MaxDB/liveCache Development Support
November 2009
MaxDB/liveCache Development Support
Heike Gursch
Christiane Hienger
November 24, 2009
Agenda

1. Introduction to Transaction DB59
   - Central Monitoring with DB59
   - Database Integration
   - Connection Test

2. Introduction to Transaction DB50
   - Menu Structure
   - Administrative Tasks

3. Alert Monitor RZ20

4. Preview of DBACockpit
   - Menu Structure
In this chapter, you will learn how to integrate SAP MaxDB databases into one central monitoring system using transaction DB59. Furthermore an overview of the database assistant transaction DB50 is given and you'll learn how to schedule regular administrative tasks using the DBA Planning Calendar. Finally you get an overview of transaction RZ20 – the SAP Alert Monitor.

Additionally a preview of the planned functionality of DBACockpit is given. You'll get an impression on how it will work and where to find the administrative tasks.
Your system landscape can consist of several different SAP and non-SAP systems. You might have an OLTP, an SCM and a Knowledge Warehouse system.

The OLTP system uses one SAP MaxDB instance, the SCM system two (the SCM database and the liveCache) and the Knowledge Warehouse system uses three SAP MaxDB instances (the KPRO database, Content Server and Cache Server). All of these databases should be monitored.
To make things easier you can choose any of your SAP systems and use that as your central monitoring system. This could be e.g. a Solution Manager system.

You can integrate all SAP MaxDB instances of your system landscape into this central monitoring system and use just one transaction to monitor all databases.

This central monitoring system may run on SAP MaxDB but monitoring transaction DB59 is also available on systems running on a different database system.
Initially DB59 only contains an entry for the system’s own MaxDB instance. In case of an APO/SCM system there are additional connection identifiers for liveCache.

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

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

Additionally all other SAP MaxDB Databases of your system landscape can be entered. Choose *Integrate Database*. 
Within our demo system (database S70) already a lot of different MaxDB databases and liveCaches have been integrated. If you want to add a new one choose Integrate Database and specify with Define Database Type if your system is a MaxDB or a liveCache instance.
On the next screen enter the requested information:
- **database name and database server**,  
- an optional **description**,  
- **dbm operator and his password**  
- **standard database user and his password**  
Then save your changes.
To check, if the entered connect information works, mark the entry of the concerning database instance and choose *Connection Test*.

You'll get a list of all application servers. Select one and choose *Connection Test* to check one after another if the connection works from all application servers.
The connection test first checks the DBMCLI connection, then the command and session mode of DBMRFC and afterwards the SQL connection. If everything is OK, a green check mark appears for this application server, otherwise a red cross is shown. In this case you have to check the log file.
This is an example for a frequent connect problem.

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

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

In this case you can find more information in the corresponding developer trace in transaction ST11.

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

Go back to the *Integration data* in transaction DB59 and reenter the user information.
When the entered connect information is correct and the connection test does not return any errors, you can open the Assistant for the integrated database. Select the newly created entry in the list of databases and choose Assistant or double-click the line.

For a SAP MaxDB instance transaction DB50 is opened, for a liveCache instance transaction LC10 is used.

These transactions are very similar – however, transaction LC10 contains some additional liveCache specific applications.
Further improvement of transaction DB50 is done continuously. The tree structure in the version you are using might differ slightly from the shown screenshots.

As it should not be allowed for all users to execute all administrative tasks there is a special user concept. You can define different roles which are described in note 452745 (and also in the documentation) specifying which tasks can be executed by the user. F.e. access to dbmcli should be restricted because not everybody should be able to start and stop the database.

This is the *Properties* section of transaction DB50.

In the properties section you can find some general information like the database name, the database server, the database version and the operational state of the database instance.
The second tab on the Properties window provides information about the directories in which the database software is installed.

In the first picture you find the directory structure of the stand alone test database. The software of S70 (second picture) is installed as it is usual in SAP environments. It is recommended to use the default for SAP installations as it simplifies error diagnosis.

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

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

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

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

The **rundirectory** contains the status files of a SAP MaxDB instance and is the most important directory for monitoring and error analysis.
On the third tab in the Properties window you can access the database log files.

The most important log files are:

**KNLDBG** or **KNLMSG** - contains status and error messages of the database kernel.

knldiag is used up to version 7.6, knlmsg is used as of version 7.7.

**KNLDBGERR** or **KNLMSGARCHIVE** - contains all error messages since database installation

**UTLPRT** - contains administrative commands sent to the database kernel (e.g. SHUTDOWN, BACKUP, CHECK DATA) including their return code(s). This logfile is obsolete as of version 7.7.

**BACKHIST** - contains all backup and recovery actions

**DBMPRT** - contains all (administrative) commands sent to the dbmserver
Open the tree item *Current Status* -> *Configuration* -> *Volumes.* The *Volumes* section shows all configured log and data volumes including their locations and sizes.

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

This is limited by the database parameter MAXDATAVOLUMES.

New volumes can be added using the Database Manager GUI or Database Studio.
This is the tree item *Current Status* -> *Kernel Threads* -> *Task Manager*.
The *Task Manager* shows the status of database tasks. With the tabs you can choose if you only want to check the currently active tasks, the runnable tasks, the user or system tasks or all tasks.

In a running system, possible states are e.g.:
- **Running** – task is in kernel code of SAP MaxDB and uses CPU
- **Runnable, Vsleep** – task is in kernel code of SAP MaxDB and waiting for a free slot in its thread (UKT)
- **LogIOWait** – task waits for completion of its log request by the archive log writer
- **IOWait (R) or IOWait (W)** – task waits for data I/O completion (read or write)
- **Vbegexcl or Vsuspend** – task waits to acquire an internal lock in SAP MaxDB
- **Vwait** – task waits for an SQL lock held by another application process to be released (locks are released after a COMMIT or ROLLBACK)
Information about the different memory areas and the hit rates of these caches can be found in *Current Status* -> *Memory Areas* -> *Caches*. The Data Cache hit rate should always be above 98%.
The **Data Area** can consist of several data volumes. The **Total Size** shows the sum of the sizes of all data volumes. You can see the filling level of the data area as well as the proportion of temporary data. The data is automatically distributed to all volumes equally. You don’t have to define table spaces. Empty data pages are reused by the database automatically. No reorganization is necessary.
The Log Area can consist of several log volumes - however they are used as one single log area.

The Log Segment Size determines how large the log backups are created by the Automatic Log Backup.

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

For test or demo systems it is possible to activate an overwrite mode for the log volumes - then you don't have to take log backups before the information on the log volume can be overwritten.

It is also possible to switch off the writing of log information.

Both of these possibilities are not recommended for productive systems as you won't be able to restore the database to the latest state.
This is an example for the `knldiag/knlmsg` file. It can be displayed using tree item Problem Analysis -> Messages -> Kernel -> Current.

File `knldiag/knlmsg` has a fixed size. It is initialized when the database is started.

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

`knldiag/Knlmsg` consists of two parts:
the first part contains information about the database start and is not overwritten.

In the second part information is logged during the runtime of the database. This part is overwritten cyclically. The current write position is marked with `--- current write position ---`

In this example restart information is provided. You may check the proceeding of the restart, identify savepoint phases and watch redo states. When the database is online, connects are logged.

In case of problems with the database you should always check file `knldiag/knlmsg` for error messages.
This is an example for the `knldiag.err/knlmsgarchive` file.

It contains message '--- Starting...' whenever the database has been started from state OFFLINE to state ADMIN and another message when the database is stopped.

All other messages are error messages - e.g. information about a crashes, including a back trace which can be used by the developers to find the cause of the crash.
If the database is not stopped correctly the most important log files are saved in the **DIAGHISTORY** folder during the next start of the database instance. This ensures that they are not overwritten and can still be analyzed to determine the cause of the crash.

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

As a default two sets of log files are held in the diaghistory.
The **DBA History** contains information about administrative tasks and is available in **Problem Analysis -> Logs -> DBA History**. E.g. you can see information about executed backup and recovery actions or consistency checks. You can display a detailed log file for each of these actions. To do so perform a double-click on the corresponding entry or use the marked button.

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

- the creation date of the backup
- the number of pages transferred
- the backup label
- the location of the backup file
This is the *Backup History* from the point of view of the database kernel. Each log backup action might create several log backup files - each of the size of one log segment. The HISTLOST entries are created whenever the log volumes are initialized - e.g. during the installation. Then you have to create a complete backup again to start a new backup history.
You can check the current parameter settings and the change history of each parameter. Therefore open the tree item Administration -> Configuration -> Parameters.

To be able to change the parameter settings you have to switch to the change mode.

Most parameters can be changed in online mode and take effect immediately. For some parameters it is necessary to restart the database before they are valid.

(See note #814704 for online changeable parameters).
In the *Parameter History* you can check when and how database parameters have been changed.
The Backup Templates section provides an overview of all defined backup templates. It is also possible to define new backup templates. For faster backups SAP MaxDB supports parallel backups to several files/tapes/pipes. Therefore a template group has to be defined which consists of several single backup templates. If a template group is used for a backup the data is distributed to the specified files/tapes/pipes. These are created in parallel which makes the backup faster. In case of a recovery all files/tapes/pipes used for the backup are needed. However, parallel created files/tapes/pipes can also be restored sequentially.

The following external backup tools can be used to create backups:

- Tivoli Storage Manager
- Networker
- Tools which support the Interface BackInt for Oracle or Backint for SAP MaxDB
In the Tools section you can execute Database Manager CLI commands. Some of the most important commands are:

- `db_state` – determines the database state
- `dbm_version` – determines the version of the dbmserver
- `db_offline` – stops the database instance – should not be executed for the systems own database instance!
- `db_online` - starts the database instance
In the **Tools** section you'll find a list of commands that can be issued in the **Database Console**.

If you choose the button „Enter Command“ you can directly specify commands for the Database Console tool `x_cons` which might be some more sophisticated.
The *DBA Planning Calendar* allows to schedule important database tasks like backups or consistency checks (like transaction DB13/DB13C in earlier SAP releases). Double-click a line in the calendar view or in the *Action Pad* to schedule a task.

For some tasks parameters are necessary – e.g. the backup template for a backup. These parameters can be specified in this window. Furthermore you can specify if the task should be executed immediately or if it should be executed at a specific date/time and if it should be executed in a certain interval (daily, weekly, every few hours, …)
If you perform a double-click on a finished action you can have a look at the log files of this task.
If you want to have a look at monitoring information call transaction RZ20. Under *SAP CCMS Monitors for Optional Components* you find *MaxDB Monitoring*. 
The Alert Monitor collects e.g. data concerning the filling level of the log and the data area, the cache hit ratio and the creation of backups.

If you perform a double-click on a node in the tree, you get detailed information about this node - e.g. the backup history. In this example no data backup exists.
You can use the *DBA Planning Calendar* to create the backup.
For the *MaxDB Alert Monitor* there are 4 data collectors which collect the data for all nodes. These collectors run in different intervals.

After you have solved the problem, you can either wait until the responsible data collector runs the next time or start the data collector manually. To figure out which data collector is responsible for this node, place the cursor on this node and press *F1* then choose *Long Text.*
To be able to start the data collector, you have to activate the maintenance function in transaction RZ20. Choose **CCMS monitor sets** -> **SAP CCMS Technical Expert Monitors** -> **All Monitoring Contexts**.

Then choose **Extras** -> **Activate maintenance function**.
Then you can select the data collector (make sure the checkbox is marked) and choose Edit -> Nodes (MTE) -> Start methods -> Start data collection method. If you refresh the display in RZ20 then, the backup node will be green.
DBACockpit functionality will sooner or later replace DB50 and DB59. You will find all functions described so far also in a newly ordered tree of DBACockpit. Furthermore if you have to administer different databases you will locate similar administrative tasks under the same titles.

Firstly choose the database in a list of system names which are ordered alphabetically.
Under *Current Status* -> *Attributes* you'll find some general information about the database that was located under *Properties* before. An overview of the most important settings is provided to get a first impression of the database state and which monitoring is activated.
Under *Performance* all functionality is summarized that might help to investigate performance problems on your system. The known MaxDB tools Database Analyzer, Command and Resource Monitor are located here. As SQL locks that are hold too long or erroneously might seriously affect the performance of a system relevant information can also be found under this subtree.

Details about kernel thread behaviour and their states can be found in the *Kernel Thread* section, administrative tasks are possible with the known task manager (e.g. cancelling of a command or ending a task). I/O operations can be checked as well.
Under *Space* different filling levels can be checked. Hit ratios of the caches are provided as well as filling degree of the data and log area on the disks.

Scheduling of actions like backup, update statistics and check data is located under *Jobs* and can be done with the *Central Calender*.

The *Alert Monitor* is integrated in DBACockpit.

Tracing possibilities and different tools for error diagnosis can be found under *Diagnostics*. Direct access to the diagnosis files is provided here.
As known from DB50 under *Administration* database parameters can be checked and backup templates created and maintained.

Direct access to the MaxDB tools is provided under *Tools*.

Find more detailed information about MaxDB/liveCache and the DBACockpit in the documentation which can be accessed directly.
Now you know how to integrate all SAP MaxDB databases into a central monitoring system. You got an impression of the numerous functions of transaction DB50 to monitor and administer SAP MaxDB databases. You also know how to schedule administrative tasks in the planning calendar and how to check the log files of these actions.
| December 8, 2009 | Session 4: Performance Analysis Part 1 |

© SAP AG 2009. All rights reserved / Page 41