



In the SAP system, you can display the parameters in transaction DB50/LC10 and DBACockpit .

The default view shows the GENERAL parameters only.

You can switch to the list of all parameters via GoTo -> Group view – shows the list of parameters grouped by the several parameter groups, General, Extended and Support parameters.

The Expert view shows the plain list of all parameters in alphabetical order.

You can generate a list of parameters and the related values via dbmcli in a *dbmcli session* with *param_directgetall* followed by *param_directgetallnext*.

You can list the parameter value of one special parameter with *dbmcli param_directget <parameter name>*

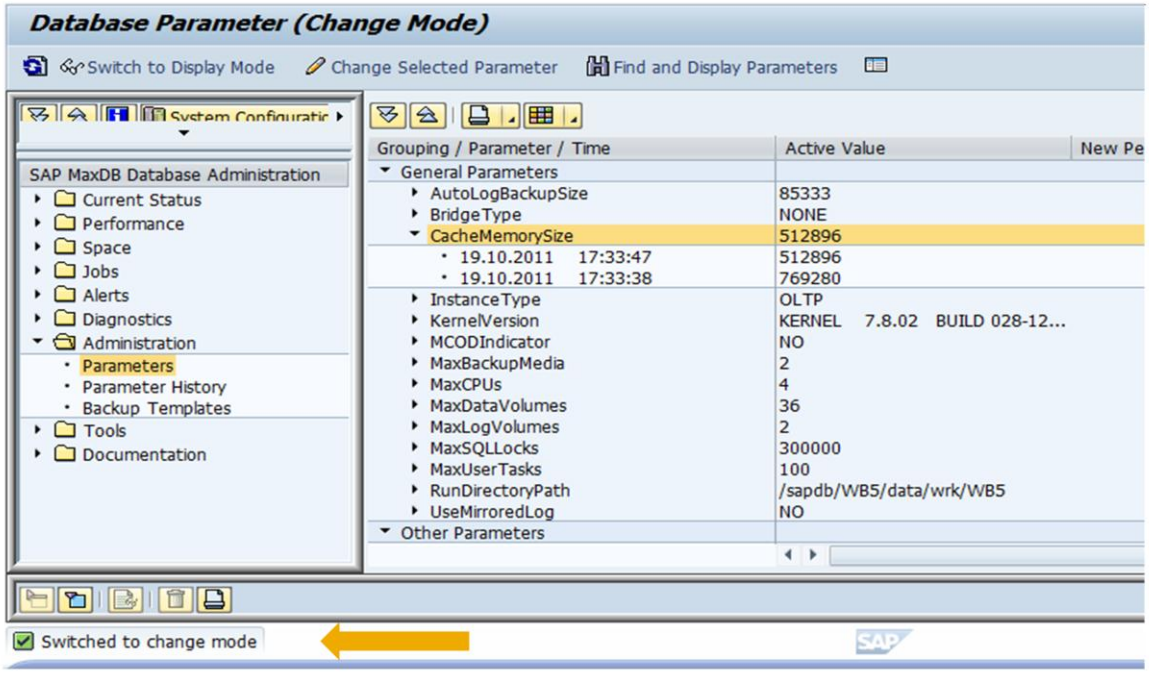
Note: the configuration parameters related to data and log configuration are not listed neither via DBAcockpit, nor via dbmcli or Database Studio.
To get information about the configured data and log volumes stored in the parameter file you use *param_getvolsall* followed by *param_getvolsallnext*

If you need to know detailed information about a special DATA or LOG volume you can use *param_getvolume <num> <mode>*
<num>: specifies the sequence number of the special volume in the total configuration

<mode>: specifies if it is a data (DATA) or a LOG (LOG) volume.

e.g. *dbmcli -U c param_getvolume 3 DATA*
OK
/sapdb/WB5/sapdata/DISKD0003
F
230400
With *switch to change mode* you can change parameter values.

Change Kernel Parameters (1)



In general, you can change status or counter values online. Some newly introduced parameters can be changed online although they influence the process or memory structure of the database.

In the SAP system, you can change the parameters in transaction DB50/LC10 and DBACockpit .

The default view shows the GENERAL parameters only.

You have to *switch to change mode* before you can change a parameter value.

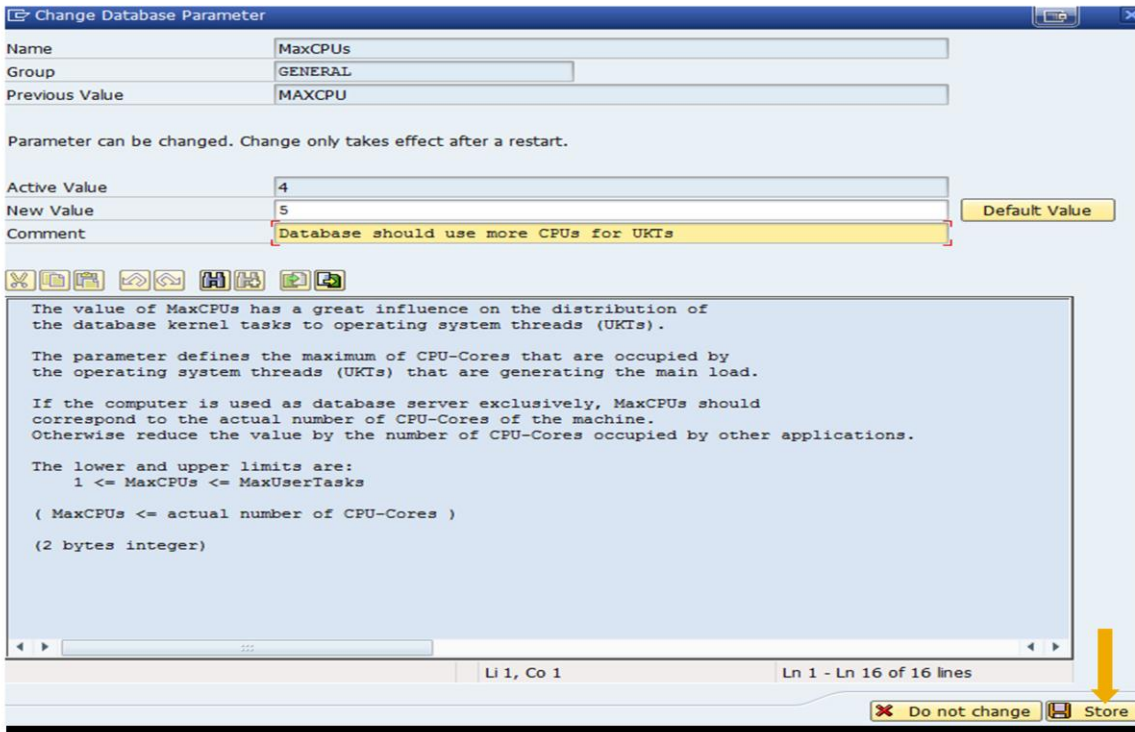
Mark the parameter name for which you want to change the value and double click.

Of course you can change parameter values via dbmcli and Database Studio as well.

Use *dbmcli param_put <parameter name> <parameter value>* to change parameter values.

Note: parameter values related to the volume configuration must not be changed via param_put commands!

Change Kernel Parameters (2)



This parameter MaxCPUs serves to inform the database kernel that multiple CPUs can be used.

At the same time, it allows the database system to restrict CPU usage. Such a restriction only applies to UKTs that contain user tasks.

Generally speaking, MaxCPUs indicates the number of CPUs simultaneously subject to intensive usage.

The value for MaxCPUs strongly influences the distribution of database kernel tasks to the operating system threads (parameter TaskCluster). If the computer is used exclusively as a database server, MaxCPUs should correspond to the number of CPU cores the computer has.

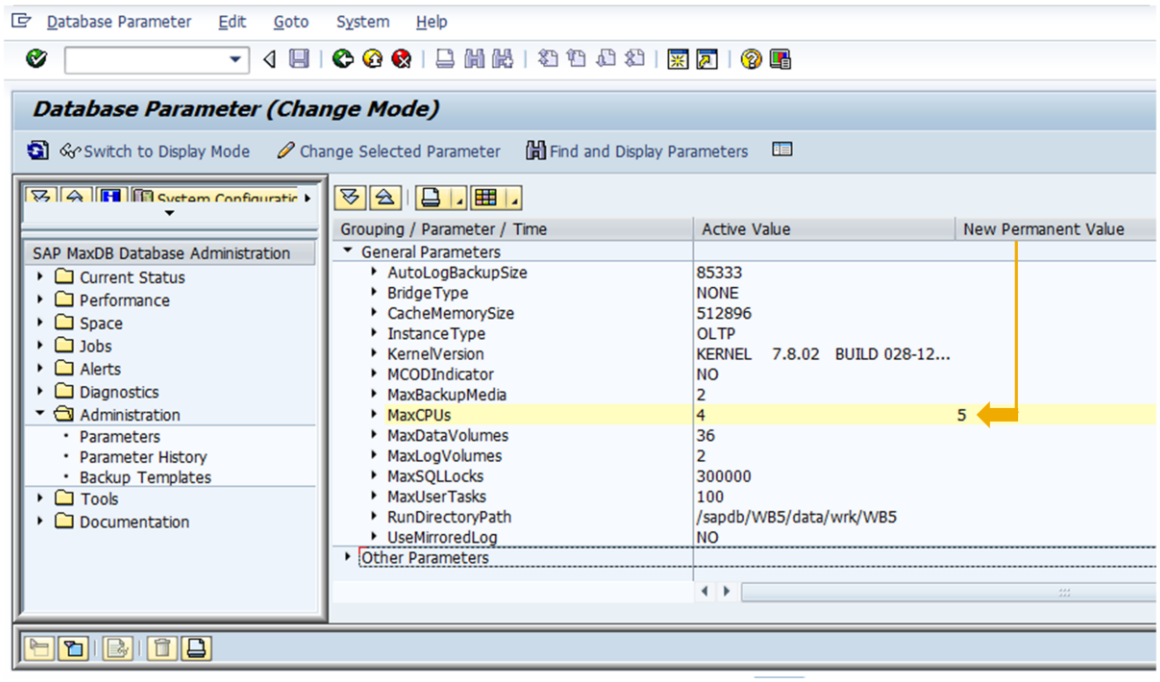
Active value in the screen shows the value of the parameter which is currently active. This value cannot be overwritten.

You can insert a new value and optional a comment which explains why this parameter has been changed. We recommend to use this functionality when a customer is changing a parameter. They should insert the CSS ticket number or the OSS note.

Store the parameter change.

The parameter MaxCPUs which is changed in this slide is not online changeable. This parameter change will get active after the next shutdown and restart of the database (db_offline & db_online).

Change Kernel Parameters (3)



The parameter change will be stored in the current parameter configuration file and in the parameter history file <DBNAME>.pah

-rw-rw---- 1 sdb sdba 1096 18. Jul 14:29 WB5.pah

-rw-r--r-- 1 sdb sdba 29669 18. Jul 14:29 WB5

After the restart was executed successfully with the current parameter file the parameter changes get active.

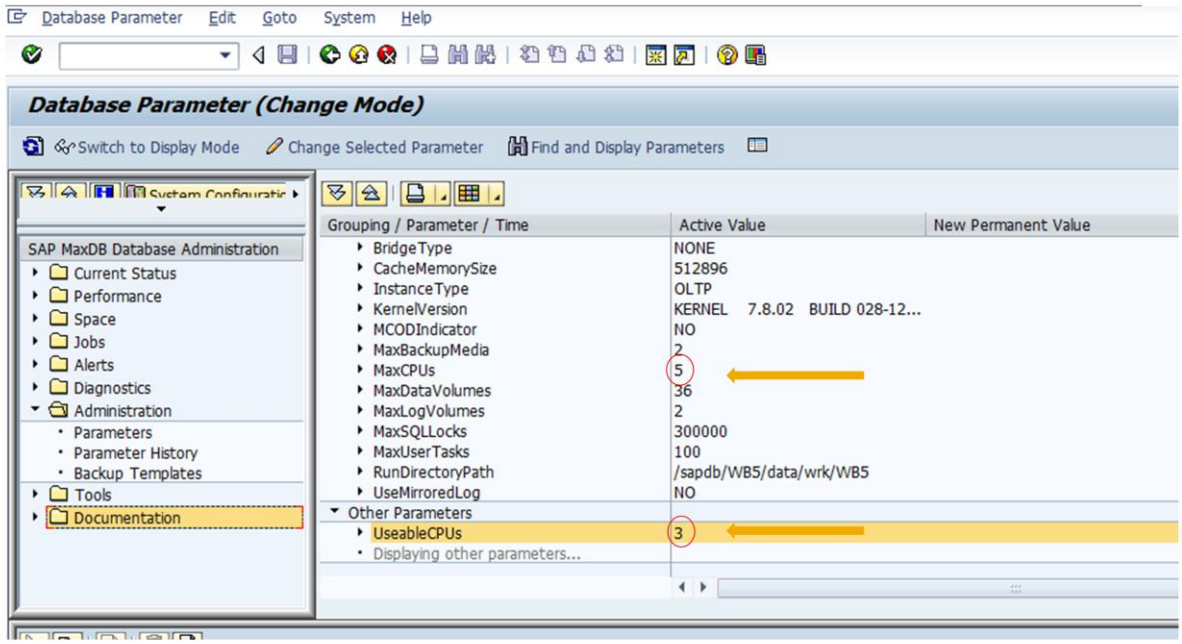
Additionally the database creates a new version of the current parameter file located in the same directory /sapdb/<SID>/data/config named <DBNAME>.01

Up to 10 parameter history files are created by default. The newest version has the smallest number (1), the oldest has the highest number (10).

Note: When the restart of the database fails because of a corrupted parameter file or caused by a mismatch of database volume configuration in the database and the volume configuration in the parameter file you can always try to get the system working again with one of the parameter history files.

How to do so will be explained in this training session as well later.

Change Kernel Parameters (4)



MaxDB development is pursuing the goal of making it possible to change most parameter values online.

Parameter MaxCPUs is one of the parameters which cannot be changed and get active without restart of the database. MaxCPUs defines the maximum number of CPU cores which can be used for user tasks.

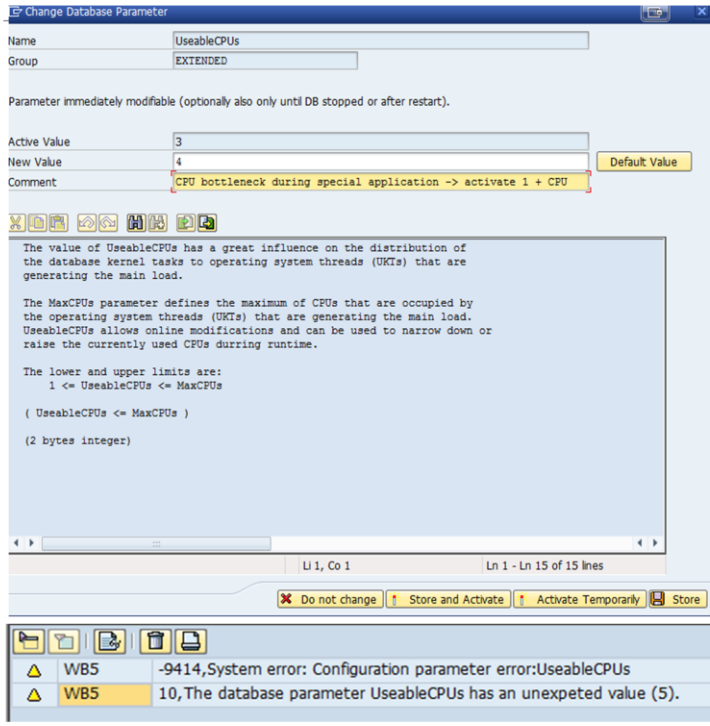
To be more flexible as of version 7.8 MaxDB can dynamically adjust the number of CPU cores to be used. The dispatcher moves user tasks out of the inactive user kernel threads when the tasks become active. This can be configured with the kernel parameter UseableCPUs.

This example shows the change of the online changeable parameter UseableCPUs.

The current value of UseableCPUs is 3. The maximum number of useable CPUs is restricted by the parameter value of MaxCPUs -> 5

Online changeable parameters can get active as soon as the parameter change is done.

Change Kernel Parameters (5)



The parameter change can be *activated temporarily* – this means the parameter change gets immediately active but the change is not stored in the parameter file. This option can be used for online changeable optimizer parameters, if you want to test if a strategy is changing caused by the parameter change.

Store and activate will change the parameter file and the change is getting active immediately. This should always be used only with confirmation of the customer.

Save will store the new value in the parameter file only, the change is getting active after the next shutdown/restart.

If you change any of the parameters the parameter change will be checked implicitly.

e.g. You set a value for UseableCPUs > MaxCPUs you'll get an error message.

Note: dbmcli does not send an error message but sets a valid value implicitly.

The parameter change will be stored in the current parameter configuration file.

The original parameter configuration file which was used for the last successful restart is copied to <DBNAME>.pah

This parameter change will get active after the next shutdown and restart of the database (db_offline & db_online)

Current Parameter configuration (1)

The current setting of the parameters is shown by the view ACTIVECONFIGURATION (online mode only)

PARAMETERNAME	PERMANEN...	CHANGEABLE	DEPRECATED	VALUE
EnableBTreeRootLockOptimization	YES	YES	NO	NO
EnableClientSpecificOmsSchema	YES	NO	NO	YES
EnableCommandMonitor	YES	NO	NO	YES
EnableDataIOCluster	YES	YES	NO	YES
EnableDataVolumeBalancing	YES	YES	NO	YES
EnableDelayedRootPageCreation	YES	YES	NO	YES
EnableExternalDumpRequest	YES	NO	NO	NO
EnableFetchReverseOptimization	YES	YES	NO	YES
EnableFileCounterInitialization	YES	YES	NO	YES
EnableFirstRowAccessOptimization	YES	YES	NO	YES
EnableGenericSecurityService	YES	NO	NO	NO
EnableIOTimeStatistic	YES	YES	NO	YES
EnableImplicitPrepareStatement	YES	YES	NO	YES
EnableIndexOnlyStrategy	YES	YES	NO	YES
EnableJoinHashTableOptimization	YES	YES	NO	YES
EnableJoinIntermediateSort	YES	YES	NO	YES

As of version 7.7.03 the parameter names were consolidated. Therewith most parameters got a new name without containing underlines. The legibility of parameter names is improved by the use of upper and lower case characters.

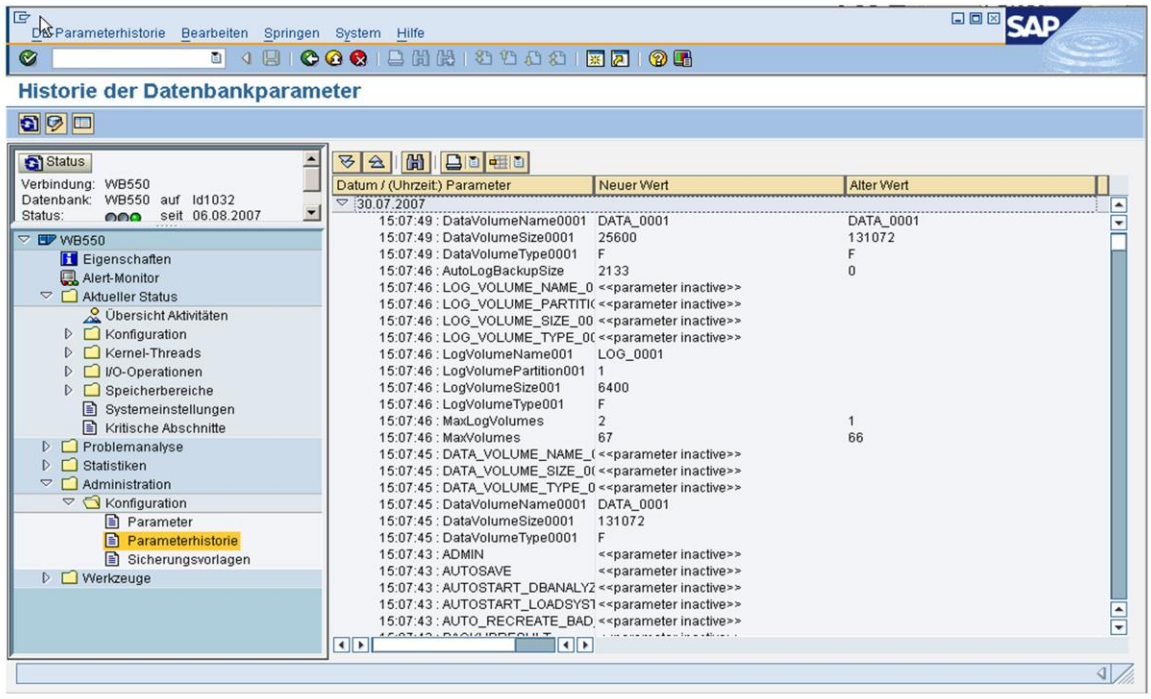
You can read and set the parameters by using the old names. The command *param_directgetall* only shows the new parameter names. The view ACTIVECONFIGURATION shows old and new parameter names.

Current Parameter configuration (2)

- The current configuration in offline mode is stored in parameter file <DBNAME> and can be listed via dbcmli param_directgetall / param_directgetallnext
- KnlMsg/KnlMsg.old shows the configuration of the database during restart

2012-07-18	15:32:21	0x613E		15	RunTime	EnableExternalDumpRequest=NO
2012-07-18	15:32:21	0x613E		15	RunTime	EnableFetchReverseOptimization=YES
2012-07-18	15:32:21	0x613E		15	RunTime	EnableFileCounterInitialization=YES
2012-07-18	15:32:21	0x613E		15	RunTime	EnableFirstRowAccessOptimization=YES
2012-07-18	15:32:21	0x613E		15	RunTime	EnableGenericSecurityService=NO
2012-07-18	15:32:21	0x613E		15	RunTime	EnableIndexOnlyStrategy=YES

DBACockpit: Configuration and History of Parameters



In transaction DB50 and DBACockpit you can view the list of all parameter changes, sorted according to the change date.

The system displays a list of the database parameters changed at this point in time, and their previous and new values.

Parameters that are no longer used by the database as of a particular date are assigned `<<parameter inactive>>` as a new value.

Parameter Check with Database Analyzer: Prerequisites

- SAP Note 1111426 Parameter check for liveCache/MaxDB instances
  - Attachments
    - DbanalyzerParamCheck.SAR
- Download the attachment into a temporary directory, e.g. /tmp
- Unpack DbanalyzerParamCheck.SAR
  - sapcar –xvf DbanalyzerParamCheck.SAR

dbanalyzer_InstanceParameterCheck.cfg

As bad performance could be caused by wrong parameter settings, you should check the database configuration first.

SAP MaxDB offers a check tool for MaxDB kernel parameter settings. This check is embedded into the *Database Analyzer*. The parameter check tool is used to check whether the configuration of your liveCache, MaxDB, OneDB or BW system corresponds to the current SAP recommendations.
In general the parameter recommendations which are described in the MaxDB parameter notes (MaxDB Version 7.7: 1004886, MaxDB Version 7.8: 1308217) are checked.

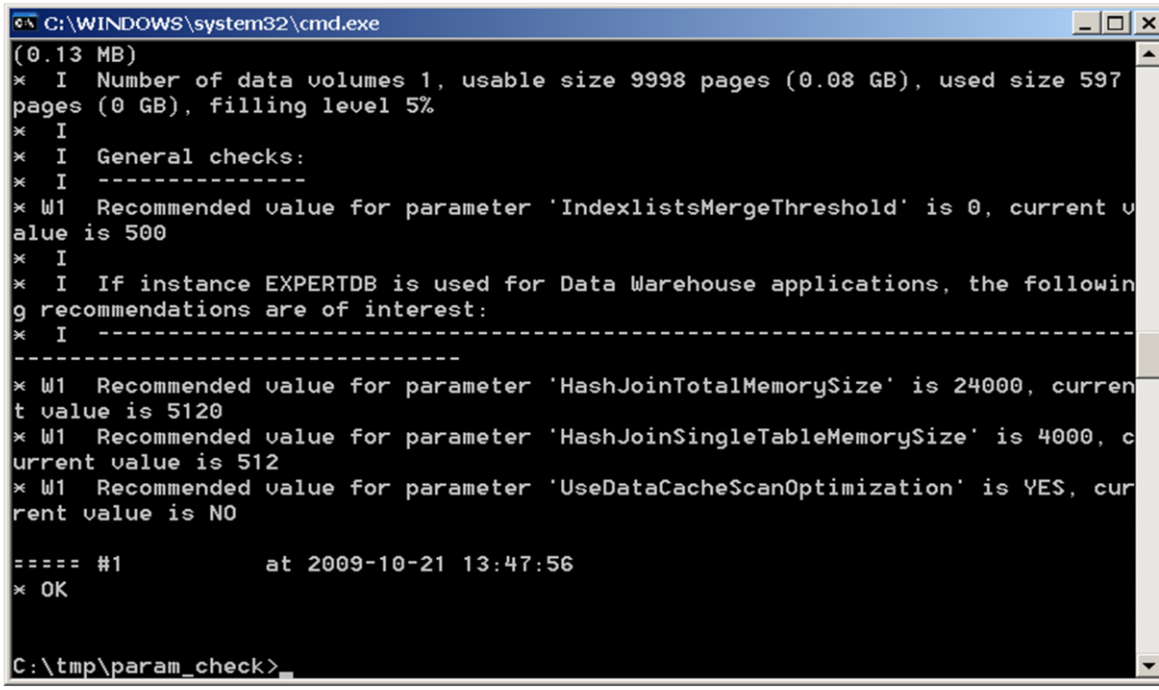
The parameter check should be executed after each upgrade to a new liveCache/MaxDB version. Different recommendations may be relevant for different database versions.

The parameter check tool uses a special *Database Analyzer* configuration file. This special configuration file is attached to note 1111426. As this file is regularly updated, you must download it again before each check. This file can be stored in a temporary directory – e.g. /tmp

Use *sapcar –xvf DbanalyzerParamCheck.sar* to extract the configuration file *dbanalyzer_instanceParameterCheck.cfg*

Do not replace the original database analyzer config file with the new one!

Parameter Check with Database Analyzer



The database instance must be in operational state ONLINE when you start the parameter check tool. Perform the automatic check as SYSDBA user (e.g. dbadmin)
dbanalyzer –d EXPERTDB –u dbadmin,secret –f c:\tmp\dbanalyzer_instanceParametercheck.cfg -o c:\tmp\param_check -i –c 1 –t 1,1 –n <server>
-i the output directory will be cleaned up
-c output will be send to screen as well
-t only 1 snapshot in an interval of one second

Analyze the screen output or the file */tmp/param_check/<YYYYMMDD>/DBAN.prt*.
Important are all messages that are marked with "* W1 to * W3"
The following checks are executed:
- general parameters
- parameters which influence the I/O performance
- optimizer parameters
- special liveCache parameters
- additional checks
  - do corrupt indexes exist?
  - is the database kernel trace activated?
  - do tables exist which do not have any file directory counters?
  - is logging activated and autooverwrite deactivated?
  - does the size of the IO Buffer Cache correspond to the SAP recommendation, which is 2% of the configured volume size for UNICODE systems and 1% for NON-UNICODE systems?

RunDirectoryPath, KernelDumpFile, Diaghistory

```
dbmcli -U c param_directget RundirectoryPath
RundirectoryPath          /sapdb/WB9/data/wrk/WB9
```

- ⇒ path where diagnosis information is stored
- ⇒ Kernel message files (KnlMsg, KnlMsg.old, KnlMsgArchive)
- ⇒ dbmServer log files (dbm.*)
- ⇒ Subdirectory DIAGHISTORY
- ⇒ Kernel dump File (knldump) – Parameter KernelDumpFileName

```
dbmcli -U c param_directget KernelDumpFileName
KerneldumpfileName        knldump
```


The rundirectory is the most important directory used in error analysis. Here all important log files are located.

The log files written by the database kernel (Knl*) are pseudo HTML and have to be converted first before the content can be analyzed.

The subdirectory DIAGHISTORY contains copies of the log files which are stored, after a crash, during restart of the database. important log files for error analysis will not be overwritten.

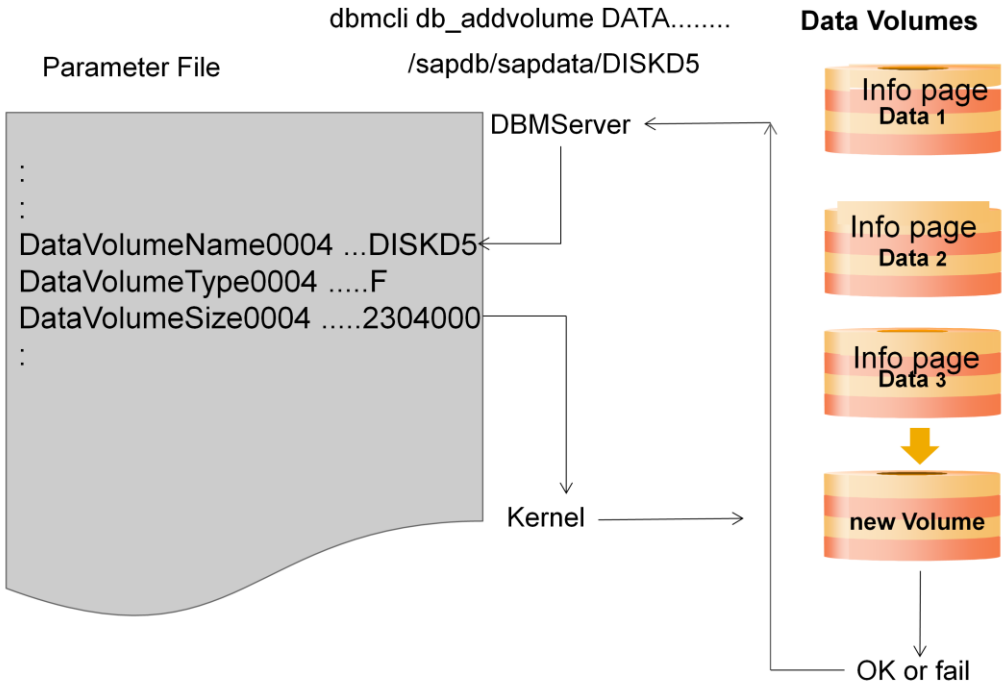
The knldump file is written when the database crashes. The size of the knldump file depends on the allocated memory size by the kernel (caches + heap).

It can be very large. By default the knldump is written into the Rundirectory.

Therefore it is recommended

- > either to check the capacity of the Rundirectory to ensure that the complete KernelDump can be stored without getting file system full situation.
- > or via the kernel parameter KernelDumpFileName to specify different path with enough capacity.

Where is the Configuration of Data Volumes stored?



Add new data volume:

1. DBMserver inserts the volume related parameter information into the parameter file Volume Name, Type and Size

2. ADD volume command is sent to the database kernel – the new volume gets a sequence number – in the example the number 4

3. IO Info page is changed (on each volume)

4. After the kernel configuration has been stored on each volume the DBMserver gets the infomation that the add volume has been finished successfully.

Note: These 4 activites are not done in a transaction. If an error happens on OS level between ADD volume executed by the kernel and the OK to the dbmserver then you have a mismatch between parameter file and kernel configuration.
The parameter file has a volume configuration which does not match the configuration stored on INFO pages in the database.

In such cases you have to use the history files of the parameter file and use param_restore.

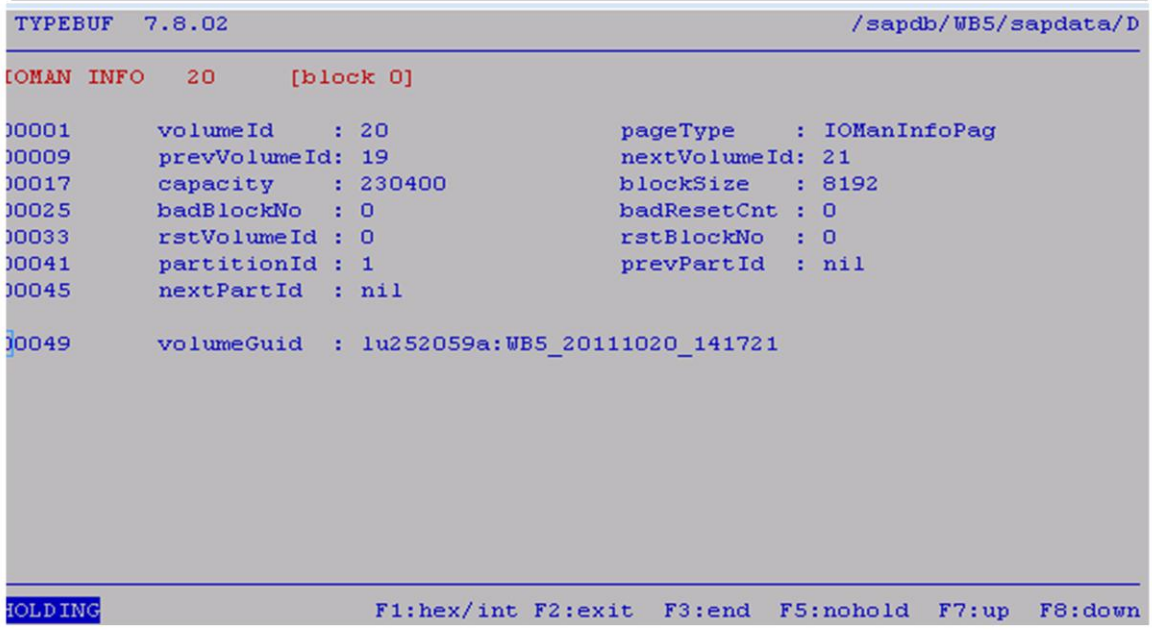
1. Copy the current parameter file to <DBNAME>.ori
2. dbmcli param_restore <no> normally you use param_restore 1 which is the most current version file before the ADD volume was executed.
3. Restart the database

The same happens if the customer drops a volume out of the parameter file only.

Note: If you delete a data volume on disk area without DBMServer commands you cannot solve this issue with a param_restore. The data of the deleted volume have not been distributed to other data volumes. Data is lost!

Always use dbm command db_deletevolume.

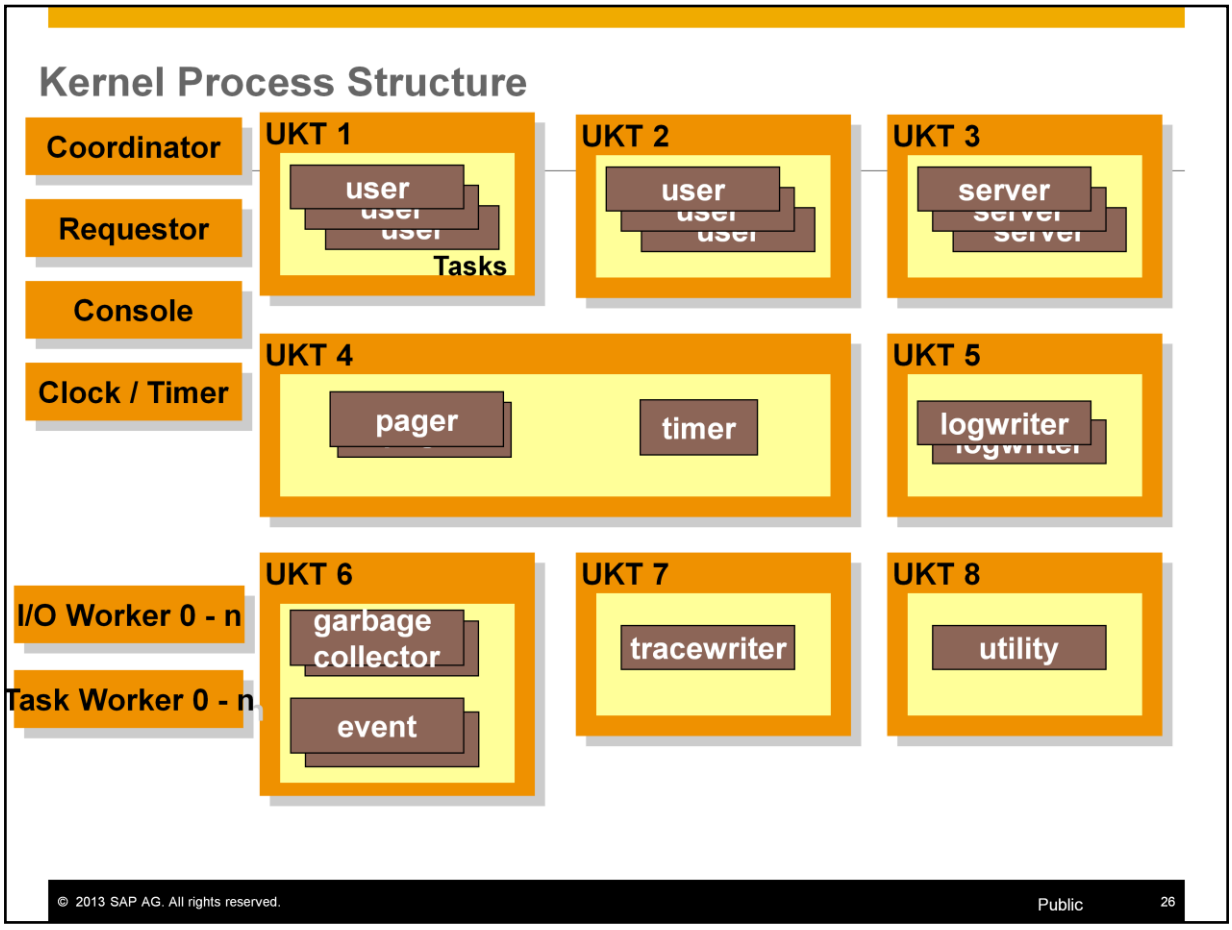
How to check the current configuration with x_diagnose (Development Support only)



/sapdb/<DBNAME>/db/bin/x_diagnose

e.g. /sapdb/WB5/sapdata/DISKD0020

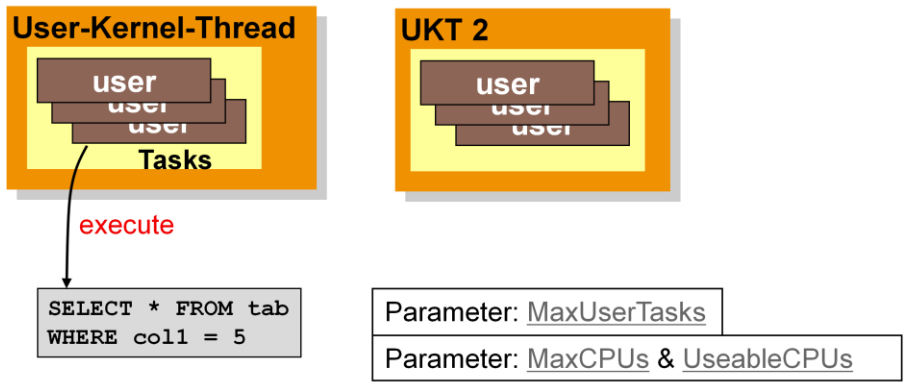
Prot file: e.g dev.prt
Input: Path of the volume which has to be checked (last one)
2 Typebuf
6 scan KF
12 Buflength
Page 0 -> IO Info page



This slide shows the process structure of the MaxDB kernel.

This process structure is configured via parameters MaxUserTasks, MaxCPUs and UseableCPUs, which implicitly influence the parameter values of parameters TaskCluster01, TaskCluster02, TaskCluster03.

MaxUserTasks, MaxCPUs, UseableCPUs



Each user session is assigned exactly one **user task** at logon.
The maximum number of available user tasks is determined by the database parameter **MaxUserTasks**. This parameter also restricts the number of user sessions that can be logged on to the database system simultaneously.
The database parameter **MaxTaskStackSize** determines the memory usage of the user tasks.

The general database parameter **MaxCPUs** specifies the number of user kernel threads among which the user tasks are distributed. Other tasks and global threads use very little CPU time. The parameter **MaxCPUs** allows you to specify how many processors the database should use in parallel.

The parameter **UseableCPUs** allows an online adjustment of the number of used user kernel threads. This makes dynamic configuration changes according to the available CPUs in the system possible.

As of version 7.4.03, user tasks can switch from one UKT to another if the previously-responsible UTK is overburdened. This results in better scaling for multiprocessor servers (SMP). To use this function, set the parameter **LoadBalancingCheckInterval** to a value greater than 0.

As of version 7.8 Load Balancing is released for MaxDB and liveCache instances and used by default. The scheduler immediately moves the task to an idle user kernel thread if the current thread is overloaded.

More Information about Parameter

FAQ: SAP MAxDB Database Parameter

https://service.sap.com/sap/support/notes/1139904

Documentation:

Parameter:
http://help.sap.com/saphelp_nw73/helpdata/en/0c/581afcc31c45158d8cf2e19617aea1/frameset.htm

Parameter file:
http://help.sap.com/saphelp_nw73/helpdata/en/44/c37590865960efe10000000a155369/frameset.htm

SAP MaxDB Parameter Notes

Parameter Notes MaxDB

SAP MaxDB Version 7.9 – SAP note: 1346964

SAP MaxDB Version 7.8 – SAP note: 1308217

SAP MaxDB Version 7.7 – SAP note: 1004886

SAP MaxDB Version 7.5/7.6 OLTP – SAP note: 767635

Parameter Notes liveCache:

Initial parameter setting SAP liveCache version 7.5/7.6 & 7.7: 719652

SAP liveCache version 7.9: 1567117

List of new liveCache parameters in version 7.9: 1693005

The parameter notes exist for each SAP MaxDB version. These notes do not describe the meaning of the MaxDB parameters, this information can be found for the general parameters in the SAP MaxDB documentation. These notes recommend parameter settings which differ from the default values.

Before you start a detailed problem analysis on the customer system please check if the parameters are set as recommended. You don't need to check each recommendation in the note with the parameter setting on customer side manually. With the DB-Analyzer parameter checker you can execute an automatic check. Please use note 1111426. Please always use the current DB-Analyzer configuration file which is attached to this note.

Questions

SAP® MaxDB™ Database Kernel Parameter

SAP® MaxDB™ – Expert Sessions Learning Map (1)

SAP® MaxDB ™ Features

- Session 1: Low TCO with the SAP MaxDB Database
- Session 6: New Features in SAP MaxDB Version 7.7
- Session 8: New Features in SAP MaxDB Version 7.8

SAP® MaxDB ™ Administration

- Session 2: Basic Administration with Database Studio
- Session 3: CCMS Integration into the SAP System
- Session 11: SAP MaxDB Backup and Recovery
- Session 13: Third-Party Backup Tools
- Session 19: SAP® MaxDB™ Kernel Parameter Handling

SAP® MaxDB ™ Problem Analysis

- Session 5: SAP MaxDB Data Integrity
- Session 14: SAP MaxDB Tracing
- Session 12: Analysis of SQL Locking Situations

SAP® MaxDB ™ Installation/Upgrade

- Session 7: SAP MaxDB Software Update Basics

All Expert Sessions (recording and slides) are available for download
http://maxdb.sap.com/training/

SAP® MaxDB™ – Expert Sessions Learning Map (2)

SAP® MaxDB™ Architecture	SAP® MaxDB ™ Performance
Session 18: Introduction MaxDB Database Architecture	Session 4: Performance Optimization with SAP MaxDB
Session 15: SAP MaxDB No-Reorganization Principle	Session 9: SAP MaxDB Optimized for SAP BW
Session 17: SAP MaxDB Shadow Page Algorithm	Session 16: SAP MaxDB SQL Query Optimization (Part 1)
Session 12: Analysis of SQL Locking Situations	Session 16: SAP MaxDB SQL Query Optimization (Part 2)
Session 10: SAP MaxDB Logging	

All Expert Sessions (recording and slides) are available for download
http://maxdb.sap.com/training/

Thank You!
Bye, Bye – And Remember Next Session

Feedback and further information:
http://www.sdn.sap.com/irj/sdn/maxdb

Next Session: 27.08.2013
SAP® MaxDB™ Remote SQL Server

Thank you

Contact information:

Christiane Hienger
Development Expert IMS
Christiane.Hienger@sap.com