SAP MaxDB version 7.8 is currently released with SAP Netweaver 7.2 (which is currently in Ramp-up phase).

It might be available for older SAP releases later on. Please always check in the Product Availability Matrix on the Service Marketplace if a new database version is released for your specific SAP release, before installing it.
This document describes *major* changes between MaxDB 7.8 and older versions. However, it does not contain *all* changes of MaxDB version 7.8. For details please refer to the (online) documentation. Some features have been implemented in lower versions as well.

Even if some of the features described below are not directly perceptible, they might have a strong impact, e.g. on performance or scalability, and thus are briefly described.
This session describes new features in the areas of Administration, Performance and Monitoring.

It starts with information about the isolated software installation, followed by the advanced database installation wizard, enhancements regarding system security, backup enhancements, new database events and the automatic recreation of indexes.

Two short demos complete this first part.
In SAP DB / SAP MaxDB versions 7.2 – 7.7 all databases running on one server are sharing the independent data and the independent programs directory. The independent data directory (e.g. /sapdb/data) contains the configuration data and rundirectories of SAP MaxDB instances. The independent programs directory (e.g. /sapdb/programs) contains programs and libraries shared by the SAP MaxDB instances and SAP MaxDB applications. These programs are downwards compatible. They are installed with the same version as the highest MaxDB software version installed on that server.

As these programs and libraries are shared by all databases running on that server, all databases are influenced, when this software is updated. For example the tcp/ip listener (X-Server), which handles remote connections to the database, is part of the independent programs directory. When the independent software is updated, all remote connections to a server need to be stopped (on Windows and in versions < 7.4.03 on UNIX).

To solve such problems, the database software is fully isolated as of version 7.8. The upgrade of one database instance does not influence any other database on the same server anymore. Furthermore it is now possible to install more than one client software on one server (in older versions the client software was part of the independent programs directory).

This new software installation is downward compatible to older MaxDB versions.
The *global program path* still contains components shared by all MaxDB installations. But these are only installer tools and the global listener, now.

Each installation now has its own X-Server, which is part of the *installation path*. The global listener only routes the remote connections to the X-Server of the specific database.

For each database or client installation, an own *installation path* needs to be specified. This directory contains all relevant software.

The *global data path* contains the installation registries and log files, data files needed by the global listener and - if the *private data path* option is not used - the rundirectories of the database instances.

**DBMCLI commands to get the relevant directories:**

```bash
dbmcli inst_enum <installation path>
```

Shows e.g. the data path and the port number used by the X-Server of this installation.

```bash
dbmcli –d <database name> dbm_getpath
```

Shows e.g. the installation path, the private data path, the global program and global data path.
The interfaces and the client software, which were part of the *independent global path* in earlier versions, are now part of each MaxDB server installation. The client software can furthermore be installed as a stand-alone installation side-by-side to the MaxDB server installations. Several client installations – of the same or of different versions – can be installed side-by-side as well. This way different application servers running on the same host can use their own client installation. Just make sure that the PATH variable used by each application server points to the correct *installation path*.
If older MaxDB versions are installed on the same server, the global listener acts as X-Server for these databases.

The *global programs directory* then contains the interfaces and clients of these older software installations.

The clients and interfaces of version 7.8 are stored in the *installation path* of that installation.
No private data path activated:
The install registry and installation log files are now stored installation-dependent in different subdirectories of the global data path. The install registry of the global components is stored in directory `<global data path>/globals`

The database rundirectories are still stored in `<global data path>/wrk` and the parameter files are still stored in `<global data path>/config`

If older MaxDB versions are installed on the same server, their installation log files are stored in `<global data path>/wrk` and an additional install registry is stored in `<global data path>/config`
For SAP installations the private data path is activated. This causes:

- the database rundirectory to be stored in `<private data path>/wrk`
- the database parameter file to be stored in `<private data path>/config`
- the install registry and installation log files to be stored in `<private data path>/installations`.

The data of older installations is still stored in the old locations. Installations with private data path enabled can exist side-by-side with installations without private data path enabled.
All Expert Sessions use a database called EXPERTDB. This short demo shows the directory structure of this database, which was updated to version 7.8 in the last Expert Session.
In the *Create Database Wizard* of Database Studio you can choose to use either a simple or an advanced mode for the database installation.

In the *simple installation mode* only a few database parameters have to be set manually, most parameters are pre-defined. The database is installed with one data and one log volume.

In the *advanced mode* all parameters can be set and any number of volumes can be configured. Furthermore in the advanced mode it is possible to restore a backup during the installation process. This is e.g. used for a system copy.
SAP MaxDB clients with software version < 7.8 can only handle users with a maximum password length of 9 characters. So if an old client software version is used in combination with a MaxDB version >= 7.8 the used passwords still can only have a length of 9 characters. Otherwise the old client won’t be able to connect to the database.
With MaxDB version 7.8 new backup features are supported. To be able to do that, the backup templates had to be extended. Therefore they are now stored in an XML format and new dbmcli commands were implemented to maintain the backup templates:

```
backup_template_create <template_name> TO ( <ext_snapshot_rule> | ( <backup_destination> <backup_content> [<template_options>]) )
backup_template_delete <backup template name>
backup_templates_file_date_show
backup_template_show ([BY] NAME <backup template>) | ALL
backup_template_show_next
```

New **Backup Template** definition supporting all relevant backup/restore scenarios

- New syntax for *Backup Template* (former Medium) creation
- Supporting external backups via snapshot clone/split mirror, encrypted backups and compressed backups
- Templates are stored in XML format in file *BackupTemplates.xml*.
A backup of the data area exists only when the data area is copied physically. This applies to the consistent split of a mirrored disk or the cloning of a consistent snapshot, for example. A snapshot in a disk system is not a data backup if there is no physical copy of the snapshot data.

Snapshot clones and split mirrors are not controlled by the MaxDB backup mechanism. For a system restore, you must be able to perform the backup without MaxDB tools.

A backup of the data volumes at file system level is not a data backup because this does not guarantee the input/output (I/O) consistency.

**DBMCLI commands for external backups:**

```bash
backup_template_create <template_name> TO EXTERNAL SNAPSHOT
backup_start <template_name>
backup_finish <template_name> ExternalBackupID <external_backup_ID>
```

| template_name | AS FAILED [WITH [ERROR] tool_rc] |
The backup compression can only be used for backup templates of the FILE and PIPE types.

In tests a compression to 25% of the original backup size has been reached for standard SAP systems.

DBMCLI commands:
backup_template_create <template_name> TO <backup_destination> COMPRESSED <backup_content> [<template_options>]
backup_start <template name>
Prerequisite: You have created a private-public key pair using sapgenpse. For more information about sapgenpse and SAP Cryptolib, see the online documentation, chapter *Using the SAP Cryptographic Library for SNC*.

When creating the backup template for your backup, specify the encryption algorithm and the private key owner in the backup template properties.

**DBMCLI commands:**

```
backup_template_create <template_name> TO <backup_destination> <backup_content> ENCRYPT [USING] <AES128-CBC | AES192-CBC | AES256-CBC | DES-CBC | DES3-CBC> [FOR] {OWNER <owner>}
```

```
backup_start <template name>
```
The database manager event dispatcher was already explained in detail in the Expert Session ‘New Features in SAP MAxDB version 7.7’. With database version 7.8 a couple of new events has been introduced. They can be configured using dbmcli or Database Studio.

**DBMCLI commands:**

```plaintext
event_dispatcher ADD NAME == <event_name> [PRIORITY == <priority>]
   [VALUE1 (==|>=|<=|>|<) <value1>] [VALUE2
   (==|>=|<=|>|<) <value2>] COMMAND == <command>

event_dispatcher SHOW

event_dispatcher DELETE <entry_ID>
```
You can instruct the database to recreate bad indexes automatically. You can also specify whether you want all bad indexes to be recreated automatically or only unique indexes that prove to be bad. If the function is ON, the system recreates bad indexes after having received a respective event from the database kernel.

DBMCLI-Command:

`auto_recreate_bad_index <mode>`

Options:

- `<mode>`: New mode for the database. Possible values are:
  - ALL: All indices that are detected to be bad are automatically repaired.
  - UNIQUE: Unique indices are automatically repaired if they are broken.
  - OFF: The database system does not perform any action if bad indices are detected.
  - SHOW: Show current status of automatic bad index recreation feature.
In this short demo it is shown how a normal and a compressed data backup can be created. The size difference is shown.
Furthermore it is explained how to configure a backup template for encryption and for external snapshots with Database Studio.
Additionally an event is configured, which sends an email in case the automatic log backup gets disabled.
In this chapter about performance improvements, the kernel features dynamic CPU adjustment, load balancing, prefetching, volume balancing, cache pinning and variable output are explained.
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.

The parameter can be changed in ONLINE mode.

This is especially useful in systems were hardware resources can be dynamically added or removed.
Load balancing enables an optimal utilization of all threads and thus of all the CPUs allocated to the database.

After the time interval of LoadBalancingCheckInterval seconds, the database kernel searches for a task to be moved to another UKT. This is helpful when one UKT has a particularly heavy load and another UKT has a smaller load.

Between the checks after LoadBalancingCheckInterval seconds, statistics are collected. The greater the time for gathering the data, the more meaningful the UKT load statistics that result. With smaller values, the shifting that occurs may not be optimal.

Moving tasks is time-consuming. It should only be done when it is expedient. Two more parameters LoadBalancingWorkloadThreshold and LoadBalancingWorkloadDeviation influence the behavior to make sure that not too many tasks are moved. In OLTP operation, unbalanced load distribution among the UKTs is often due to poorly-optimized statements with high CPU loads for a single job. Such statements should be identified and optimized. But nevertheless there are situations where load balancing makes sense and improves scalability.

In MaxDB versions < 7.8 load balancing was particularly used for liveCache instances. These often run very CPU-intensive LCA routines over long periods. Multiple LCA routines should not work sequentially on one CPU if another CPU is free.

The database console (x_cons) provides information about tasks that have been moved.

The parameter LoadBalancingCheckInterval can be changed in ONLINE mode.
MaxDB uses parallel I/O requests to speed up table scans and table range scans. User tasks read index level 1 pages into the cache, determine the volume block positions of the pages stored in the separators by reading the converter and send asynchronous I/O requests to the I/O system. The user task doesn't wait for every single I/O before sending the next I/O request.

User tasks use asynchronous I/O requests if the size of the scan exceeds the number of pages specified as value of the parameter ReadAheadTableThreshold. The query optimizer evaluates the range size before the statement execution starts.

The database uses asynchronous I/O for scans only, if the number of current running I/O requests is below the value of MaxDataAreaReadAheadRequests. The determination of the current running I/O requests happens during the operation on the index level 1 page. This operation prevents the system from I/O overload situations. I/O orders for short SQL commands should not be blocked by the asynchronous parallel I/O.

Asynchronous I/O read requests have the priority low.
An important role in the access performance of the database is played by the MaxDB striping mechanism, which distributes the data pages evenly on the disks. Additional striping can be performed by the hardware. Striping guarantees even distribution of the I/O load on the available disks. Even load balancing of all the data volumes in the database also prevents individual data volumes from overflowing. A table can be larger than a single data volume without the need for maintenance tasks to be carried out.

If new data volumes are added the database must ensure that some data is copied automatically to the new disk to get an even distribution again. This additional I/O must be done without affecting the normal operation.

In versions 7.4 – 7.7.05 volume balancing is not available. As of version 7.7.06.08 the feature has been introduced again.
As of version 7.8 MaxDB can hold pages of tables and indexes having the CACHE attribute in a LRU pin area of the data cache. The parameter `DataCachePinAreaThreshold` defines the maximum percentage of the data cache used for tables having the CACHE attribute.

Pages that have to be swapped out of the pin area will be inserted into the global LRU list at the beginning of the last 10%. The pages will be linked back into the beginning of the PIN LRU list when they are reused.

Additionally the SQL Manager can occupy pages in the cache. The pages are marked as “in use” for a longer timeframe. Those pages will not be swapped out of the cache because they are in use. This allows a direct memory access by the SQL Manager to the cache pages. The creation of result sets speeds up significantly. (Affected Parameter is `TempResultsetPinAreaThreshold`.)

All liveCache data is considered to have the CACHE attribute and thus will use the part of the data cache reserved via `DataCachePinAreaThreshold` if the parameter `EnableLivecachePinAreaUsage` has the value YES.
The parameters **EnableVariableInput** and (as of 7.8 additionally) **EnableVariableOutput** determine how the system transfers the content of variables between client and kernel.

The system either transfers the actual length of the content or the maximum possible length of the content. In the latter case, the system fills up the content up to the maximum length (depending on the data type) which is space consuming. Setting the parameter to YES reduces communication costs.

Changes to these parameters only apply after a restart of the database.
In this last part of this session two monitoring enhancements of Database Studio are presented: the Databases Monitor and the SQL Performance Analysis.
The Databases Monitor provides an overview of the state of all databases integrated in one landscape in Database Studio. On the first glance you’ll get information about critical database states like db_full, log_full or database down.

Opening the Databases Monitor:
In the explorer tree, go to the top level using the (Up One Level) button.
Select the database landscape that you want to monitor.
In the context menu of the database landscape, choose Databases Monitor.

It is possible to add or remove columns from the displayed view. Furthermore the refresh interval can be specified and the Databases Monitor can be configured to be opened automatically, when Database Studio is started.
SQL monitoring measures runtimes, I/O accesses, and the selectivity of SQL statements, and calculates the total costs for repeatedly executed SQL statements. This lets you identify SQL statements that have short runtimes individually, but whose frequent execution results in a heavy load.

This monitoring is based on the Shared SQL cache and is activated as a default. The data is collected in system table COMMANDSTATISTICS.
SQL performance analysis in Database Studio lets you investigate which SQL statements that the application triggers are particularly cost-intensive for the database, and why this might be the case.

As already explained, SQL monitoring is activated as a default.

If you activate Detail Collection as well, then the system records the executions of SQL statements that meet the criteria you define, as well as the parameters used. If you do not activate the Detail Collection, the system only records aggregated values for the execution of SQL statements.

Filters let you restrict the list of SQL statements that the system records in SQL monitoring. All the SQL statements that fulfill at least one of the filter conditions are displayed (OR operator). The following types of filters are available:

Cumulative Filters: These filters use the information that the system collects during simple SQL monitoring. These filters help you find SQL statements that the system runs frequently. Minor improvements in the performance of such SQL statements can result in large performance gains in the overall system.

Single Filters: These filters use information that the system collected in the Detail Collection. They help you find SQL statements for which single executions are particularly slow. When you change the filter condition of a single filter, the system modifies the corresponding Detail Collection condition in the SQL monitoring settings automatically – i.e. the recording criteria of the Detail Collection is changed.
System table SESSIONS contains the CURRENTCOMMANDID. Using this column a JOIN to table COMMANDSTATISTICS can be executed, which provides the information, which SQL statement is currently executed by a specific database session. Further useful information is provided in column APPLICATIONPROCESS. With this information one can check in transaction SM50/SM66 which application process and which application program is executing that SQL statement.
As of SAP release 7.20 Basis SP 03 the *Task Manager* in transaction DBACOCKPIT shows the SQL statements which are executed by active user tasks. A double-click on the statement shows the complete SQL statement.
Thanks a lot for your interest in this session.
The recording and the PDF will be available at [http://maxdb.sap.com/training](http://maxdb.sap.com/training) soon.
Please also mark your calendar for our next session. Topic *SAP MaxDB Optimized for SAP Business Warehouse* will be presented on May 04, 2010

| May 04, 2010 | Session 9: SAP MaxDB Optimized for SAP Business Warehouse |
Thank you!