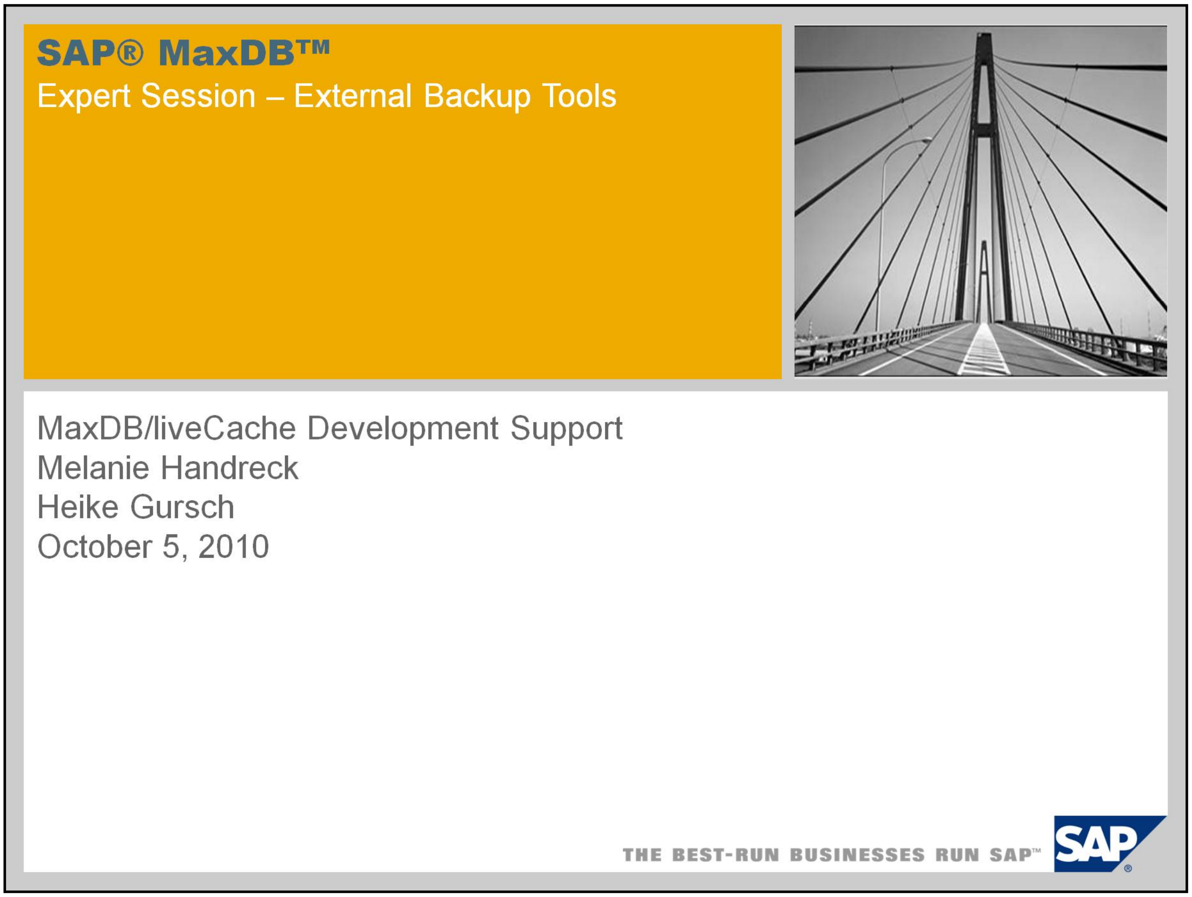


SAP® MaxDB™

Expert Session – External Backup Tools

MaxDB/liveCache Development Support
Melanie Handreck
Heike Gursch
October 5, 2010

THE BEST-RUN BUSINESSES RUN SAP™

Together with SAP MaxDB database tools, you can use third-party backup tools to backup and restore data. You can use third-party backup tools for the following actions:

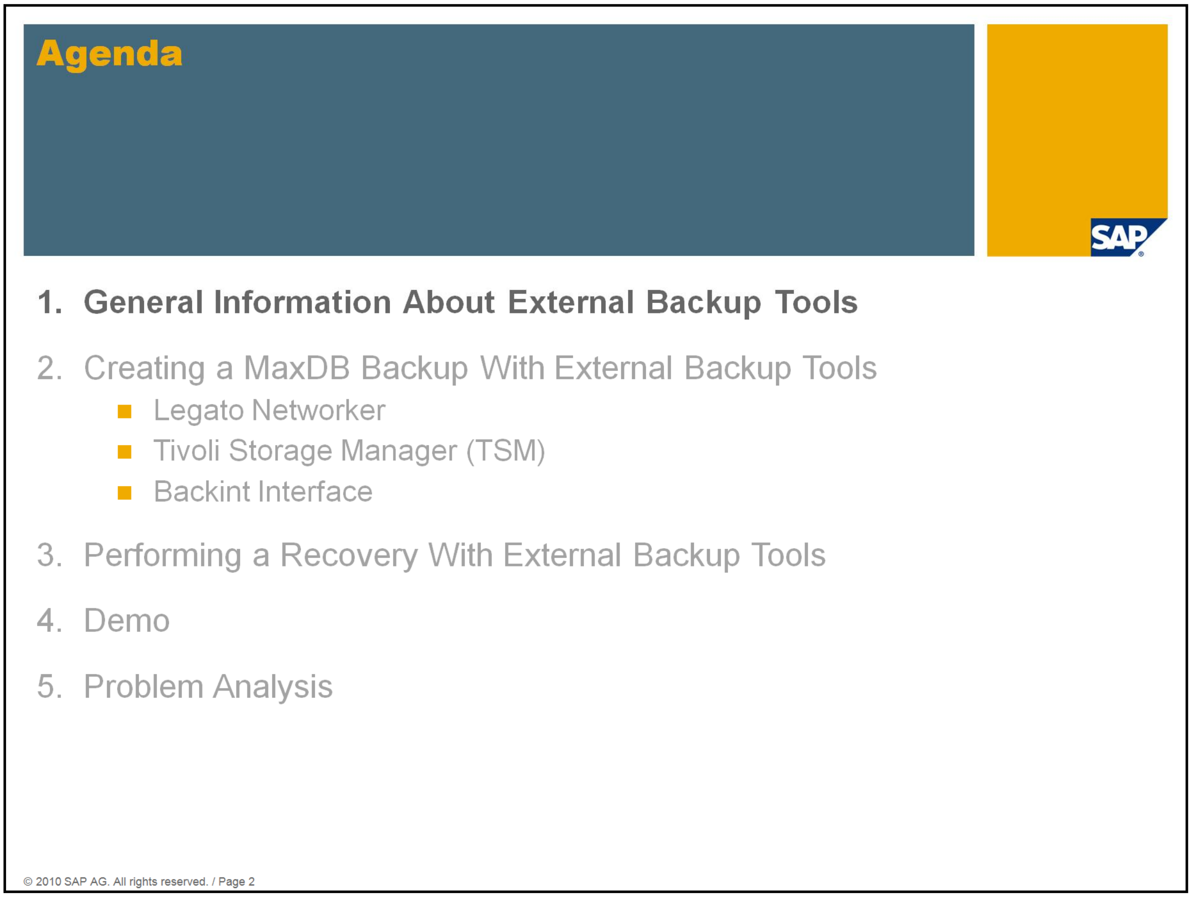
Backing up to data carriers

- Complete data backups
- Incremental data backups
- Log backups
- Archiving log backup files

It is not possible to save the data created by the automatic log backup directly to external backup tools. The automatic log backup can only create versioned files, which have to be saved to an external backup tool later on.

Restoring backups

- Restoring the database using data backups, incremental backups, log backups, and archived log backups



Agenda

1. **General Information About External Backup Tools**
2. Creating a MaxDB Backup With External Backup Tools
   - Legato Networker
   - Tivoli Storage Manager (TSM)
   - Backint Interface
3. Performing a Recovery With External Backup Tools
4. Demo
5. Problem Analysis

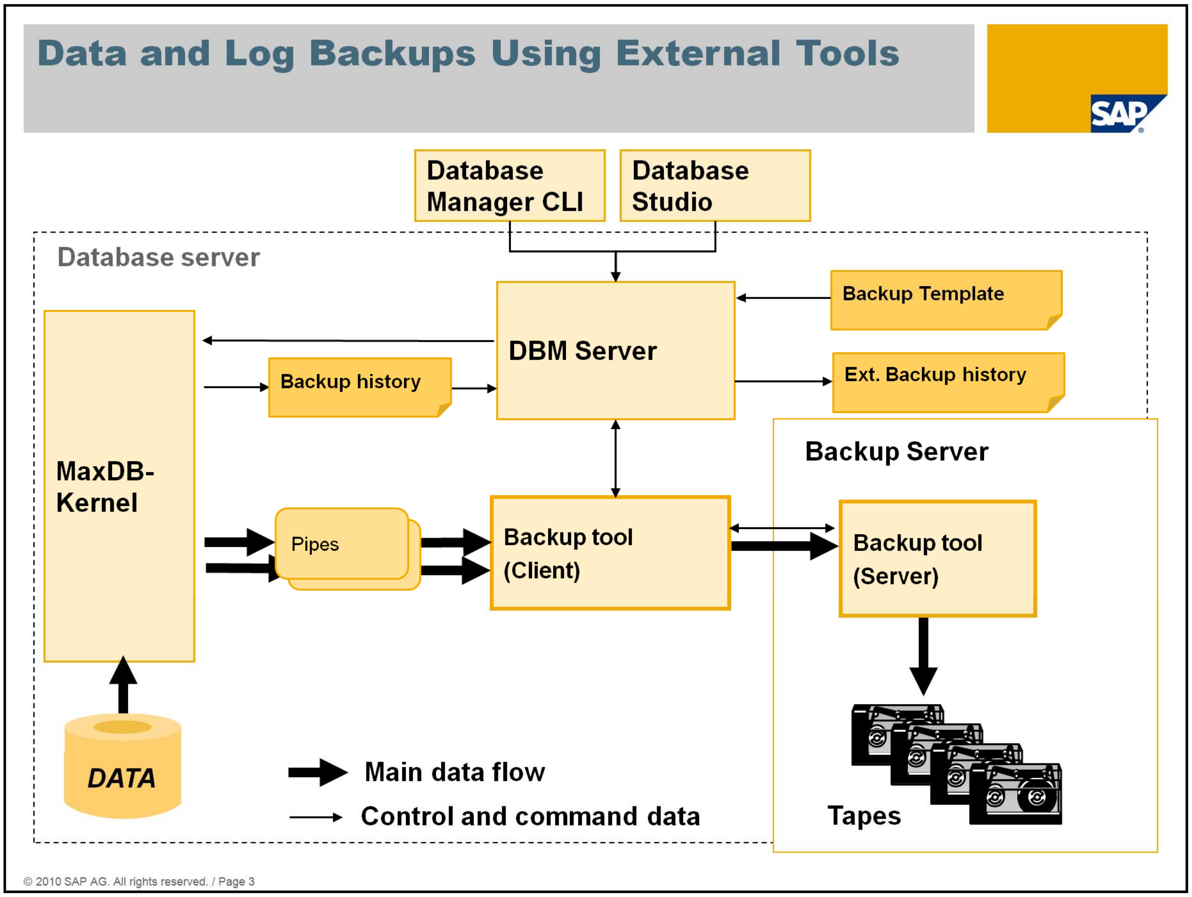
In one of the previous SAP MaxDB expert sessions the general concept of backup and recovery was explained. It was also shown, how a backup directly to files or tapes can be created. This session provides detailed information about the usage of external backup tools like Networker and Tivoli Storage Manager or the Backint interface. The first chapter concentrates on the general usage of external backup tools. It is explained, which tools are supported and how a backup to an external backup tool is controlled.

Afterwards the usage of different backup tools is explained in more detail – especially how these backup tools need to be configured for backing up a MaxDB instance.

The next chapter explains how a recovery can be performed, when an external backup tool is used.

Then we show in a short live demo, how a backup and recovery with the backint interface is done, using Database Studio.

And the last chapter provides an overview of the log files, to enable you to analyze errors which might occur during such a backup or recovery.



Data and Log Backups Using External Tools

Database Manager CLI

Database Studio

Database server

DBM Server

Backup Template

Ext. Backup history

Backup history

MaxDB-Kernel

Pipes

Backup tool (Client)

Backup Server

Backup tool (Server)

DATA

Tapes

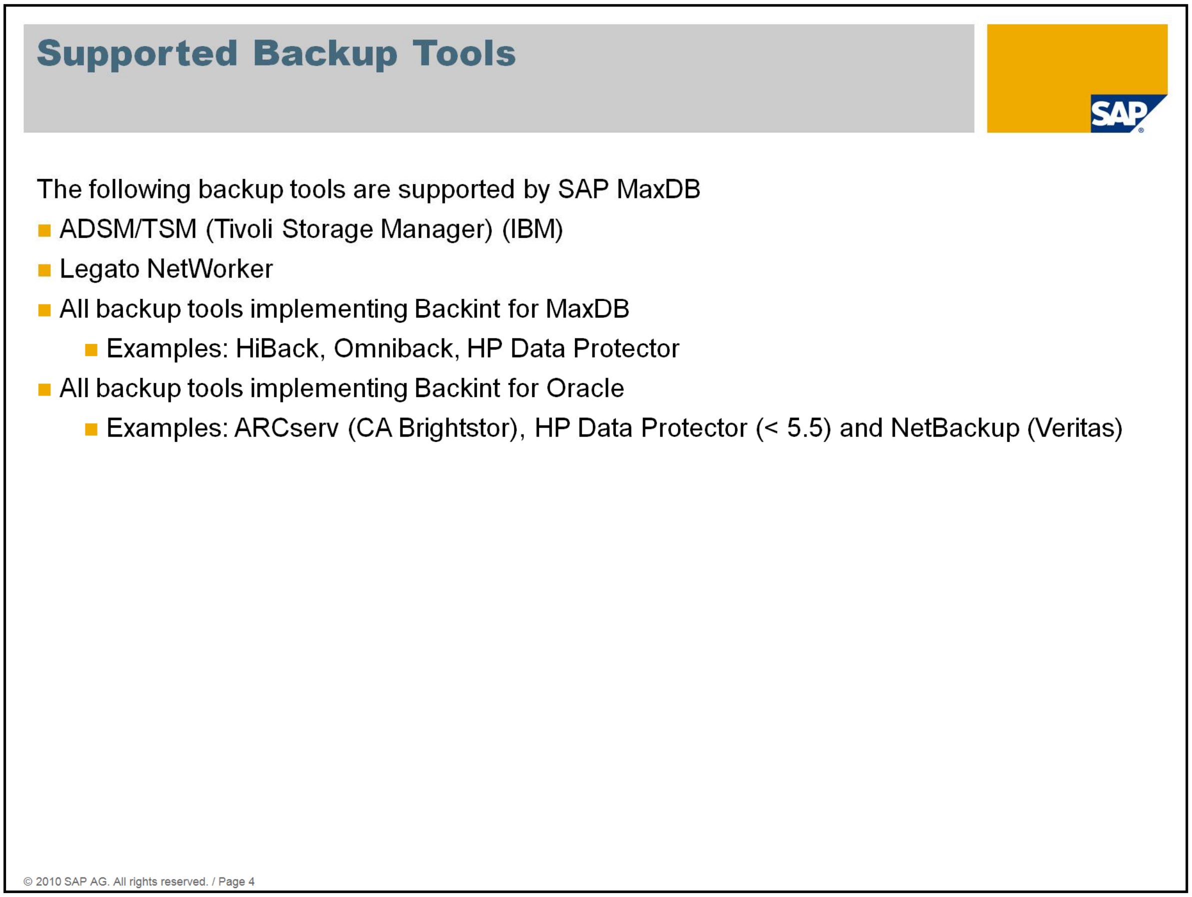
Main data flow

Control and command data

Backups for SAP MaxDB are always triggered by the DBM Server – either through the DBMCLI or through Database Studio. The DBM Server is also the component which starts the external backup tool. The backup procedure works as follows:

1. The DBM Server sends the backup command to the database kernel.
2. The database kernel creates and opens one or more pipes (as specified in the backup template used by the DBM Server).
3. The DBM Server starts the backup client of the backup tool as soon as the database kernel opens the first pipe. Which backup tool is to be used is also specified in the backup template.
4. The backup tool opens the pipes, transfers the data to the backup server, and stores it on tape.
5. The database kernel records the result of the backup in the backup history.
6. The DBM server requests the unique backup IDs (External Backup ID) from the backup tool and enters these in the External Backup History (dbm.ebf). -This makes it possible to link the backup IDs generated by the database kernel with the backup ID of the external backup tool.
7. The backup is logged in the External Backup Protocol (dbm.ebp).

External backup tools can not be used directly for automatic log backups. Automatic log backups are triggered directly by the database kernel, which isn't aware of the configuration of external backup tools. Automatic log backups can only be performed to versioned files. However, the usage of a so called log staging area is supported which can be configured in a way that the versioned files created by the database kernel are backed up to an external backup tool. Details about this follow later in the session.



Supported Backup Tools

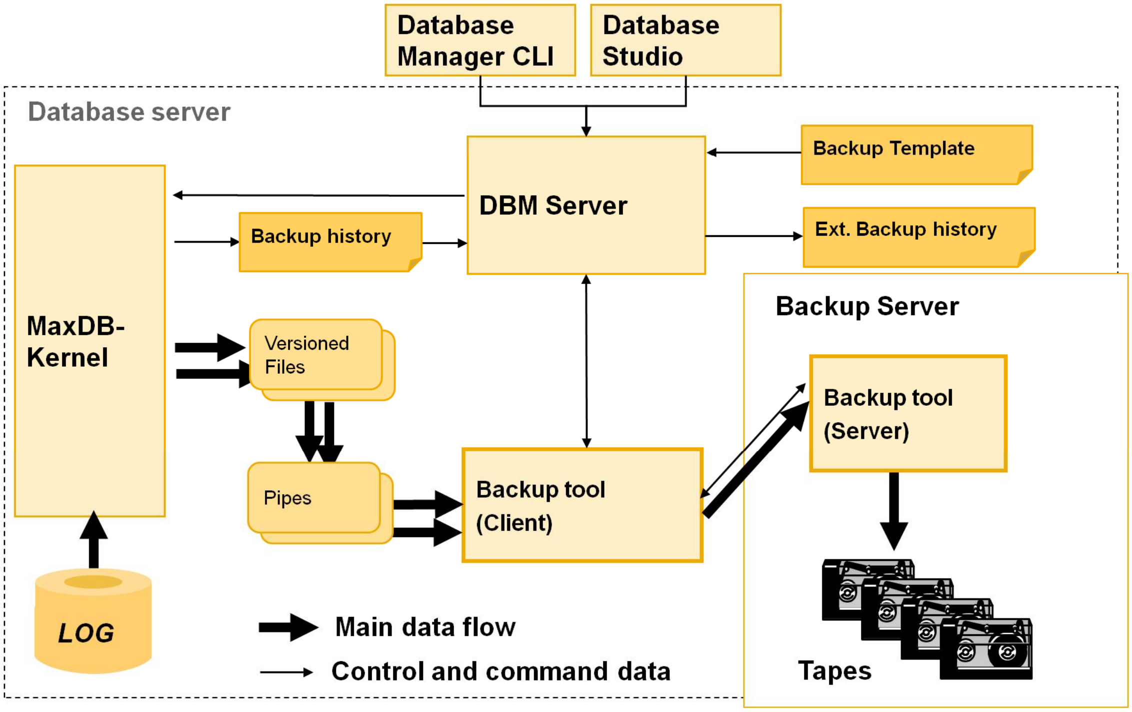
The following backup tools are supported by SAP MaxDB

- ADSM/TSM (Tivoli Storage Manager) (IBM)
- Legato NetWorker
- All backup tools implementing Backint for MaxDB
  - Examples: HiBack, Omniback, HP Data Protector
- All backup tools implementing Backint for Oracle
  - Examples: ARCserv (CA Brightstor), HP Data Protector (< 5.5) and NetBackup (Veritas)

To be able to support any backup tool which implements the Backint for Oracle interface, SAP provides an implementation of the Backint for MaxDB interface.

This program can be configured to work with an implementation of the Backint for Oracle interface (works as an adapter).

Automatic Log Backups Using External Tools



Database Manager CLI

Database Studio

Database server

DBM Server

Backup Template

Backup history

Ext. Backup history

MaxDB-Kernel

Versioned Files

Pipes

Backup tool (Client)

Backup Server

Backup tool (Server)

LOG

Tapes

Main data flow

Control and command data

If automatic log backup is switched on, the database system continuously writes redo log entries to backup files.

The database kernel only supports to create automatic log backup files into versioned files – direct usage of external backup tools or tapes is not possible. This is because the support of external backup tools is implemented in the DBM Server – not the database kernel. But the automatic log backup is completely managed by the database kernel.

After the automatic log backups are completed and stored in the log staging area, these files must be postprocessed, that is, saved to a secure destination (e.g tapes, external backup tools).

In earlier versions of the SAP MaxDB (< 7.4.02), administrators had to create their own scripts/concepts to do this.

The "archive_stage" DBMCLI command now provides a simple solution, whereby the DBM Server assumes the task of sending the AutoLog versioned files to an external backup solution. The DBM Server always knows which files are being written and which are ready for further processing at any time.

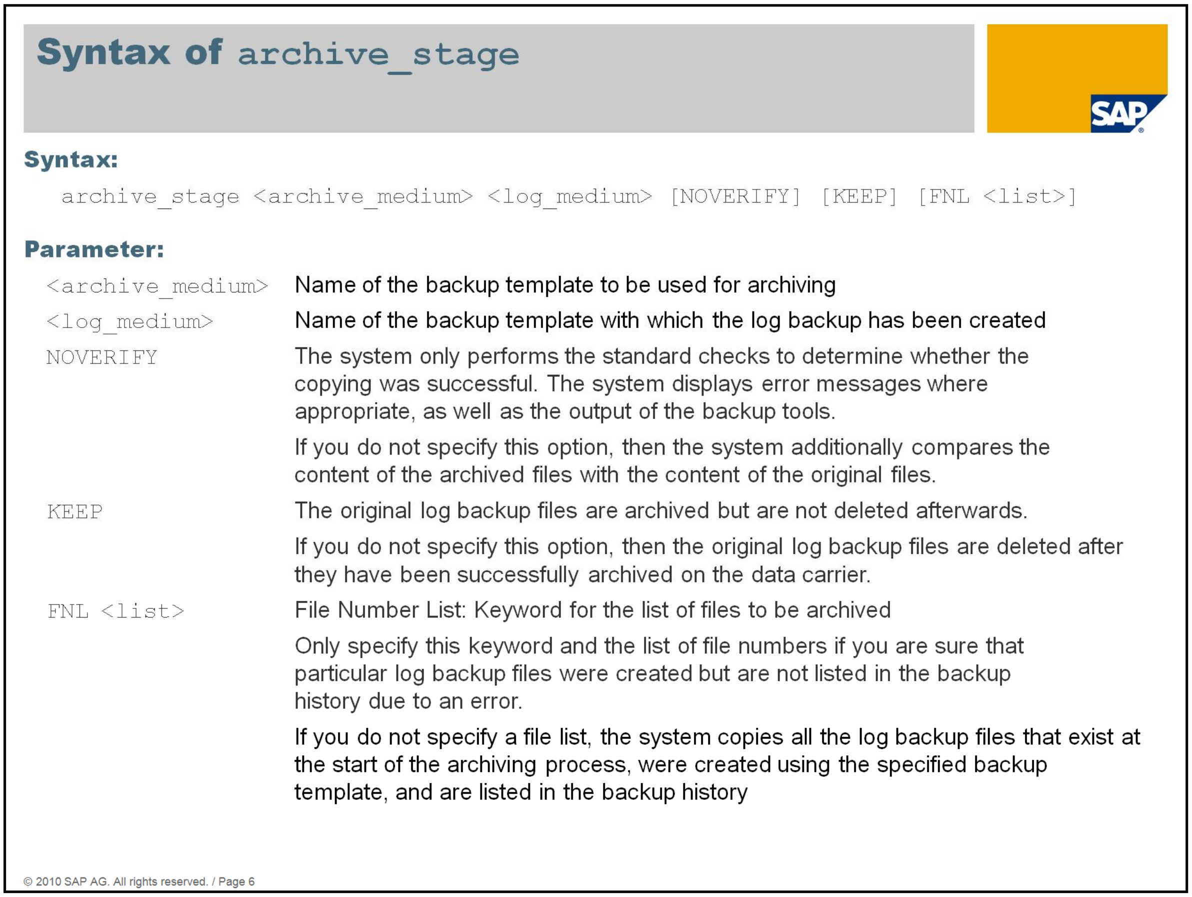
The DBM Server takes the finished files from the log staging area and sends them to the external backup tool using an existing log backup template.

When the archive_stage command is started the system copies all the log backup files that

- exist at the start of the archiving process,
- were created using the specified backup template,
- and are listed in the backup history.

If not specified otherwise, then the original log backup files are deleted after they have been successfully archived on the external backup tool.

The archive_stage command should be scheduled regularly using an OS-scheduler (e.g. cron) to archive the created log backup files. However, it must not be started, while another backup of this database to the external backup tool is already running.



Syntax of archive_stage

Syntax:

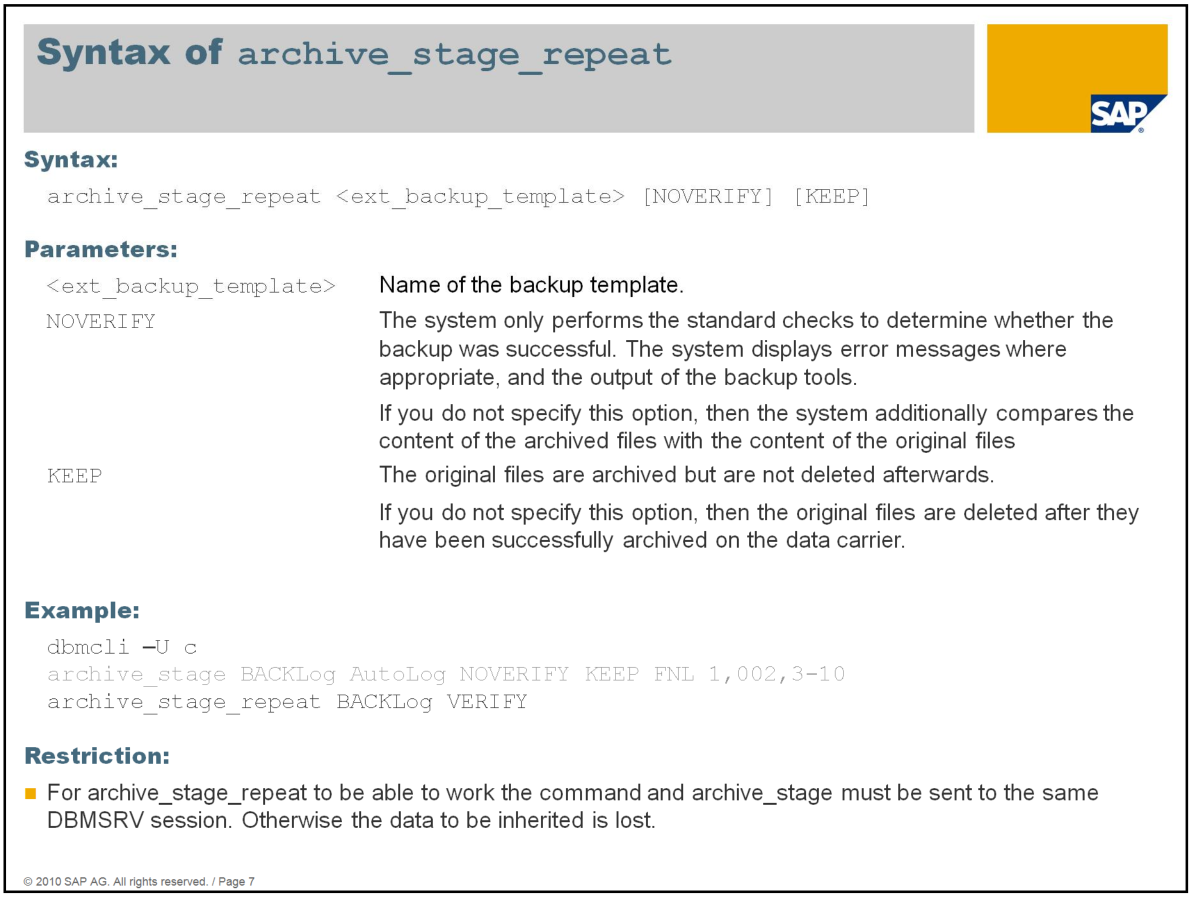
```
archive_stage <archive_medium> <log_medium> [NOVERIFY] [KEEP] [FNL <list>]
```

Parameter:

`<archive_medium>`	Name of the backup template to be used for archiving
`<log_medium>`	Name of the backup template with which the log backup has been created
`NOVERIFY`	The system only performs the standard checks to determine whether the copying was successful. The system displays error messages where appropriate, as well as the output of the backup tools. If you do not specify this option, then the system additionally compares the content of the archived files with the content of the original files.
`KEEP`	The original log backup files are archived but are not deleted afterwards. If you do not specify this option, then the original log backup files are deleted after they have been successfully archived on the data carrier.
`FNL <list>`	File Number List: Keyword for the list of files to be archived Only specify this keyword and the list of file numbers if you are sure that particular log backup files were created but are not listed in the backup history due to an error. If you do not specify a file list, the system copies all the log backup files that exist at the start of the archiving process, were created using the specified backup template, and are listed in the backup history

This slide explains the syntax of the archive_stage concept: We strongly recommend that you use the VERIFY option. After the versioned files have been transferred to the external backup tool, it allows you to retrieve these files and compare the data stream with the original files. Using this functionality, you can continuously test whether the external backup solution is functioning safely.

The DBMCLI command archive_stage must be scheduled using cron, at or a central planning calendar. This can also be done remotely using the remote capabilities of the DBMCLI.



Syntax of `archive_stage_repeat`

Syntax:

```
archive_stage_repeat <ext_backup_template> [NOVERIFY] [KEEP]
```

Parameters:

`<ext_backup_template>` Name of the backup template.

`NOVERIFY` The system only performs the standard checks to determine whether the backup was successful. The system displays error messages where appropriate, and the output of the backup tools.

If you do not specify this option, then the system additionally compares the content of the archived files with the content of the original files

`KEEP` The original files are archived but are not deleted afterwards.

If you do not specify this option, then the original files are deleted after they have been successfully archived on the data carrier.

Example:

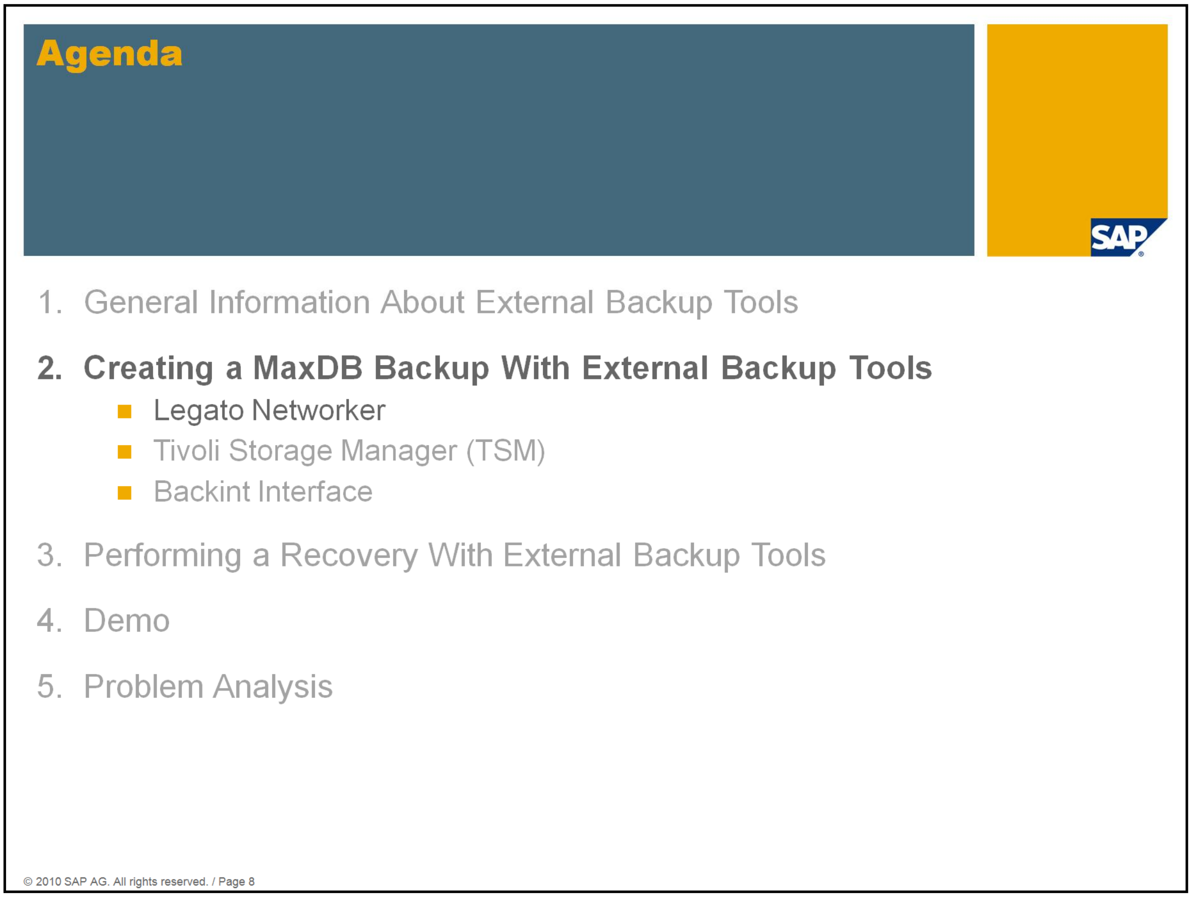
```
dbmcli –U c
archive_stage BACKLog AutoLog NOVERIFY KEEP FNL 1,002,3-10
archive_stage_repeat BACKLog VERIFY
```

Restriction:

- For archive_stage_repeat to be able to work the command and archive_stage must be sent to the same DBMSRV session. Otherwise the data to be inherited is lost.

With archive_stage_repeat, MaxDB makes it possible to send the versioned files to the external backup tool more than once. This enables you to send the files to different tapes or tape drives.

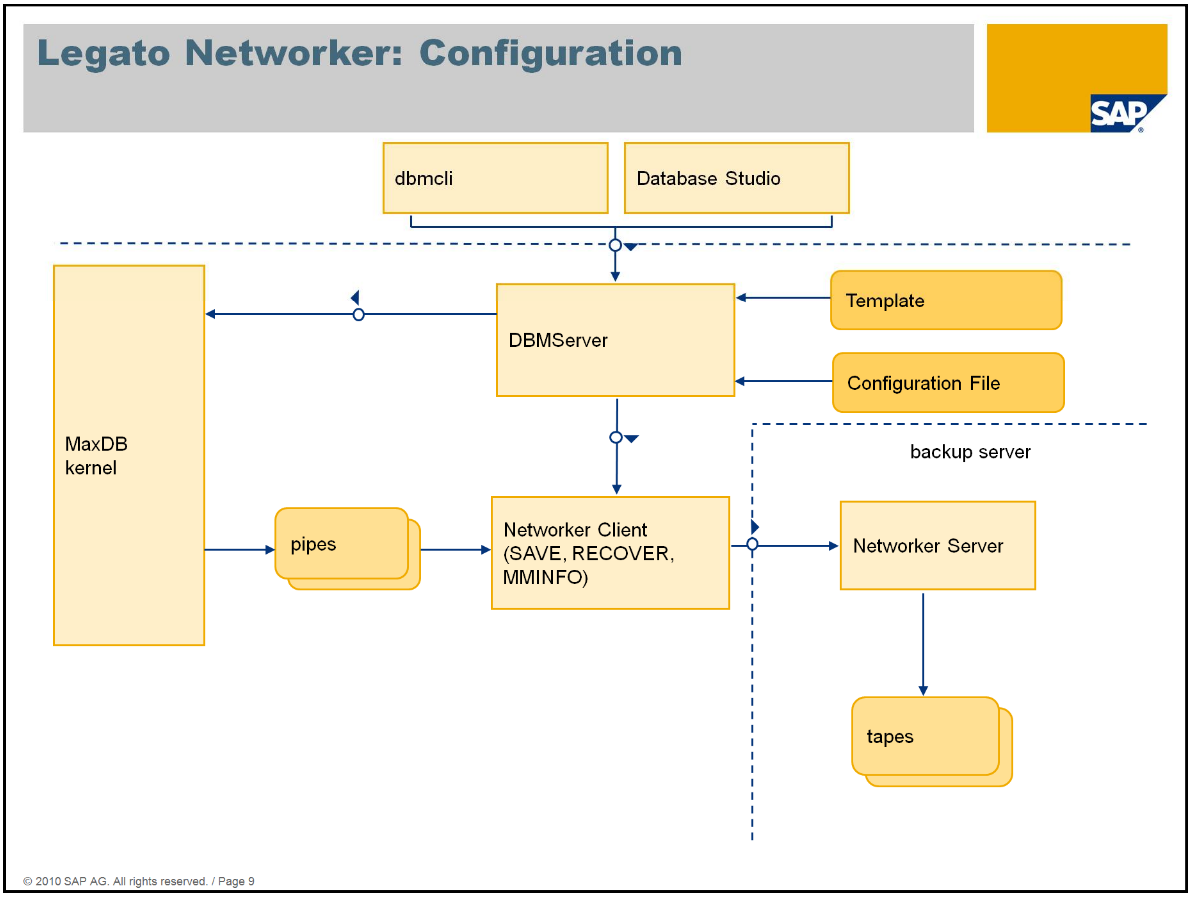
At present, the only restriction is that archive_stage_repeat must be run in the same DBM Server session in which archive_stage was previously executed.



Agenda

1. General Information About External Backup Tools
2. **Creating a MaxDB Backup With External Backup Tools**
   - Legato Networker
   - Tivoli Storage Manager (TSM)
   - Backint Interface
3. Performing a Recovery With External Backup Tools
4. Demo
5. Problem Analysis

The next chapter explains, how different external backup tools need to be configured, to be able to create a SAP MaxDB backup. At first the configuration of Legato Networker is explained.



Legato Networker: Configuration

dbmcli

Database Studio

MaxDB kernel

DBMServer

Template

Configuration File

backup server

pipes

Networker Client (SAVE, RECOVER, MMINFO)

Networker Server

tapes

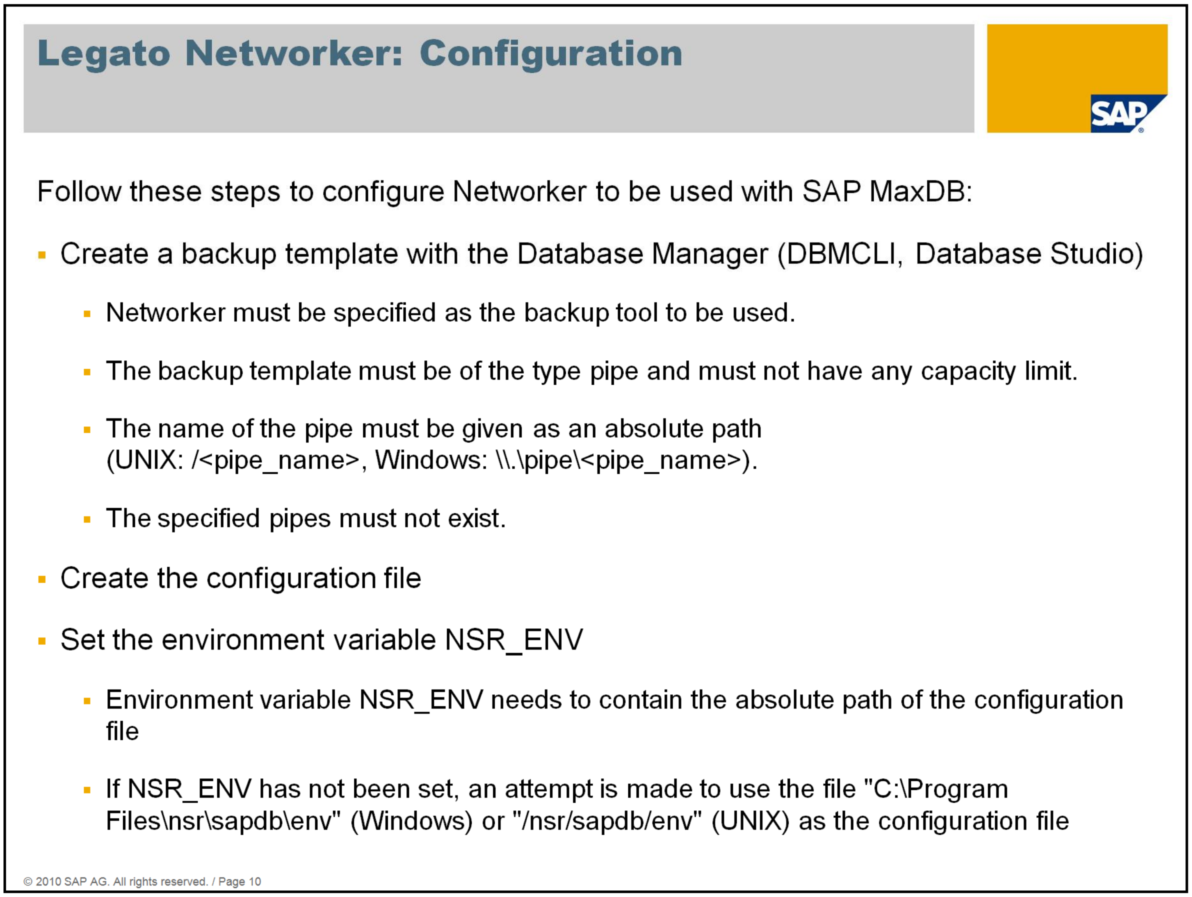
You can connect NetWorker using the command line clients SAVE, RECOVER and MMINFO that are included in the Networker software, or using the Backint for Oracle interface. The command line clients deliver better performance and this is the procedure described on the following slides.

The DBM Server uses the backup template definition to request the backup from the database kernel.

The database kernel opens the pipes sequentially.

The DBMServer uses the configuration file to request the NetWorker client to backup the data from the pipes as soon as the database kernel opens the first pipe.

The NetWorker Client opens the pipes, transfers the data to the backup server and writes the data to tape.



Legato Networker: Configuration

Follow these steps to configure Networker to be used with SAP MaxDB:

- Create a backup template with the Database Manager (DBMCLI, Database Studio)
  - Networker must be specified as the backup tool to be used.
  - The backup template must be of the type pipe and must not have any capacity limit.
  - The name of the pipe must be given as an absolute path (UNIX: /<pipe_name>, Windows: \\.\pipe\<pipe_name>).
  - The specified pipes must not exist.
- Create the configuration file
- Set the environment variable NSR_ENV
  - Environment variable NSR_ENV needs to contain the absolute path of the configuration file
  - If NSR_ENV has not been set, an attempt is made to use the file "C:\Program Files\nsr\sapdb\env" (Windows) or "/nsr/sapdb/env" (UNIX) as the configuration file

The absolute name of the pipe must be the same at backup and restore, due to NetWorker restrictions. Note also that linked files or directories are resolved by the NetWorker. For example, if you backup on a computer A via /tmp/nsr_pipe, and /tmp is only a link to /var/tmp, the system must be recovered with the restore on a computer B via the pipe /var/tmp/nsr_pipe.
This also results in special requirements for parallel backups: as a save/recover command is started for every pipe and NetWorker requires the exact name of the relevant backup pipe for the recovery, the template group must be configured for the recovery in exactly the same way as for the backup. This includes the sequence of the templates in the template group.

If you want to use NetWorker on Unix to create a database backup on one computer and to restore this backup in a second database on a second computer, both the databases must be run by operating system users with the same user ID. Otherwise NetWorker fails and consequently, MaxDB cannot restore the backup in the second database on the second computer.

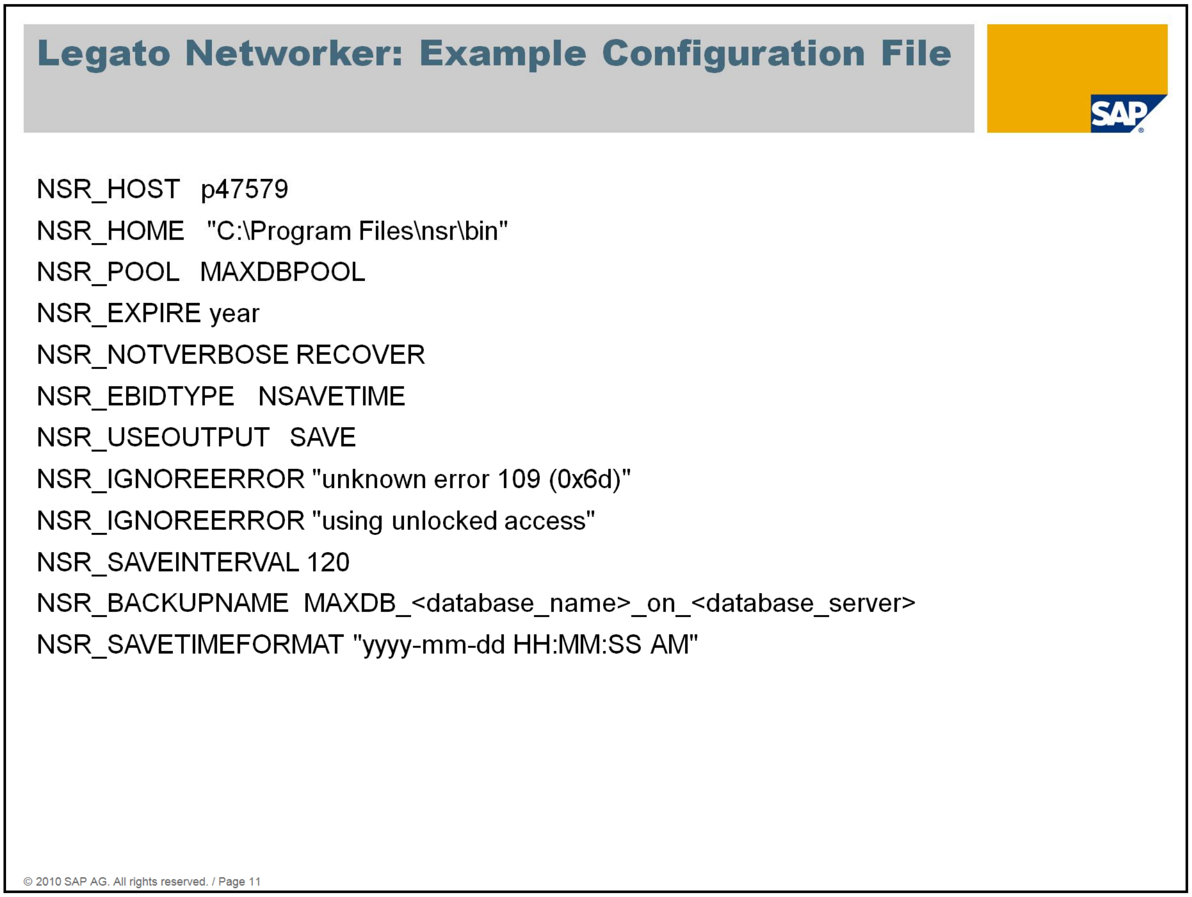
The value for environment variable NSR_ENV can be made known on the database server as follows:

```
dbmcli -d <database_name> -n <database_computer> -u <dbm_operator>,<dbm_operator_password>
dbm_configset -raw NSR_ENV <value>
```

<value>: Value of the environment variable NSR_ENV

Details regarding the usage of environment variables for the DBM Server can be found in the MaxDB documentation.

The timeout used by the DBM Server for monitoring NetWorker can be configured according to Note 600464.



Legato Networker: Example Configuration File

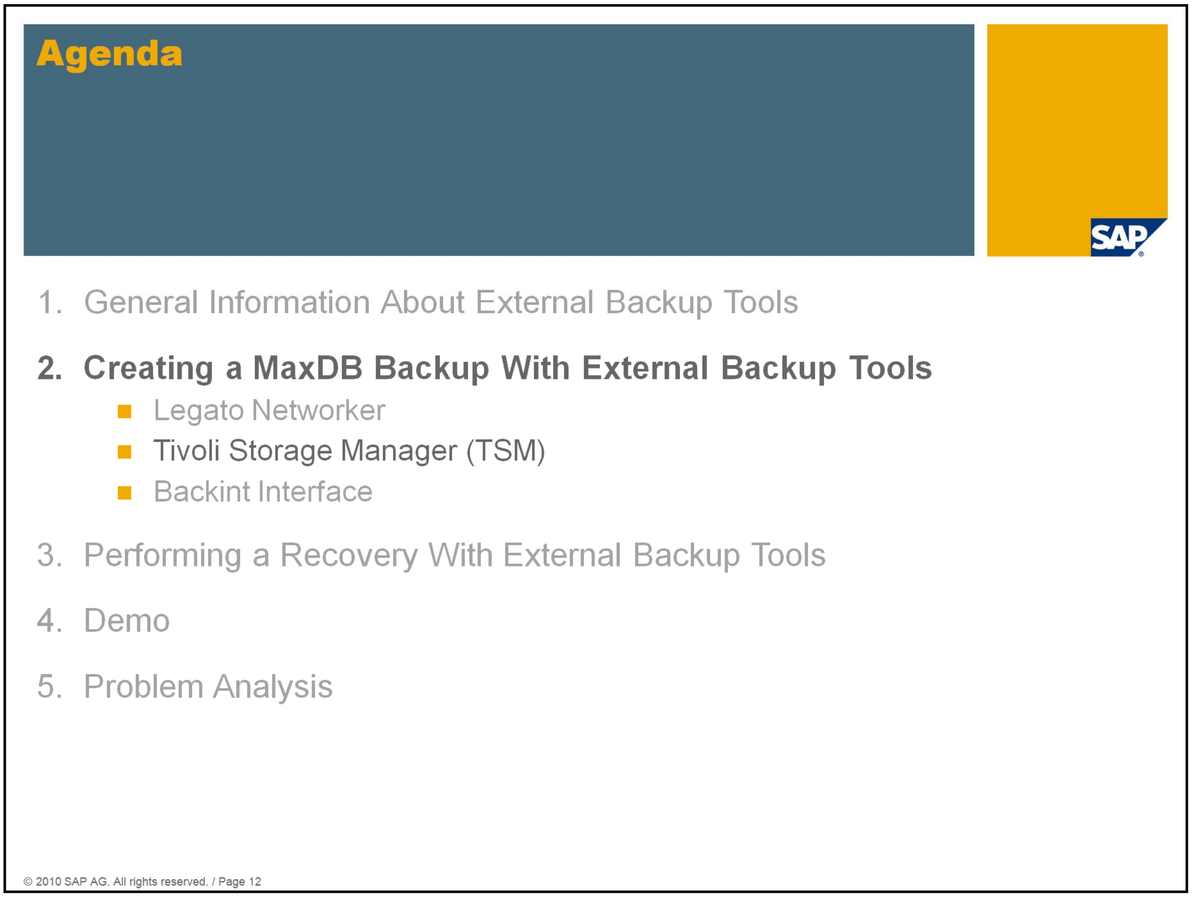
```
NSR_HOST   p47579
NSR_HOME   "C:\Program Files\nsr\bin"
NSR_POOL   MAXDBPOOL
NSR_EXPIRE year
NSR_NOTVERBOSE RECOVER
NSR_EBIDTYPE   NSAVETIME
NSR_USEOUTPUT   SAVE
NSR_IGNOREERROR "unknown error 109 (0x6d)"
NSR_IGNOREERROR "using unlocked access"
NSR_SAVEINTERVAL 120
NSR_BACKUPNAME  MAXDB_<database_name>_on_<database_server>
NSR_SAVETIMEFORMAT "yyyy-mm-dd HH:MM:SS AM"
```

In each line of the configuration file you can enter one parameter. For this, a key word must be given at the beginning of the line. Lines without one of the permissible key words are ignored. The key word is followed by the desired value of the relevant parameter. If this value contains a space character, the value must be set in quotes.

Detailed information about all possible parameters is available in the SAP MaxDB online documentation.

This configuration file specifies that

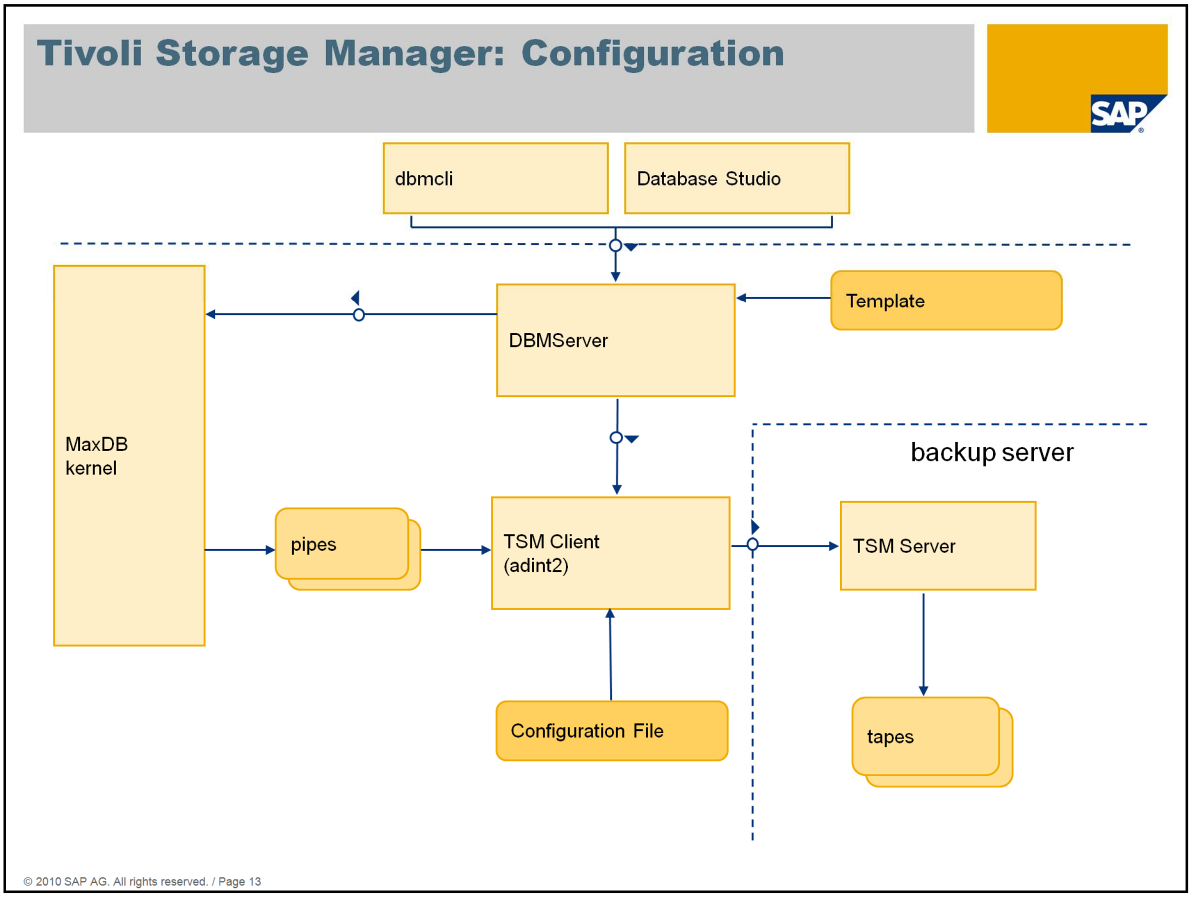
- the NetWorker server runs on the server p47579,
- the NetWorker client is installed on the database server under "C:\Program Files\nsr\bin",
- backups go into the NetWorker pool MAXDBPOOL,
- they have a NetWorker expiration time of one year,
- the NetWorker program recover is called up without the option –v,
- the backup time for the identification of the backups is used,
- the output of NetWorker program save is examined for errors and checked for a success message at every backup,
- the errors "unknown error 109 (0x6d)" and "using unlocked access" are ignored,
- with parallel backups, the SAVE commands are issued every 2 minutes
- the backups are stored under the name MAXDB_<database_name>_on_ <database_server> in NetWorker,
- it is expected that MMINFO puts out dates in the format "yyyy-mm-dd HH:MM:SS AM".



Agenda

1. General Information About External Backup Tools
2. **Creating a MaxDB Backup With External Backup Tools**
   - Legato Networker
   - Tivoli Storage Manager (TSM)
   - Backint Interface
3. Performing a Recovery With External Backup Tools
4. Demo
5. Problem Analysis

In the next part the configuration of Tivoli Storage Manager is explained.



Tivoli Storage Manager: Configuration

dbmcli

Database Studio

DBMServer

Template

MaxDB kernel

pipes

TSM Client (adint2)

Configuration File

backup server

TSM Server

tapes

You can connect TSM using the TSM client adint2 or using the Backint for Oracle interface.

With adint2, you achieve better performance. However, this program is not available on all platforms.

The adint2 program is not contained in a standard installation of the TSM client. The program is provided separately by IBM. For more information, see http://www.ibm.com

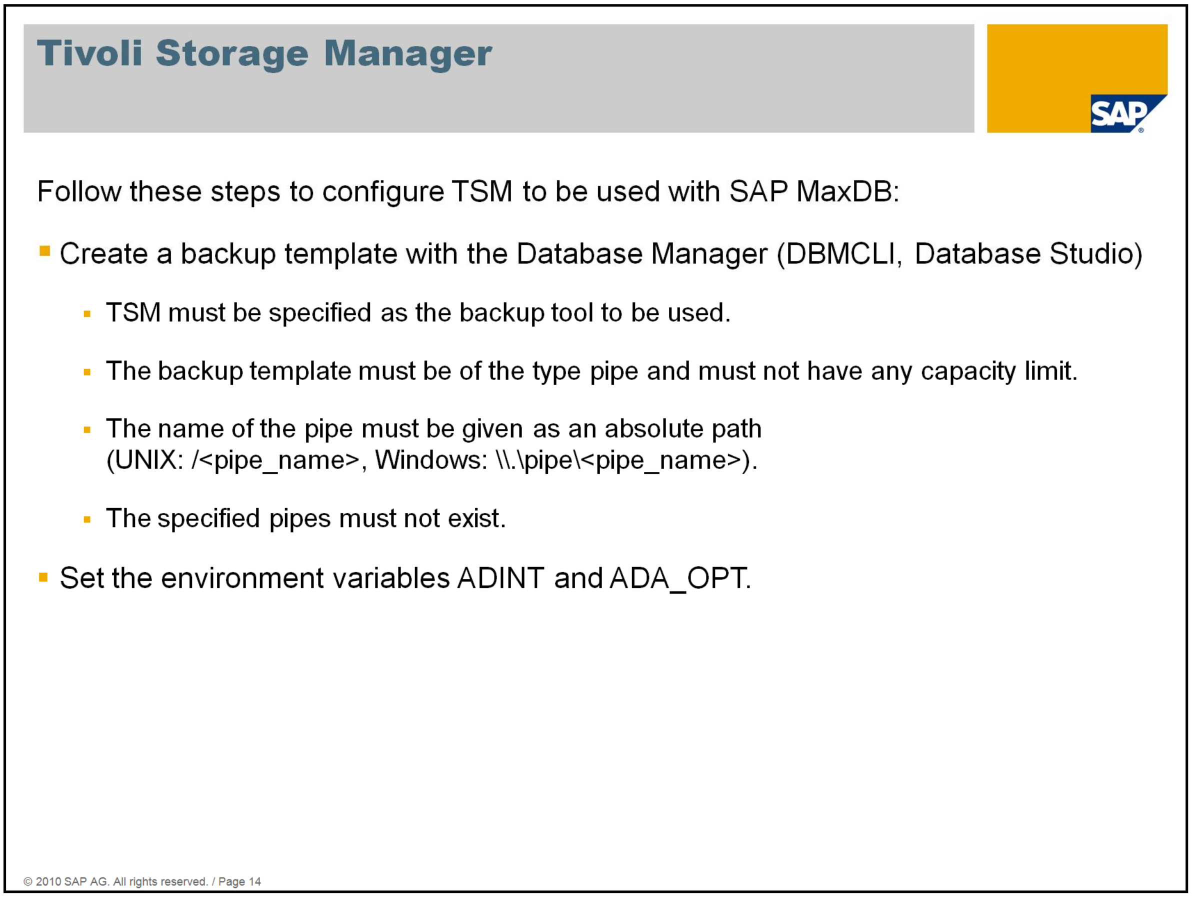
Using adint2 is the procedure described on the following slides.

The DBM Server uses the backup template definition to request the backup from the database kernel.

The database kernel opens the pipes sequentially.

The DBMServer requests adint2 to backup the data from the pipes as soon as the database kernel opens the first pipe.

Adint2 opens the pipes, transfers the data to the backup server and writes the data to tape.



Tivoli Storage Manager

Follow these steps to configure TSM to be used with SAP MaxDB:

- Create a backup template with the Database Manager (DBMCLI, Database Studio)
  - TSM must be specified as the backup tool to be used.
  - The backup template must be of the type pipe and must not have any capacity limit.
  - The name of the pipe must be given as an absolute path (UNIX: /<pipe_name>, Windows: \\.\pipe\<pipe_name>).
  - The specified pipes must not exist.
- Set the environment variables ADINT and ADA_OPT.

To carry out a backup/recovery to/from TSM, you must create a pipe-type backup template. In this template TSM must be specified as the backup tool to be used. Bear in mind that pipes on Windows have names of the form \\.\pipe\<pipe name> and that for a backup/recovery on UNIX, the specified pipe must not exist in the file system.

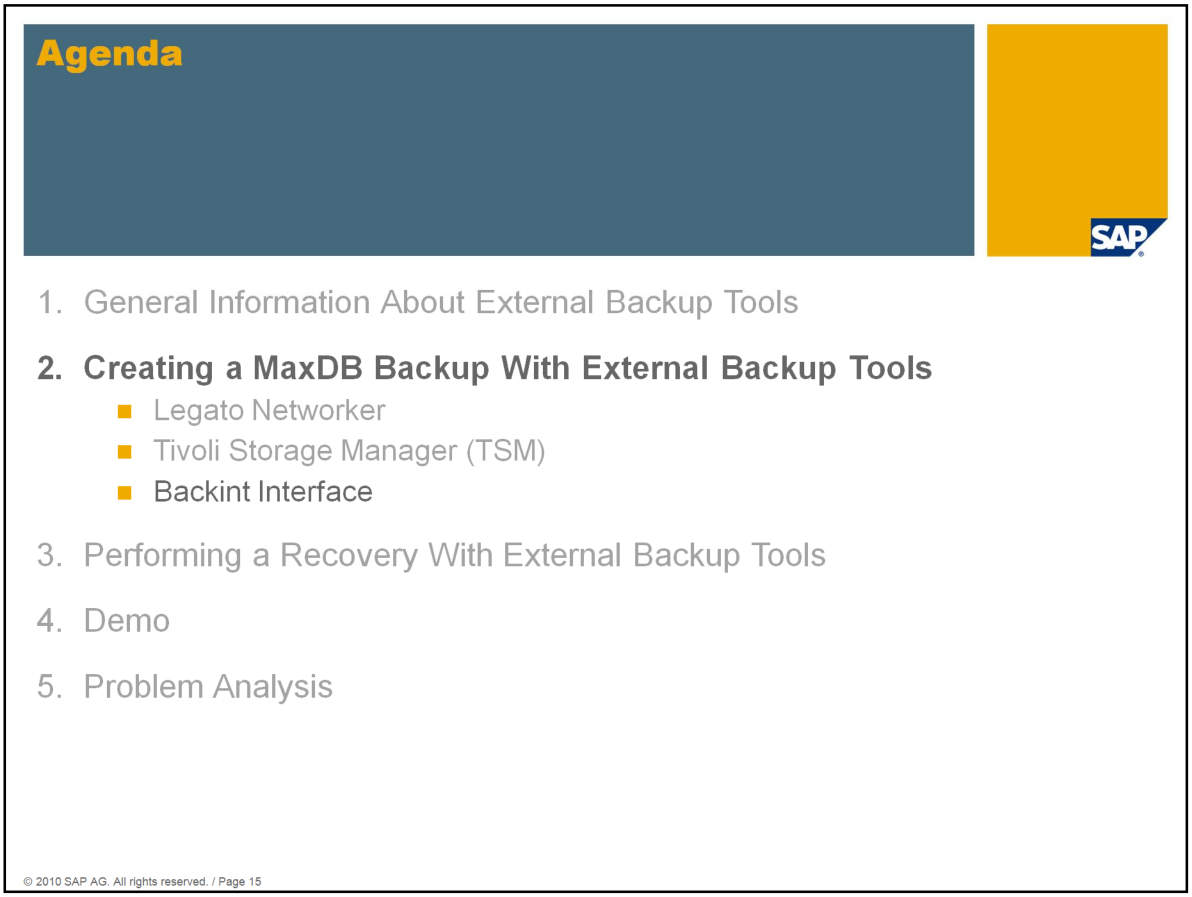
To enable the Database Manager on the database computer to access the required environment variables ADINT and ADA_OPT, you must define these variables using Database Manager CLI.

ADINT: Specifies the directory in which the program adint2 (Unix) or adint2.exe (Microsoft Windows) is stored. Specify this directory as an absolute path without a closing slash (/) or backslash (\).

ADA_OPT: Absolute name of the adint2 configuration file. An example configuration file initSID.utl is located in the directory of the adint2 program.

```
dbmcli -d <database_name> -n <database_computer> -u <dbm_operator>,<dbm_operator_password>
dbm_configset -raw <variable> <value>
```

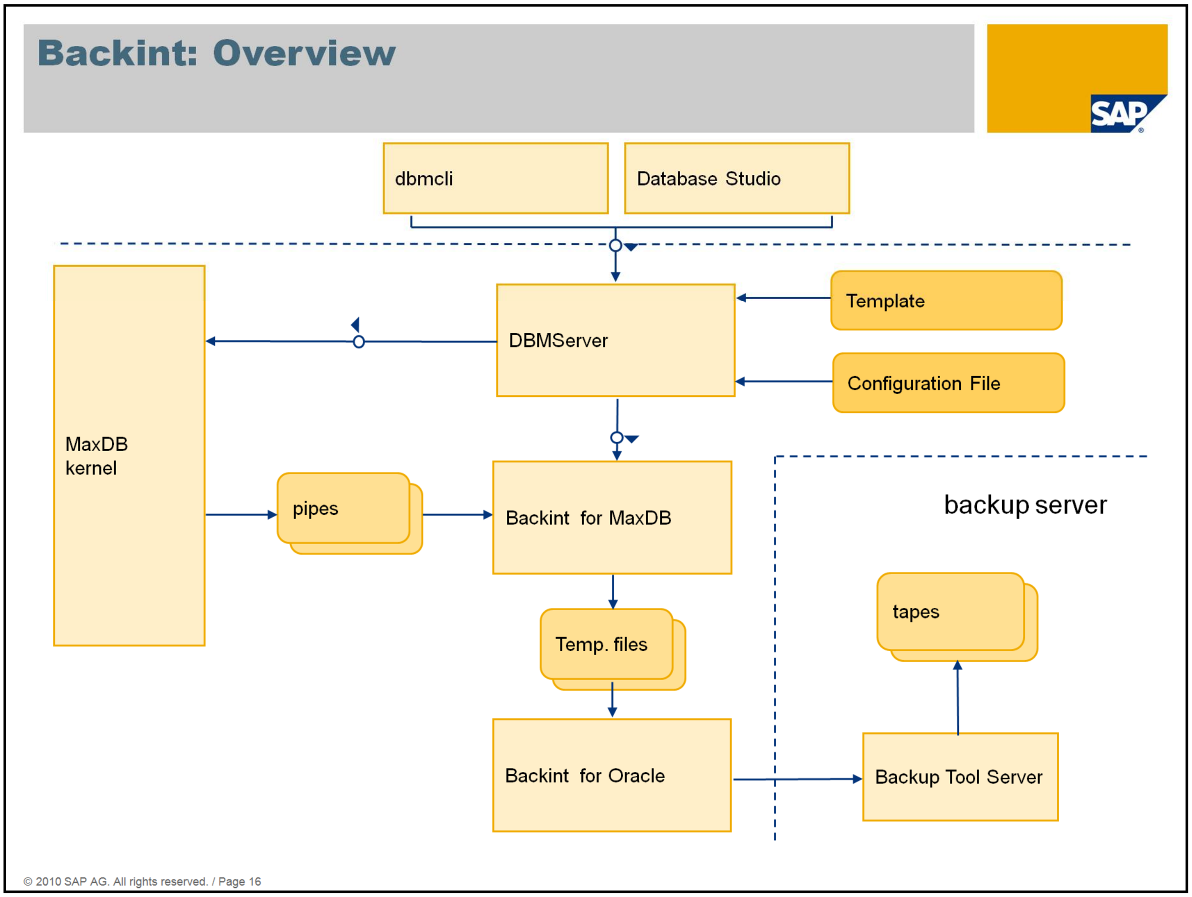
You can configure the timeout used by the DBM Server to monitor adint2 in accordance with note 600464.



Agenda

1. General Information About External Backup Tools
2. **Creating a MaxDB Backup With External Backup Tools**
   - Legato Networker
   - Tivoli Storage Manager (TSM)
   - Backint Interface
3. Performing a Recovery With External Backup Tools
4. Demo
5. Problem Analysis

In the next part the Backint interface is explained. Backint is the short form of *Backup Interface* and is the genus for two backup tool interfaces – Backint for Oracle and Backint for MaxDB. Both interfaces define a command line tool, that can be used by database management tools like MaxDB's DBM Server to backup and recover data into or from a backup tool.



Backint: Overview

Many third-party backup tools have a Backint for Oracle interface. In principle, the program Backint for Oracle can backup any files in the file system, regardless of whether the Oracle database system is installed on the computer or not. To connect Backint for Oracle to SAP MaxDB, SAP provides an adapter program with SAP MaxDB which implements the Backint for MaxDB specification. Backint for Oracle actually saves the backup data using the backup tool onto a permanent backup medium. To integrate Backint for Oracle, you therefore need both Backint for MaxDB contained in the delivery of SAP MaxDB and Backint for Oracle provided by the backup tool manufacturer. The adapter program (Backint for MaxDB) was implemented to enable backups via pipes to external backup tools which don‘t support pipes as input media.

The backup procedure using the Backint for Oracle interface is controlled by the DBMServer:

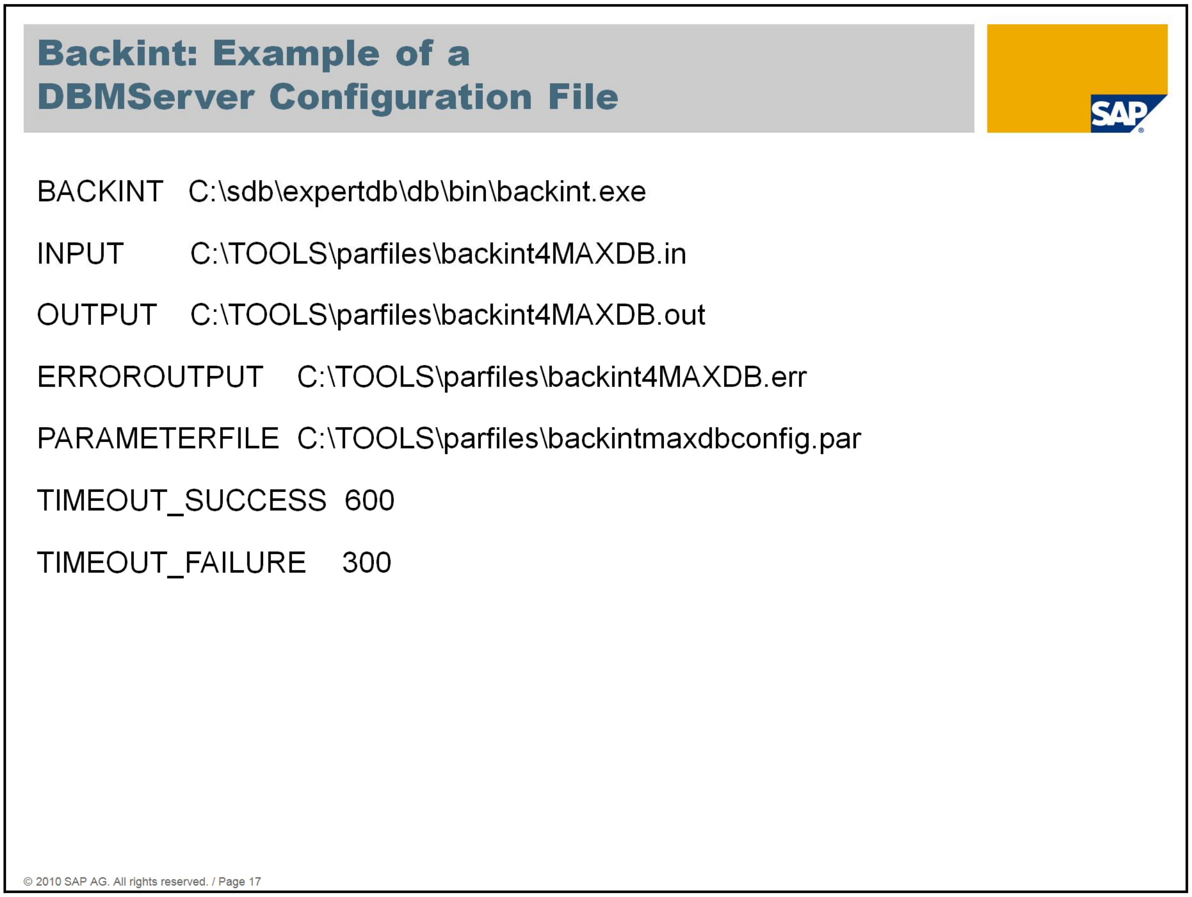
- The DBM Server uses the backup template definition to request the backup from the database kernel.
- The database kernel opens the pipes sequentially.
- The DBMServer requests Backint for MaxDB to backup the data from the pipes as soon as the database kernel opens the first pipe.
- Backint for MaxDB opens the pipes, transfers the data to temporary files (of configurable size) and requests Backint for Oracle to backup these files.
- Backint for Oracle transfers the temporary files to the backup server.
- Backint for MaxDB deletes the temporary files.

If necessary, the procedure of creating and saving temporary files is repeated until all backup data has been processed.

To be able to control this procedure, the DBMServer needs the backup template definition and a configuration file, which contains e.g. the information, where the Backint for MaxDB can be found.

The configuration is carried out with the following steps:

- Use the database manager (DBMCLI or Database Studio) to create a backup template. The Backup Tool has to be specified as Backint and the pipes used must not exist.
- Set the environment variable BSI_ENV so that it contains the path of the configuration file. If you do not set BSI_ENV, the system looks for the bsi.env file in the run directory of the database.
- Create the configuration file.



Backint: Example of a DBMServer Configuration File

```
BACKINT   C:\sdb\expertdb\db\bin\backint.exe
INPUT       C:\TOOLS\parfiles\backint4MAXDB.in
OUTPUT    C:\TOOLS\parfiles\backint4MAXDB.out
ERROROUTPUT    C:\TOOLS\parfiles\backint4MAXDB.err
PARAMETERFILE  C:\TOOLS\parfiles\backintmaxdbconfig.par
TIMEOUT_SUCCESS  600
TIMEOUT_FAILURE    300
```

In our demo scenario this configuration file is called bsi.env and it is located in directory C:\TOOLS\parfiles. Therefore the environment variable BSI_ENV has been set to this directory with the DBMCLI command *dbm_configset –raw BSI_ENV “C:\TOOLS\parfiles\bsi.env”*.

In every line of the configuration file, you can specify one parameter. To do this, you must specify a keyword at the beginning of the line. Lines without a key word will be ignored.

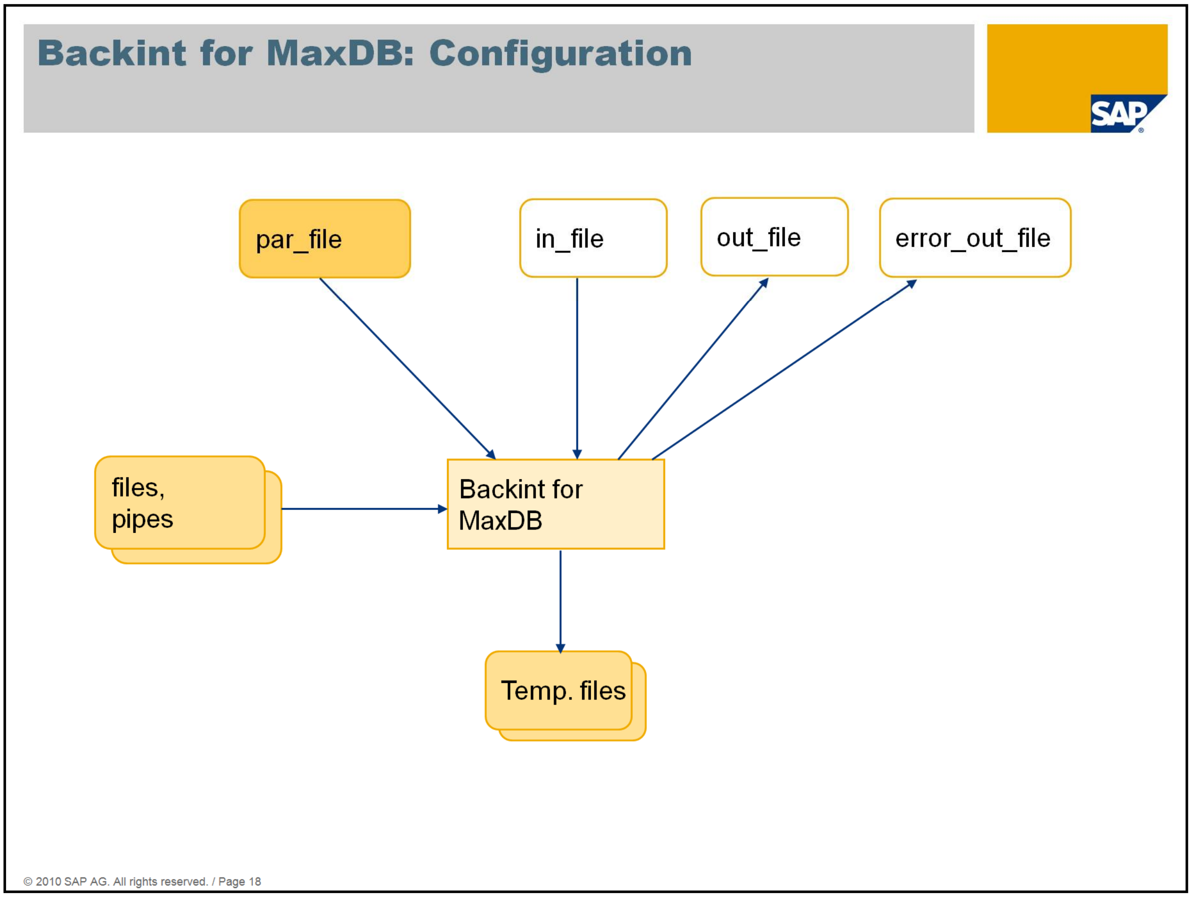
Detailed information about all possible parameters is available in the SAP MaxDB online documentation.

This example configuration file determines that the database manager calls the tool C:\sdb\expertdb\db\bin\backint.exe.

The standard input, standard output and standard error output files for Backint for MaxDB are created by the DBMServer as C:\TOOLS\parfiles\backint4MAXDB.in, C:\TOOLS\parfiles\backint4MAXDB.out or C:\TEMP\backint4MAXDB.err files.

The Backint program uses the parameter file C:\TOOLS\parfiles\backintmaxdbconfig.par.

If the process is successful the database manager will wait for a maximum of 10 minutes for the end of Backint for MaxDB. In case of an error, it will only wait for 5 minutes.



Backint for MaxDB: Configuration

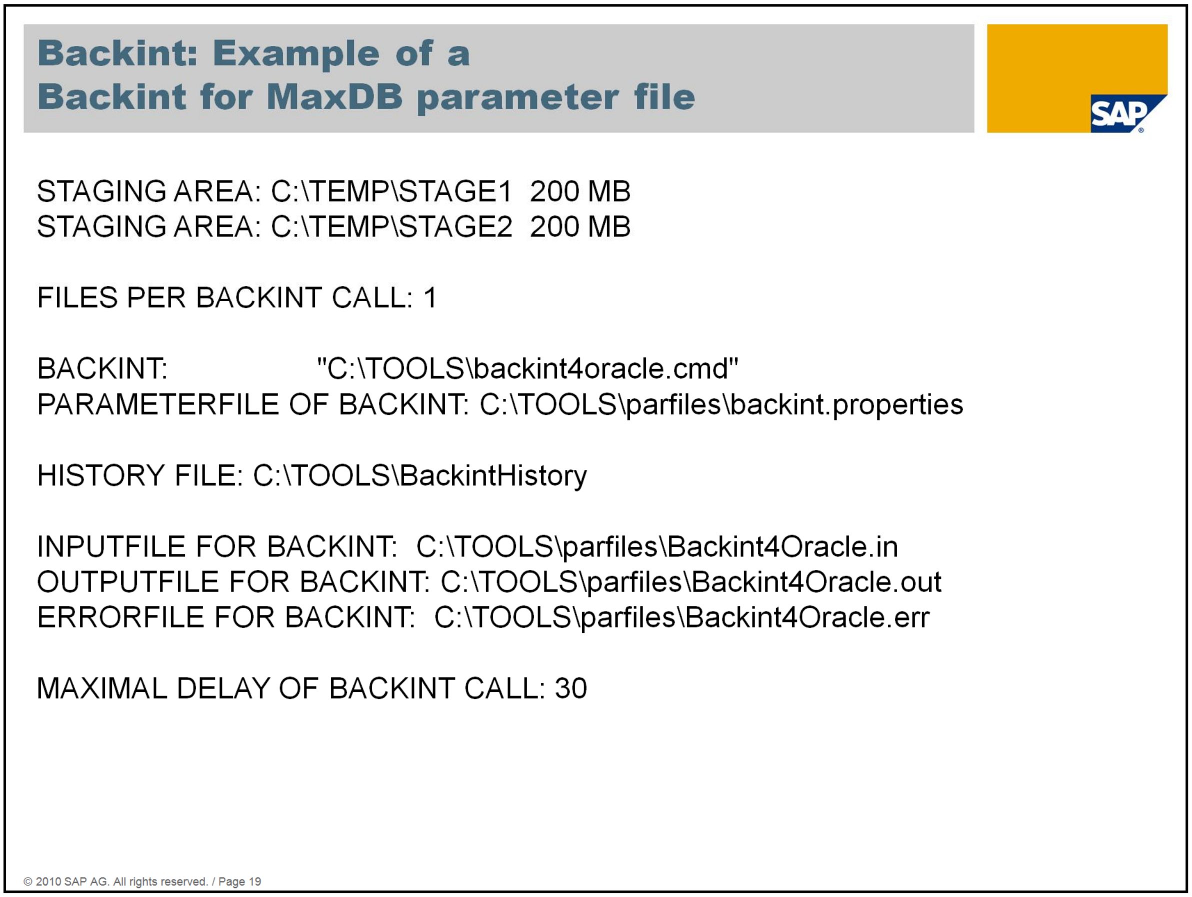
The backup data of the database system is received by one or more pipes and saved in temporary files. Then Backint for MaxDB calls the third-party backup tool and saves the temporary files by using the Backint for Oracle program.

The names of the files and/or pipes to be backed up are specified by the DBMServer in the in_file.

The number and size of the temporary files to be created have to be specified by the administrator in the parameter file of the Backint for MaxDB. Additionally the location of the Backint for Oracle program needs to be specified in this parameter file.

The in_file, out_file and the error_out_file are created by Backint for MaxDB and are used by the DBMServer to provide information about a successful or failed backup in the DBMServer log files.

To speed up the backup process, it is possible to enable parallel backups. To do this, you define a group of parallel data carriers and a separate staging area for each pipe.



Backint: Example of a Backint for MaxDB parameter file

```
STAGING AREA: C:\TEMP\STAGE1  200 MB
STAGING AREA: C:\TEMP\STAGE2  200 MB

FILES PER BACKINT CALL: 1

BACKINT:                "C:\TOOLS\backint4oracle.cmd"
PARAMETERFILE OF BACKINT: C:\TOOLS\parfiles\backint.properties

HISTORY FILE: C:\TOOLS\BackintHistory

INPUTFILE FOR BACKINT:  C:\TOOLS\parfiles\Backint4Oracle.in
OUTPUTFILE FOR BACKINT: C:\TOOLS\parfiles\Backint4Oracle.out
ERRORFILE FOR BACKINT:  C:\TOOLS\parfiles\Backint4Oracle.err

MAXIMAL DELAY OF BACKINT CALL: 30
```

This is an example for the parameter file. In our example it is called C:\TOOLS\parfiles\backintmaxdbconfig.par.

Define an attribute in each row of the parameter file. For this, you have to specify defined keywords at the beginning of the line. The system ignores rows that do not begin with keywords.

Detailed information about all possible parameters is available in the SAP MaxDB online documentation.

This example configuration file specifies that the adapter program can save or restore a maximum number of two pipes at the same time. The temporary files generated in this process in each case reach a size of 200 MB as long as the end of the pipes is not reached.

It is also possible to specify the size of these files in Byte (default, if no unit is specified), KB or GB.

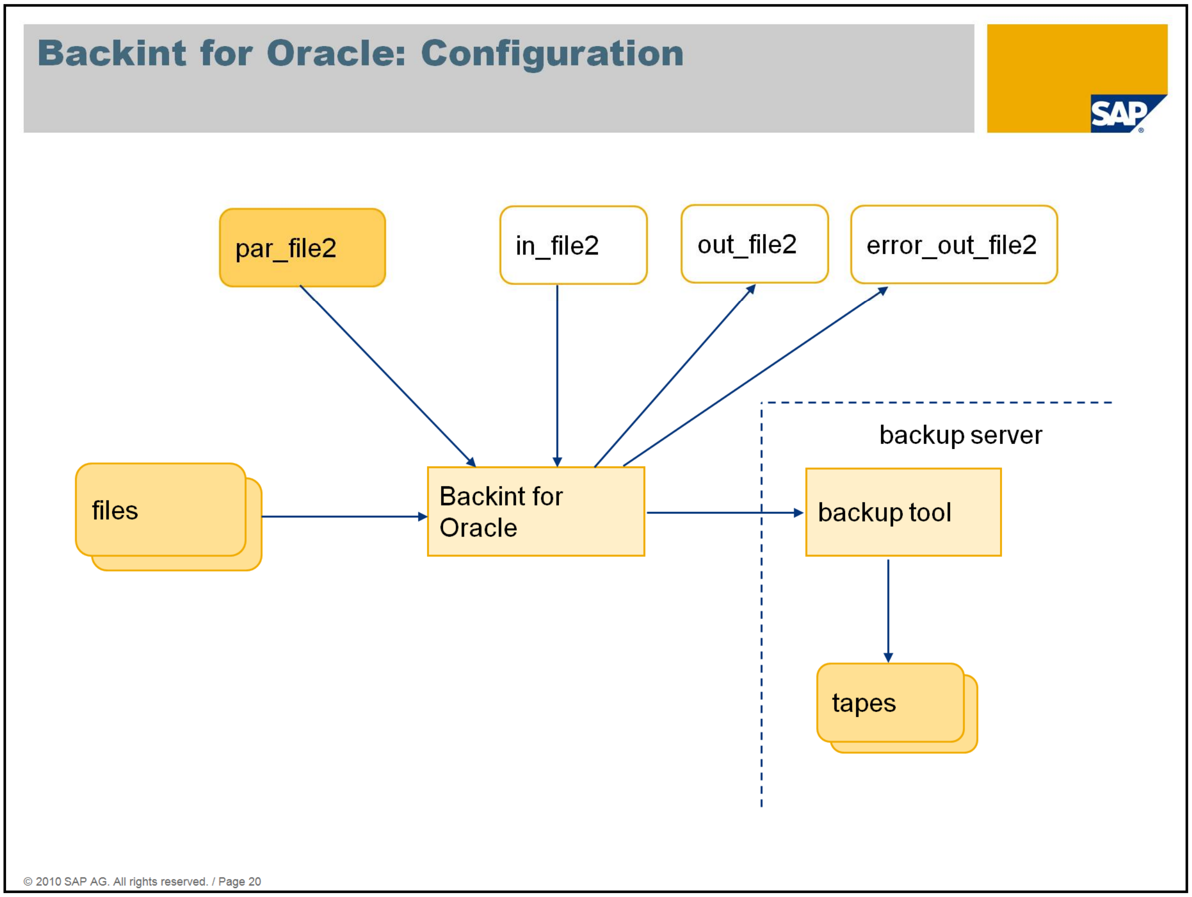
If one of these files is fully created, this is saved using the Backint for Oracle: "C:\TOOLS\backint4oracle.cmd".

The Backint for Oracle program uses the parameter file C:\TOOLS\parfiles\backint.properties.

Upon completion of a backup, the adapter program stores in the history file C:\TOOLS\BackintHistory, which temporary files belong to which pipe. This information is required in case of a recovery.This history file is then saved with another Backint for Oracle call.

The standard input, standard output and standard error output files are C:\TOOLS\parfiles\Backint4Oracle.in, C:\TOOLS\parfiles\Backint4Oracle.out and C:\TOOLS\parfiles\Backint4Oracle.err.

If the temporary files are created at different speeds and one temporary file is already complete, the adapter program waits a maximum of 30 seconds for one of the temporary files to be fully created. If none of the other files is completed within 30 seconds, the existing temporary file is backed up using Backint for Oracle.



Backint for Oracle: Configuration

par_file2

in_file2

out_file2

error_out_file2

backup server

files

Backint for Oracle

backup tool

tapes

As already explained, Backint for Oracle is only able to backup files – no data from pipes. Therefore it cannot be used without the Backint for MaxDB interface to create a backup of a MaxDB database.

Backint for Oracle is implemented by the backup tool vendor – not by SAP. It is available for a large number of backup tools.

The needed configuration files are similar to those needed for Backint for MaxDB:

Backint for Oracle learns from the par_file2, where the backup tool can be found and how to use it. The format of this configuration file is described by the backup tool vendor. Some Backint for Oracle implementations do not need such a configuration file (but use environment variables instead).

Backint for Oracle learns from the in_file which files (or raw devices) must be backed up.

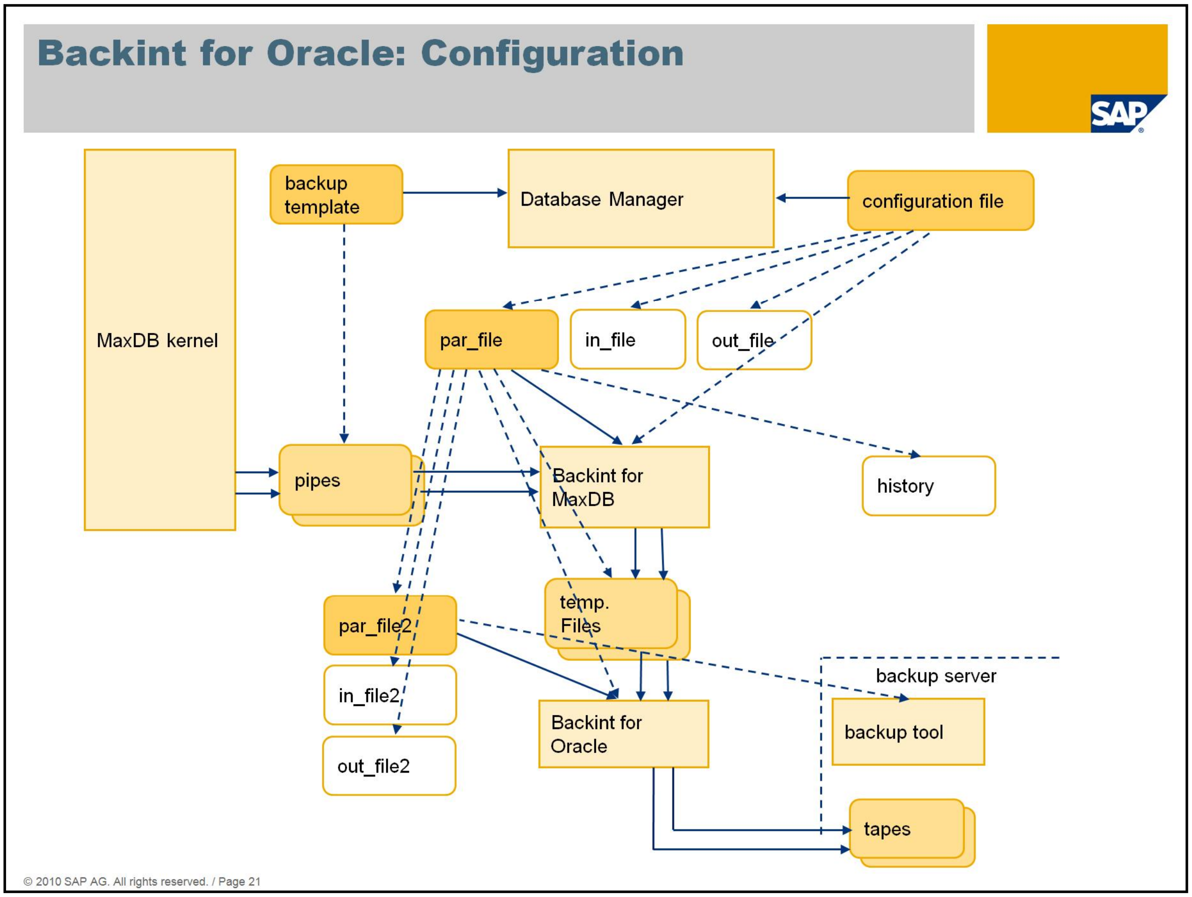
Backint for Oracle reads the data and transfers it to the backup tool.

The backup tool writes the data to tapes.

Backint for Oracle reports to the out_file which files could be backed up successfully.

For successfully backed up files Backint for Oracle supplies a Backup Identifier (BID), but only the combination of user, file name and BID is identifying the backed up version of a file unambiguously

Backint for Oracle supplies a return code following the rule: 0 for success, 1 for success with warnings and 2 for errors.



Backint for Oracle: Configuration

After seeing the configuration of the two Backint programs separately, this slide shows the complete configuration. You generally have to create three configuration files: a configuration file for the Database Manager, a parameter file for the Backint for MaxDB (par_file) and a parameter file for Backint for Oracle (par_file2).

The DBMServer learns the name of the pipes and that Backint for MaxDB should be used from the backup template.

From the configuration file the DBMServer learns where to find Backint for MaxDB, where to find the parameterfile par_file of Backint for MaxDB and which files must be used as in_file and out_file (and error_out file) for Backint for MaxDB.

From the file par_file Backint for MaxDB learns, where to find Backint for Oracle, where to find the parameter file par_file2 of Backint for Oracle and which files must be used as in_file2 and out_file2 (and error_out2 file) for Backint for Oracle. It also learns where to find its own history file.

The DBMServer supplies the names of the pipes to Backint for MaxDB via the file in_file and interprets the Backint for MaxDB output via the out_file.

Backint for MaxDB reads the backup data from the pipes and writes it to temporary files, after that it calls Backint for Oracle.

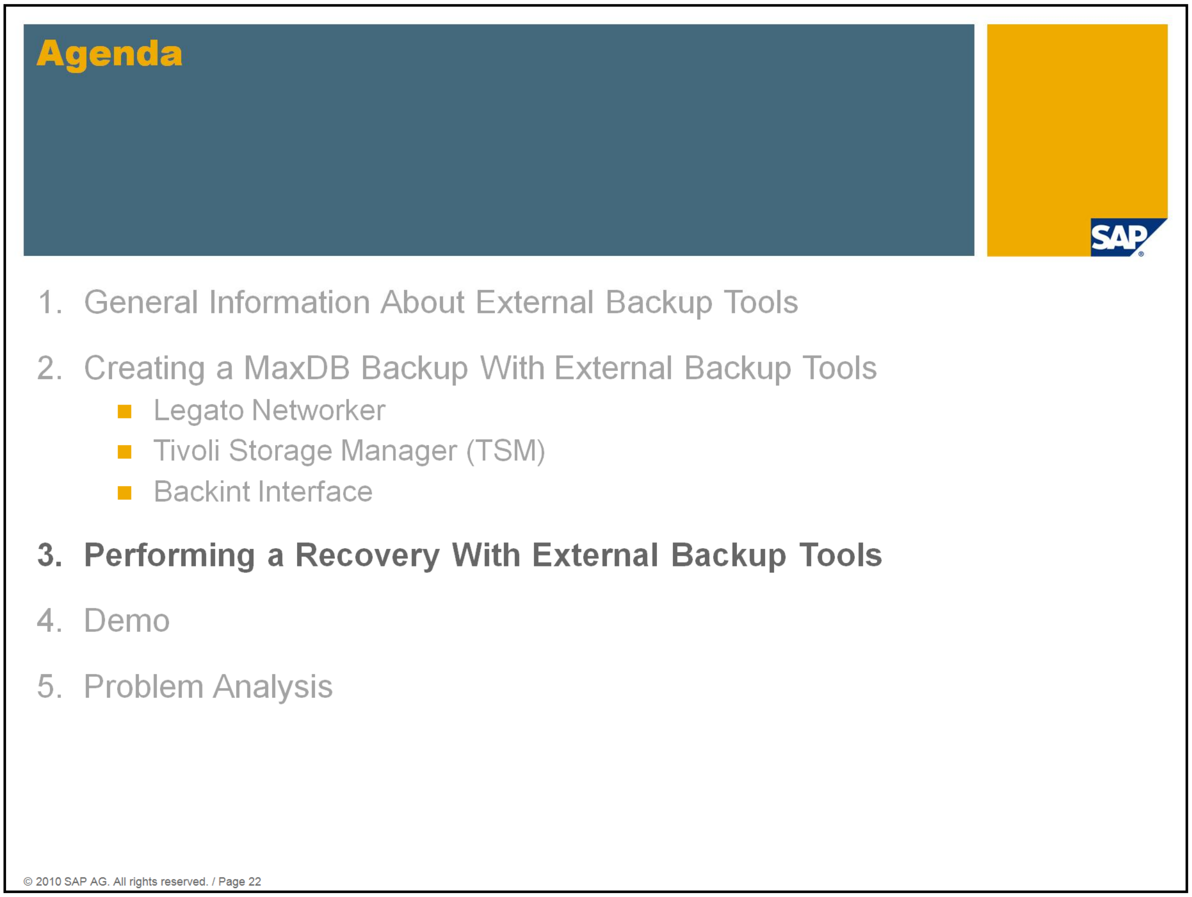
From its configuration file par_file2 Backint for Oracle learns where to find the backup tool and how to use it. The format of this configuration file is described by the backup tool vendor. Some Backint for Oracle implementations do not need such a configuration file (but use environment variables instead).

Backint for MaxDB supplies the names of the temporary files to Backint for Oracle via the file in_file2 and interprets the Backint for Oracle output via the out_file2.

Backint for Oracle backs up those temporary files with the help of the backup tool to tapes.

Backint for MaxDB deletes the temporary files.

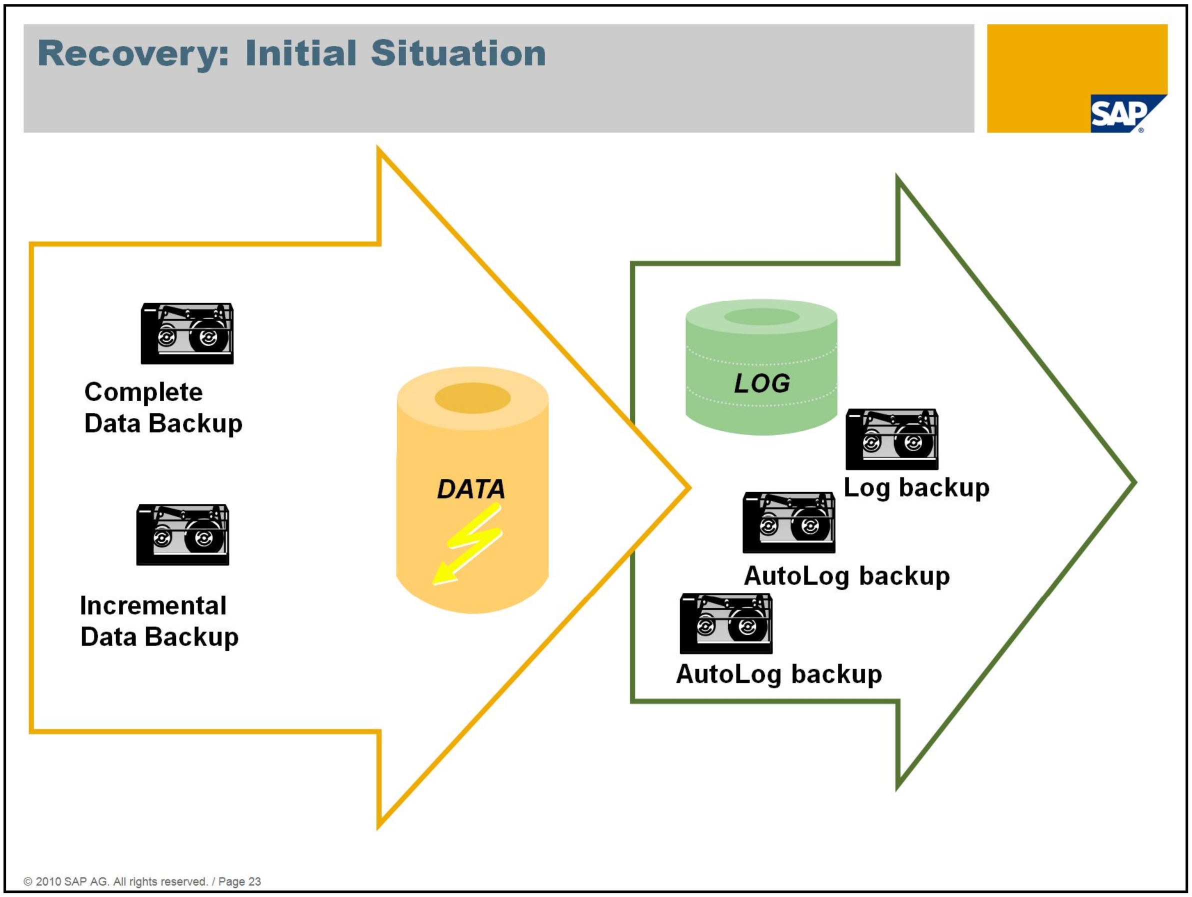
These steps are repeated until all data of the pipes is backed up to tapes.



Agenda

1. General Information About External Backup Tools
2. Creating a MaxDB Backup With External Backup Tools
   - Legato Networker
   - Tivoli Storage Manager (TSM)
   - Backint Interface
3. **Performing a Recovery With External Backup Tools**
4. Demo
5. Problem Analysis

It is essential for your system‘s safety to be able to create backups successfully. However, it is just as important to be able to restore such a backup. The next chapter therefore explains how the restore of a backup created with external backup tools works.



Recovery: Initial Situation

Complete Data Backup

Incremental Data Backup

DATA

LOG

Log backup

AutoLog backup

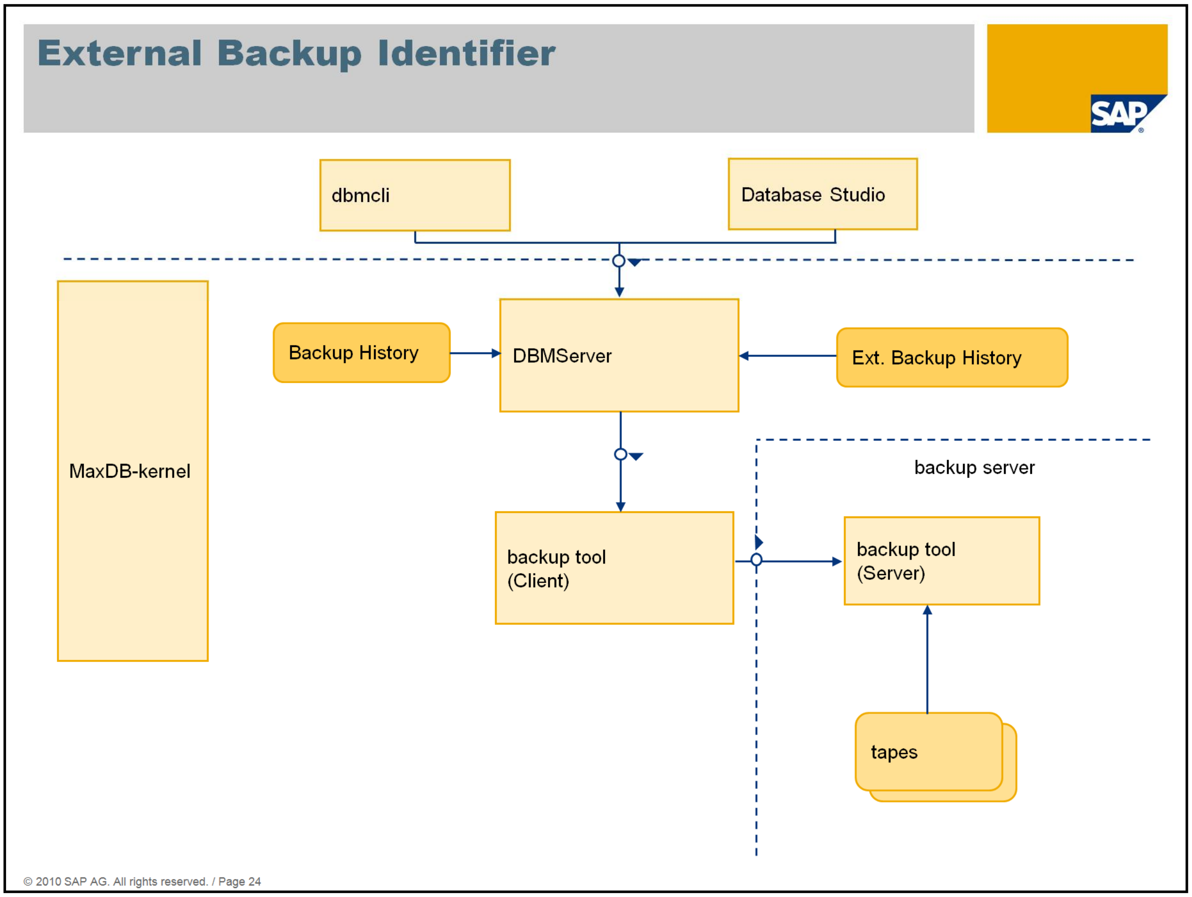
AutoLog backup

If you follow SAP's recommendations for the disk configuration of your database instances and the backup strategy, the current log entries and at least four backup generations are always available for you to restore the content of the database instance if problems occur. It is then very unlikely that you will lose any data.

If a data volume sustains physical damage, a complete database recovery needs to be performed. The basis for this type of recovery are normally the complete and incremental data backups as well as log backups of the latest backup generation.

If a logical error occurs in the SAP system, making it necessary to reset the system to a previous state, you also do this by performing a database recovery using a complete data backup and then importing incremental data and log backups. The administrator can specify whether all available log information is to be recovered up to the most recent point in time possible, or only up to a specific time in the past without the most recent transactions.

To ensure you are well prepared for a recovery, we recommend that you train at least two employees to regularly test a complete database recovery using the backups from the production system. For these tests, you require a test server comparable to the database server. This could, for example, be your quality assurance system.



External Backup Identifier

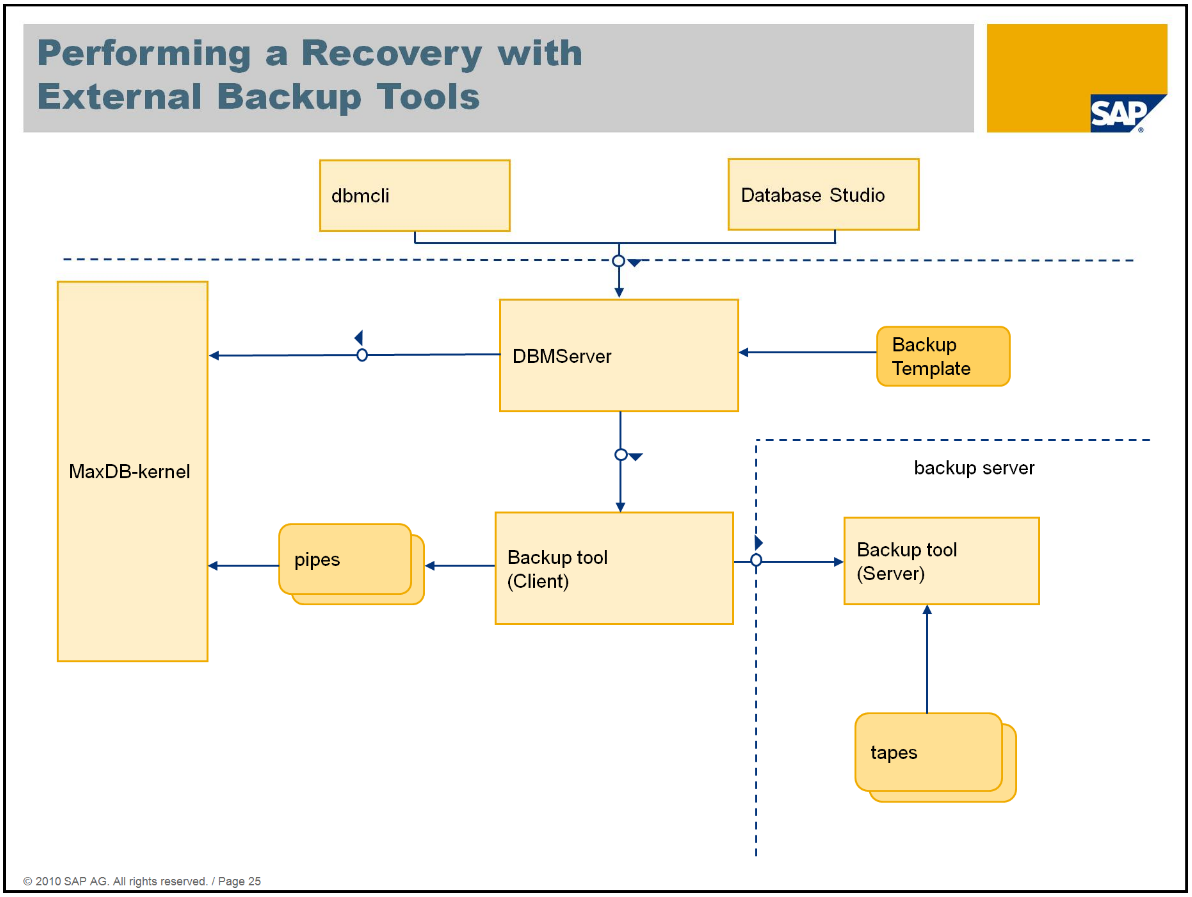
Every time a backup is created with an external backup tool, the backup tool creates a so called external backup identifier (EBID) to identify this specific backup. This EBID is needed, when such a backup should be restored. Independent of the external backup tool used, the EBIDs of available backups can be obtained with the DBMCLI commands backup_ext_ids_get and backup_ext_ids_list. With the first command, the DBMServer inquires the backup tool regarding available backups. The resulting list of available backups can be requested via the backup_ext_ids_list and the backup_ext_ids_listnext command. Database Studio also displays the information about the External Backup IDs of the backup in the backup history under *Details* in addition to the media information.

The database kernel is not required to determine the data. This is determined by a communication of the DBMServer with the external backup tool. Depending on the used external backup tool, the specific client tool of this backup tool is used to get the required information – e.g. for backint function INQUIRE is used, for Networker tool mminfo is used.

The different backup tools create different external backup IDs. Examples for EBIDs are:

- P47579_DB72_2001.03.30_15.51.20_SAVEDATA_ADSM
- NST 985877420 P47579
- DB72 985963853 \\.\pipe\b1

External Backup IDs may contain spaces. If the External Backup Identifier contains spaces, it must be put in quotes ("<ExtBackupID>") when used in DBMCLI commands.



Performing a Recovery with External Backup Tools

dbmcli

Database Studio

DBMServer

Backup Template

MaxDB-kernel

backup server

pipes

Backup tool (Client)

Backup tool (Server)

tapes

The DBM Server controls and manages the recovery process.

The DBM Server uses the backup template to identify the external backup tool to be used and then determines how to proceed.

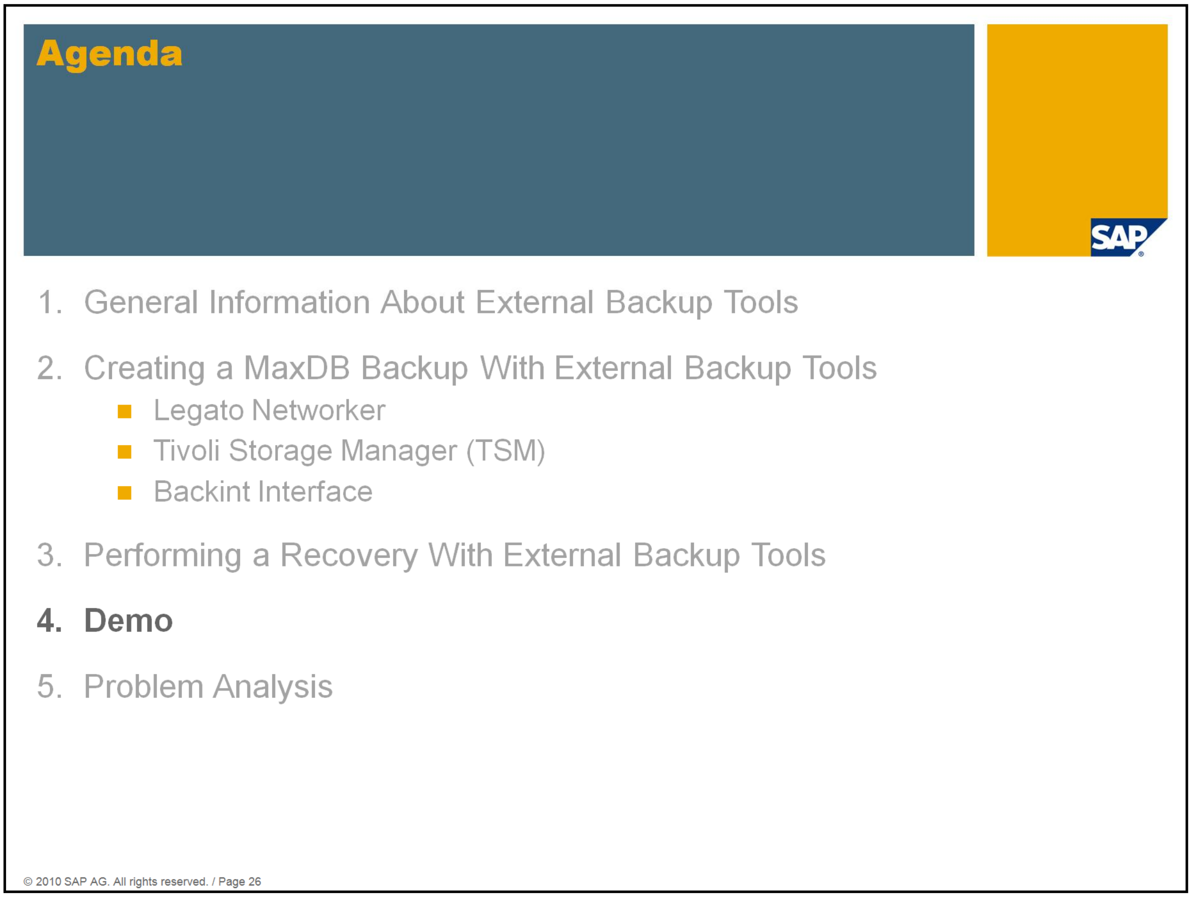
The DBM Server sends the recovery command to the database kernel, which then sequentially opens the required pipes.

When the database kernel attempts to open the first pipe, the DBM Server starts the client of the backup tool. The list of External Backup IDs provided via the recover_start command is used to request the correct backups from the backup tool.

The backup tool provides the required backup data in the pipes. The database kernel receives the data and distributes it across the data volumes.

After the action is completed, the DBM Server interprets the response of the database kernel and the return code of the backup tool and reports the result of the recovery attempt.

For more information (such as which DBMCLI commands are used for this), see http://maxdb.sap.com.



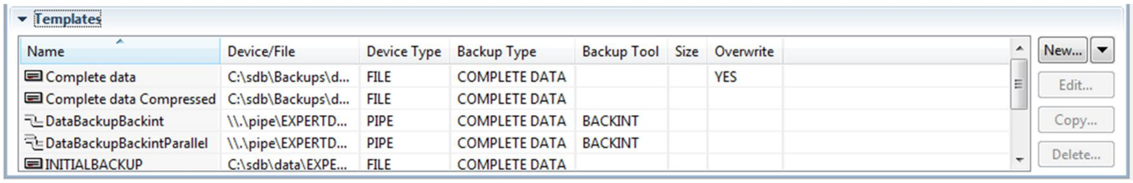
Agenda

1. General Information About External Backup Tools
2. Creating a MaxDB Backup With External Backup Tools
   - Legato Networker
   - Tivoli Storage Manager (TSM)
   - Backint Interface
3. Performing a Recovery With External Backup Tools
4. **Demo**
5. Problem Analysis

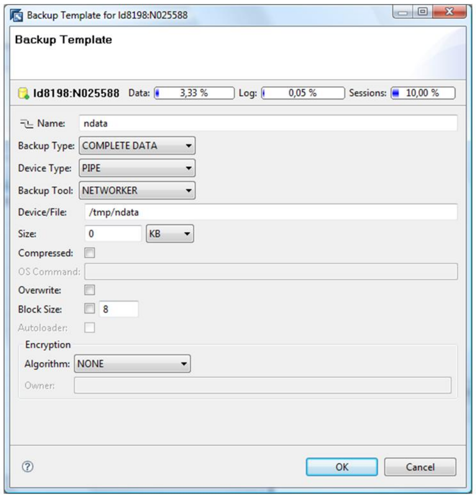
Next part of this session is a live demo of a backup and a recovery using the Backint for MaxDB and Backint for Oracle interface. The backup and recovery are started via the administration tool Database Studio.

These slides contain only a few screenshots of the shown demo, with some additional information.

External Backup Tools

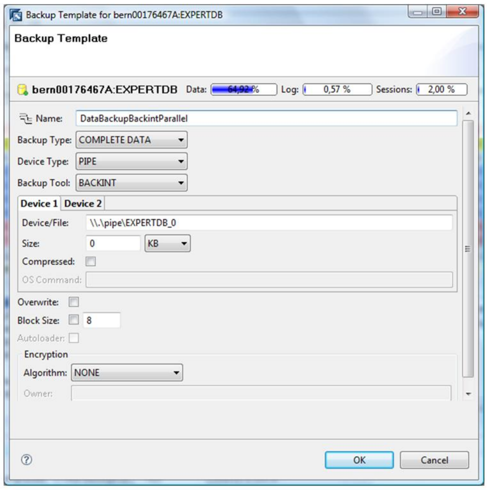


Name	Device/File	Device Type	Backup Type	Backup Tool	Size	Overwrite
Complete data	C:\sdb\Backups\d...	FILE	COMPLETE DATA			YES
Complete data Compressed	C:\sdb\Backups\d...	FILE	COMPLETE DATA			
DataBackupBackint	\\.\pipe\EXPERTD...	PIPE	COMPLETE DATA	BACKINT		
DataBackupBackintParallel	\\.\pipe\EXPERTD...	PIPE	COMPLETE DATA	BACKINT		
INITIALBACKUP	C:\sdb\data\EXPE...	FILE	COMPLETE DATA			



Id8198:N025588 Data: 3,33 % Log: 0,05 % Sessions: 10,00 %

Name: ndata
Backup Type: COMPLETE DATA
Device Type: PIPE
Backup Tool: NETWORKER
Device/File: /tmp/ndata
Size: 0 KB
Compressed: □
OS Command:
Overwrite: □
Block Size: □ 8
Autoloader: □
Encryption
Algorithm: NONE
Owner:



bern00176467A:EXPERTDB Data: 64,92 % Log: 0,57 % Sessions: 2,00 %

Name: DataBackupBackintParallel
Backup Type: COMPLETE DATA
Device Type: PIPE
Backup Tool: BACKINT
Device/File: \\.\pipe\EXPERTDB_0
Size: 0 KB
Compressed: □
OS Command:
Overwrite: □
Block Size: □ 8
Autoloader: □
Encryption
Algorithm: NONE
Owner:

Before you can perform backups, you must define the relevant backup templates. You can create and change backup templates or template groups of parallel backup media in Database Studio in the backup section of the Administration window by choosing Templates.

To be able to create a parallel backup template, you must set the value of the "MaxBackupMedia" parameter to match the number of individual templates in a parallel backup template. For example, if a template group is to comprise 10 individual templates, the value of the "MaxBackupMedia" parameter must be "10“ (or higher).

You can specify the following information for the template:

Name of the backup template. This name is freely definable and is not dependent on the storage location used (Device/File).

Backup Type: Specify the type of backup for which this template is to be used.

Device Type: Tape, file, or pipe – if an external backup tool is to be used, the Device Type must be set to pipe.

Backup Tool: Type of external backup tool (if applicable)

Device/File: Path to a device, name of a defined pipe, or name of a file including its path. If you do not specify a path, a file is created in the run directory of the database instance.

Size: Maximum size of the backups that can be created on this template (if you do not make an entry in this field, files of unlimited size can be created).

OS Command: In this field, you can specify operating system commands for backups to tape.

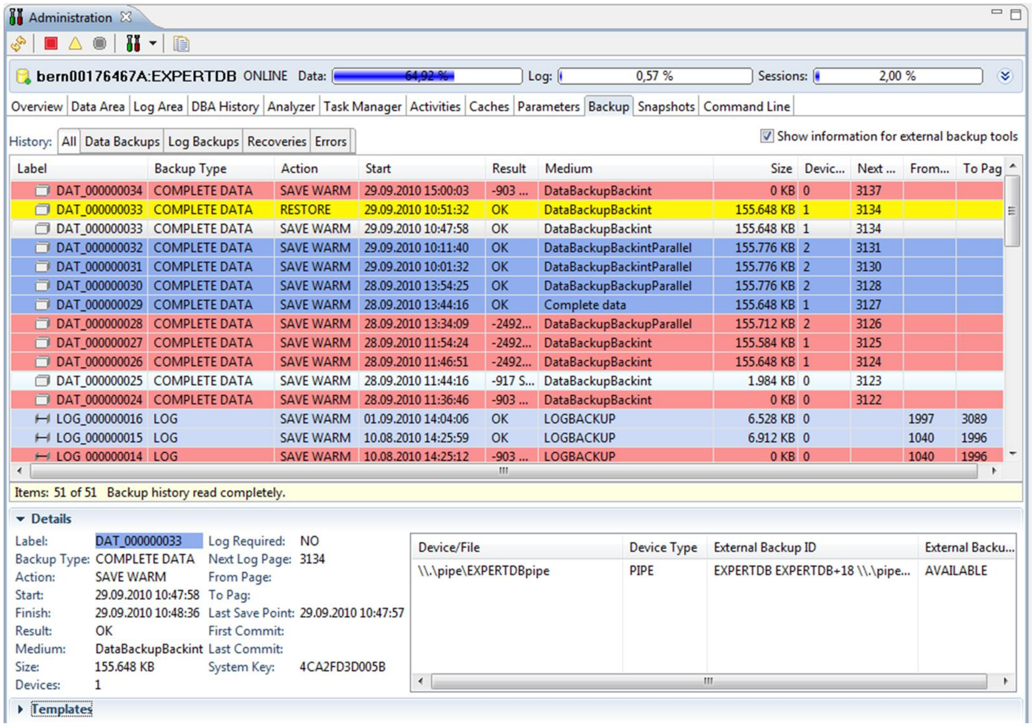
Overwrite: This option enables you to perform successive backups to the same file, overwriting the previous backup each time. Use this function carefully since it makes it impossible to restore one of the previous backups.

Block Size: The entry in this field defines the size of the data blocks to be written to the template. If page clustering is used for the instance, the value in this field must be larger than a multiple of the cluster size used (minimum block size, for example, of "64").

Autoloader: Select the Autoloader checkbox if you want to use a tape device with automatic tape swapping.

The above examples show one template which can be used for a backup to Networker and a template group comprising of 2 single templates which can be used for a parallel backup with Backint.

Backup History / Backup Details



Administration
bern00176467A:EXPERTDB ONLINE Data: 64,92 % Log: 0,57 % Sessions: 2,00 %
Overview
Data Area
Log Area
DBA History
Analyzer
Task Manager
Activities
Caches
Parameters
Backup
Snapshots
Command Line
History: All Data Backups Log Backups Recoveries Errors

☑ Show information for external backup tools

Label	Backup Type	Action	Start	Result	Medium	Size	Devic...	Next ...	From...	To Pag
DAT_000000034	COMPLETE DATA	SAVE WARM	29.09.2010 15:00:03	-903 ...	DataBackupBackint	0 KB	0	3137		
DAT_000000033	COMPLETE DATA	RESTORE	29.09.2010 10:51:32	OK	DataBackupBackint	155.648 KB	1	3134		
DAT_000000033	COMPLETE DATA	SAVE WARM	29.09.2010 10:47:58	OK	DataBackupBackint	155.648 KB	1	3134		
DAT_000000032	COMPLETE DATA	SAVE WARM	29.09.2010 10:11:40	OK	DataBackupBackintParallel	155.776 KB	2	3131		
DAT_000000031	COMPLETE DATA	SAVE WARM	29.09.2010 10:01:32	OK	DataBackupBackintParallel	155.776 KB	2	3130		
DAT_000000030	COMPLETE DATA	SAVE WARM	28.09.2010 13:54:25	OK	DataBackupBackupParallel	155.776 KB	2	3128		
DAT_000000029	COMPLETE DATA	SAVE WARM	28.09.2010 13:44:16	OK	Complete data	155.648 KB	1	3127		
DAT_000000028	COMPLETE DATA	SAVE WARM	28.09.2010 13:34:09	-2492...	DataBackupBackupParallel	155.712 KB	2	3126		
DAT_000000027	COMPLETE DATA	SAVE WARM	28.09.2010 11:54:24	-2492...	DataBackupBackint	155.584 KB	1	3125		
DAT_000000026	COMPLETE DATA	SAVE WARM	28.09.2010 11:46:51	-2492...	DataBackupBackint	155.648 KB	1	3124		
DAT_000000025	COMPLETE DATA	SAVE WARM	28.09.2010 11:44:16	-917 S...	DataBackupBackint	1.984 KB	0	3123		
DAT_000000024	COMPLETE DATA	SAVE WARM	28.09.2010 11:36:46	-903 ...	DataBackupBackint	0 KB	0	3122		
LOG_000000016	LOG	SAVE WARM	01.09.2010 14:04:06	OK	LOGBACKUP	6.528 KB	0		1997	3089
LOG_000000015	LOG	SAVE WARM	10.08.2010 14:25:59	OK	LOGBACKUP	6.912 KB	0		1040	1996
LOG 000000014	LOG	SAVE WARM	10.08.2010 14:25:12	-903 ...	LOGBACKUP	0 KB	0		1040	1996

Items: 51 of 51 Backup history read completely.

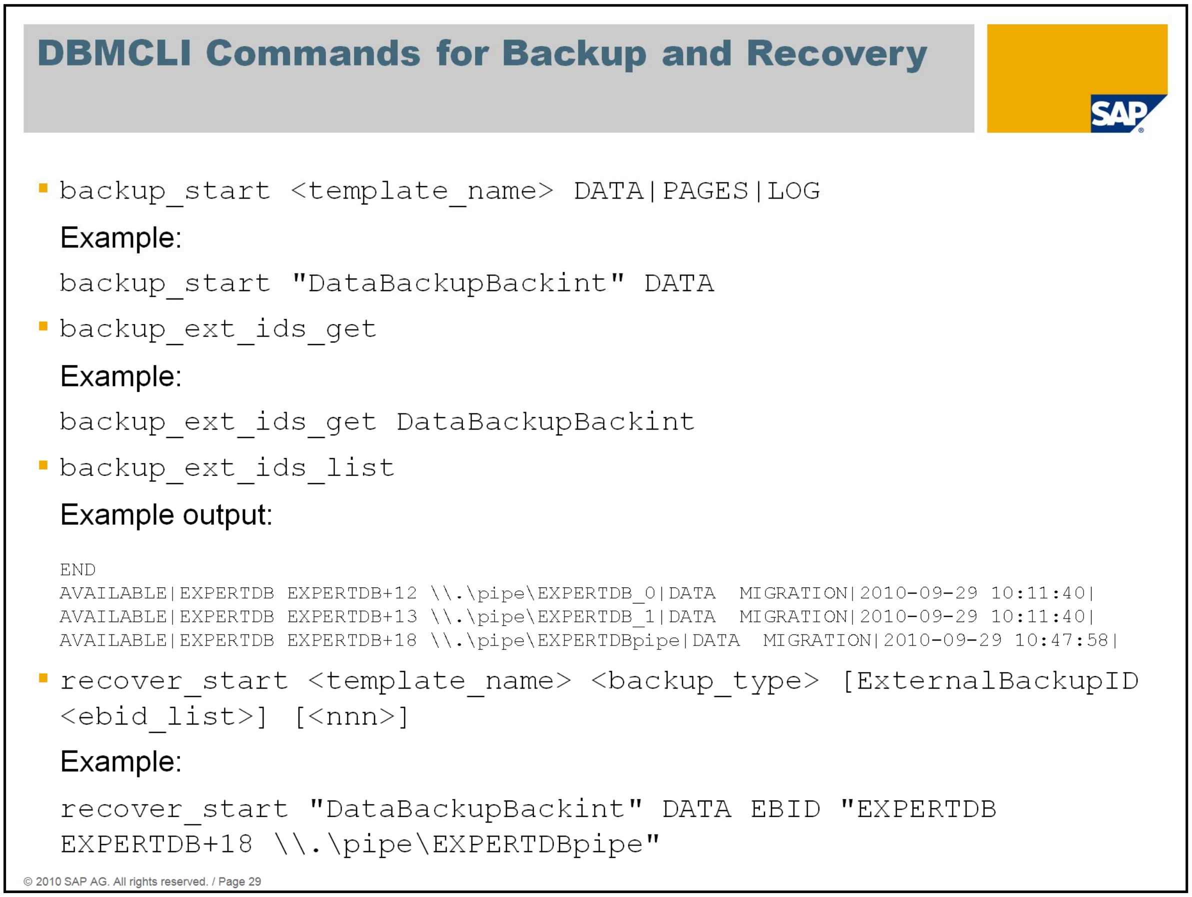
Details

Label:	DAT_000000033	Log Required:	NO
Backup Type:	COMPLETE DATA	Next Log Page:	3134
Action:	SAVE WARM	From Page:	
Start:	29.09.2010 10:47:58	To Pag:	
Finish:	29.09.2010 10:48:36	Last Save Point:	29.09.2010 10:47:57
Result:	OK	First Commit:	
Medium:	DataBackupBackint	Last Commit:	
Size:	155.648 KB	System Key:	4CA2FD3D005B
Devices:	1		

Device/File	Device Type	External Backup ID	External Backu...
\\.\pipe\EXPERTDBpipe	PIPE	EXPERTDB EXPERTDB+18 \\.\pipe...	AVAILABLE

Templates

The Backup History contains information about all successful and unsuccessful backups. Detailed information fo each entry is available in the Details section. Here also the external backup ID is displayed, if an external backup tool was used.



DBMCLI Commands for Backup and Recovery

- `backup_start <template_name> DATA|PAGES|LOG`

  Example:

  `backup_start "DataBackupBackint" DATA`

- `backup_ext_ids_get`

  Example:

  `backup_ext_ids_get DataBackupBackint`

- `backup_ext_ids_list`

  Example output:

```
END
AVAILABLE|EXPERTDB EXPERTDB+12 \\.\pipe\EXPERTDB_0|DATA  MIGRATION|2010-09-29 10:11:40|
AVAILABLE|EXPERTDB EXPERTDB+13 \\.\pipe\EXPERTDB_1|DATA  MIGRATION|2010-09-29 10:11:40|
AVAILABLE|EXPERTDB EXPERTDB+18 \\.\pipe\EXPERTDBpipe|DATA  MIGRATION|2010-09-29 10:47:58|
```

- `recover_start <template_name> <backup_type> [ExternalBackupID <ebid_list>] [<nnn>]`

  Example:

  `recover_start "DataBackupBackint" DATA EBID "EXPERTDB EXPERTDB+18 \\.\pipe\EXPERTDBpipe"`

When using DBMCLI, a backup of the database is done with the help of the backup_start command.

As the DBMServer recognizes the backup tool to be used from the backup template, there is no difference in the backup command between a backup with and a backup without a backup tool.

As more than one DBMServer command is needed for displaying the External Backup Identifiers, an interactive dbmcli session must be used.

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

The list has the following format:
<Availability>|<External Backup ID>|<backup type>|<date_time>|

If in an answer to backup_ext_ids_list or backup_ext_ids_listnext a line with a keyword CONTINUE follows the line with the keyword OK, the next part of the list can be requested with the backup_ext_ids_listnext command.

A restore is done with the commands recover_start and recover_replace (for restoring more than one log backup).

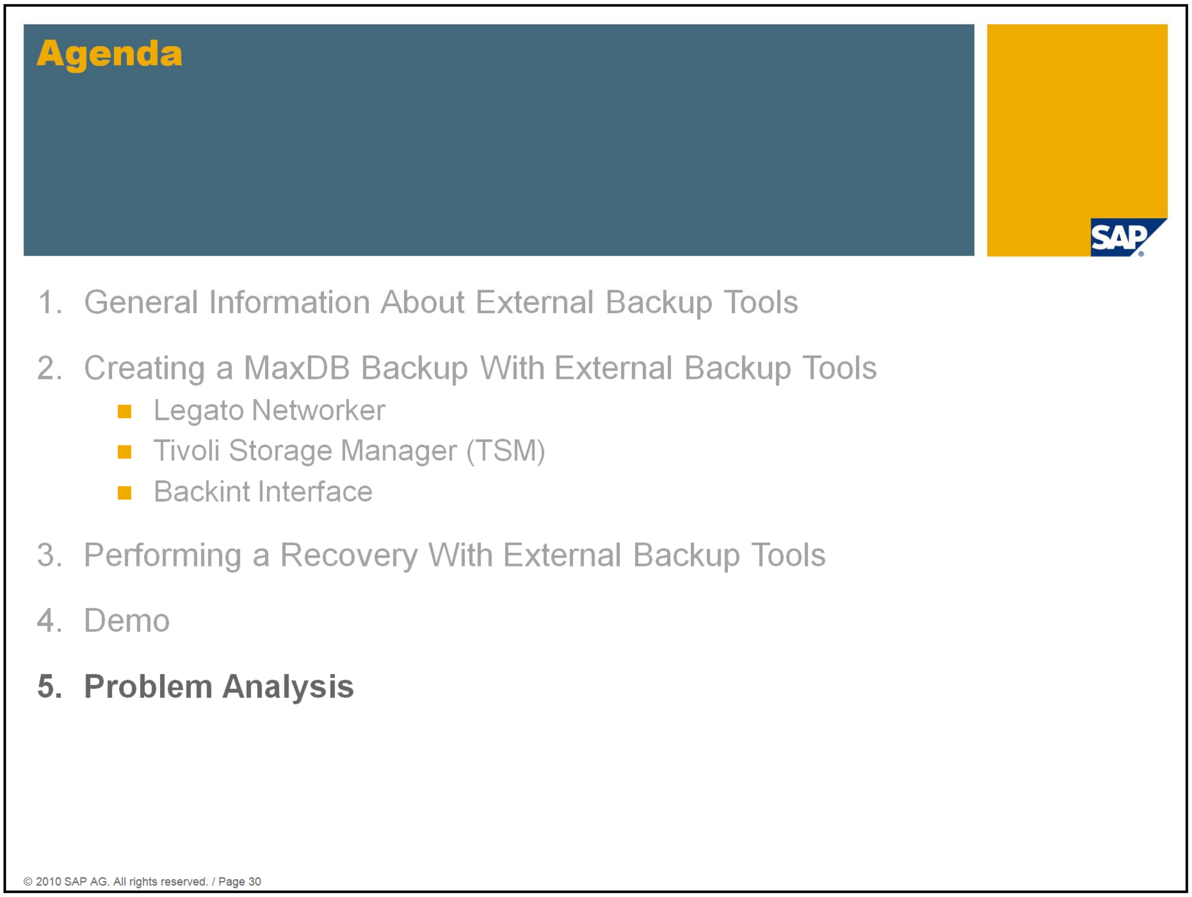
The keyword EBID (or ExternalBackupID) is followed by a comma-separated list of External Backup IDs. With parallel backups, all External Backup Identifiers of the individual backup parts must be transmitted as a comma-separated list enclosed in double quotes ("<ExtBackupID_1>, <ExtBackupID_2>, ..., <ExtBackupID_n>").

Further Examples:

recover_start ADSM LOG EBID P47579_DB7_2001.03.30_15.51.20_SAVELOG_ADSM

recover_start NSR DATA EBID "NST 985877420 P47579"

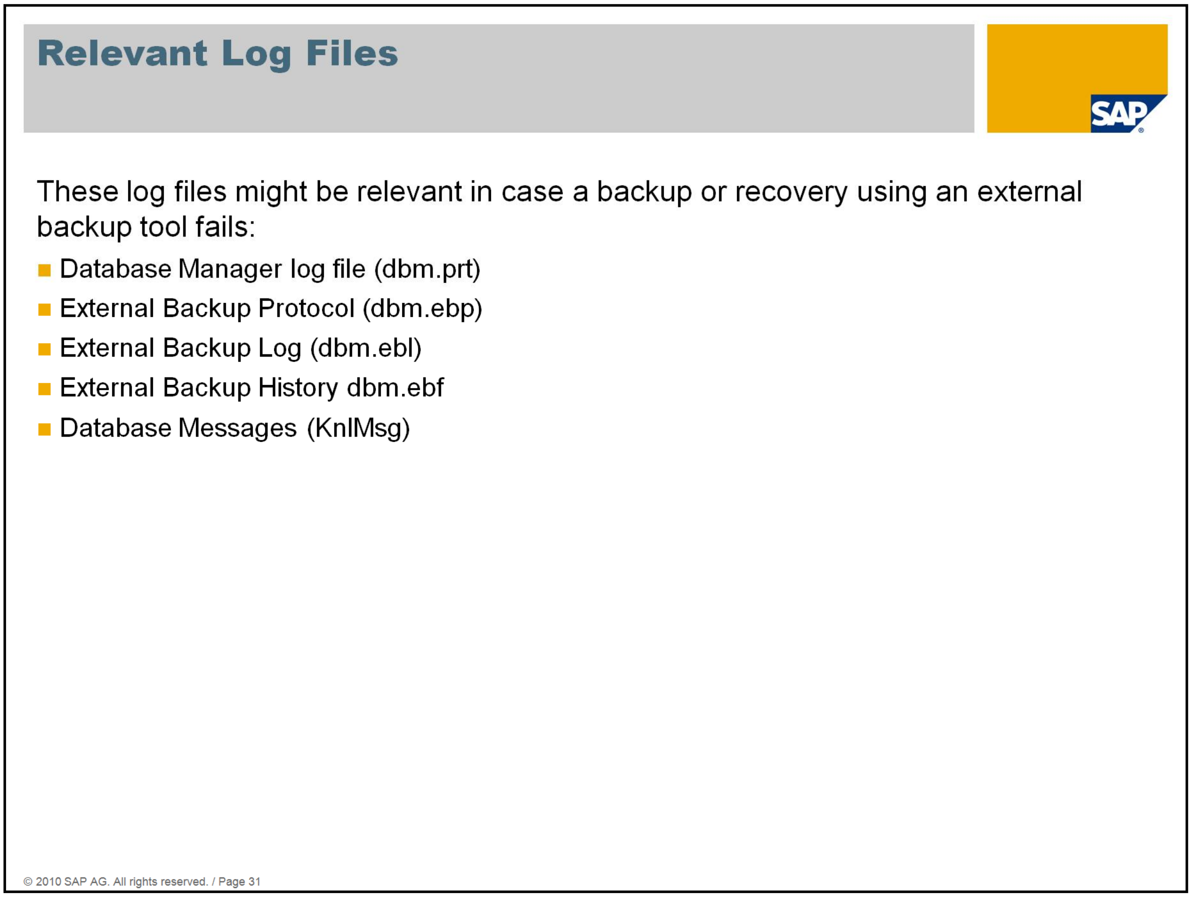
recover_start BACK PAGES EBID "DB72 985963853 \\.\pipe\b1,DB72 985963913 \\.\pipe\b2"



Agenda

1. General Information About External Backup Tools
2. Creating a MaxDB Backup With External Backup Tools
    - Legato Networker
    - Tivoli Storage Manager (TSM)
    - Backint Interface
3. Performing a Recovery With External Backup Tools
4. Demo
5. **Problem Analysis**

In case a backup or a recovery fails, you need to know, how the problem can be analyzed. This is explained in the next chapter.



Relevant Log Files

These log files might be relevant in case a backup or recovery using an external backup tool fails:

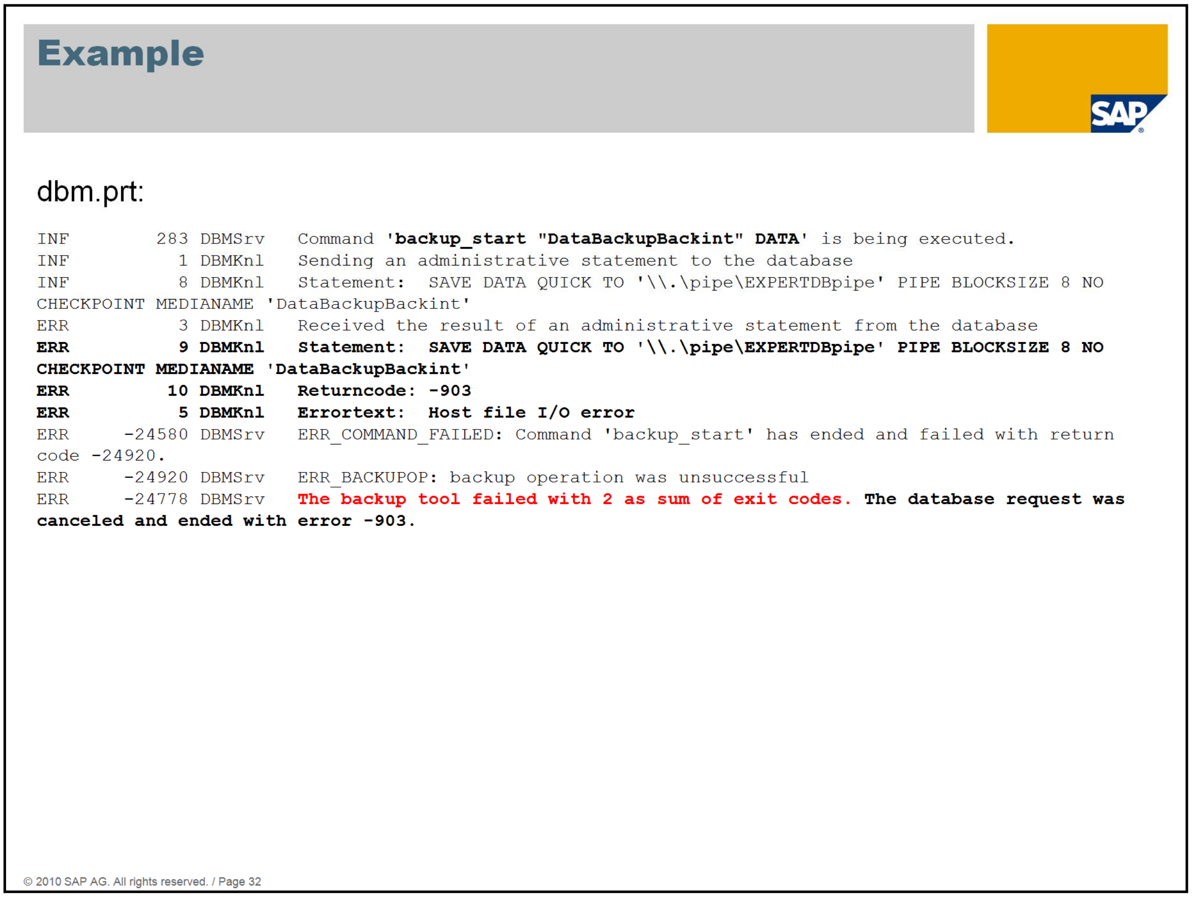
- Database Manager log file (dbm.prt)
- External Backup Protocol (dbm.ebp)
- External Backup Log (dbm.ebl)
- External Backup History dbm.ebf
- Database Messages (KnlMsg)

The database manager log file contains the backup and recovery calls and – if an error occured - the error message. Therefore this log file can (in addition to the backup history and the external backup history) be used to check the success of a backup/recovery.

Detailed information regarding the backup/recovery can be found in the external backup protocol (or if this file has already been overwritten in the external backup log). In addition to information about the configuration parameter of the external backup 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 which component was responsible for the problem.

Depending of the cause of the error, it might be necessary to analyze log files of the backup tool.

In case the cause for the backup or recovery failure is not the communication with the external backup tool or problems of the external backup tool, but in the actual processing of the data by the database kernel, the database messages file should be checked for more detailed information regarding the problem.



Example

dbm.prt:

```
INF        283 DBMSrv   Command 'backup_start "DataBackupBackint" DATA' is being executed.
INF          1 DBMKnl   Sending an administrative statement to the database
INF          8 DBMKnl   Statement:  SAVE DATA QUICK TO '\\.\pipe\EXPERTDBpipe' PIPE BLOCKSIZE 8 NO
CHECKPOINT MEDIANAME 'DataBackupBackint'
ERR          3 DBMKnl   Received the result of an administrative statement from the database
ERR          9 DBMKnl   Statement:  SAVE DATA QUICK TO '\\.\pipe\EXPERTDBpipe' PIPE BLOCKSIZE 8 NO
CHECKPOINT MEDIANAME 'DataBackupBackint'
ERR         10 DBMKnl   Returncode: -903
ERR          5 DBMKnl   Errortext:  Host file I/O error
ERR     -24580 DBMSrv   ERR_COMMAND_FAILED: Command 'backup_start' has ended and failed with return
code -24920.
ERR     -24920 DBMSrv   ERR_BACKUPOP: backup operation was unsuccessful
ERR     -24778 DBMSrv   The backup tool failed with 2 as sum of exit codes. The database request was
canceled and ended with error -903.
```

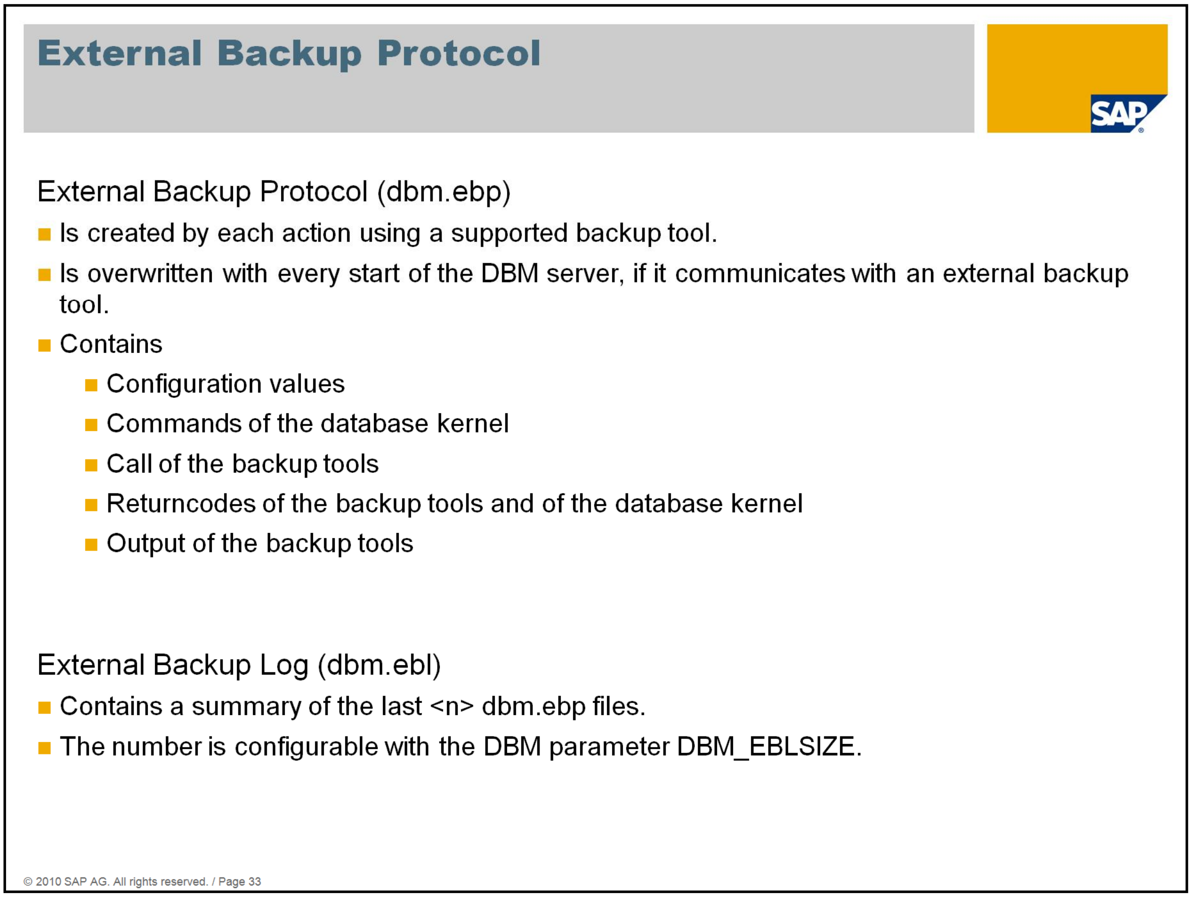
In file dbm.prt you can see that the backup was started to a backup template called DataBackupBackint. The exact statement sent to the database kernel is logged as well as an error messages.

Error message „The backup tool failed with 2 as sum of exit codes. The database request was canceled and ended with error -903.“ indicates, that the backup tool caused the problem and that the database request was only cancelled as a result of that failure. So the error analysis has to concentrate on the backup tool and its configuration.

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

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

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



External Backup Protocol

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.
- Contains
  - 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

External Backup Log (dbm.ebl)

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

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 DB50: *Properties -> Files -> BACKEBP*

Access via Database Studio: *Diagnosis Files -> External Backup Log File (former External Backup Protocol)*

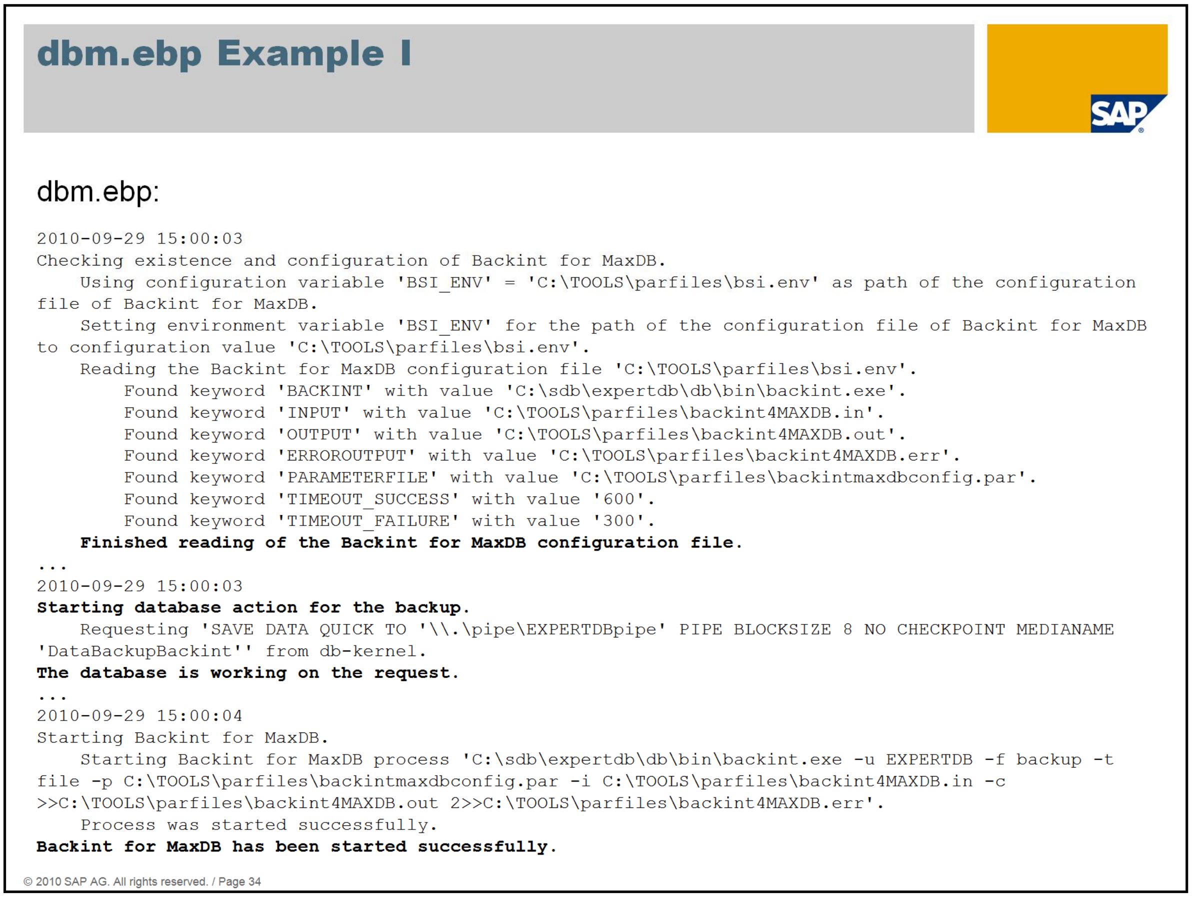
Note that this file is overwritten after each start of the DBM server when it communicates with the external backup tool. A new DBM server is started with each dbmcli call, to name one example.

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.

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

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



dbm.ebp Example I

dbm.ebp:

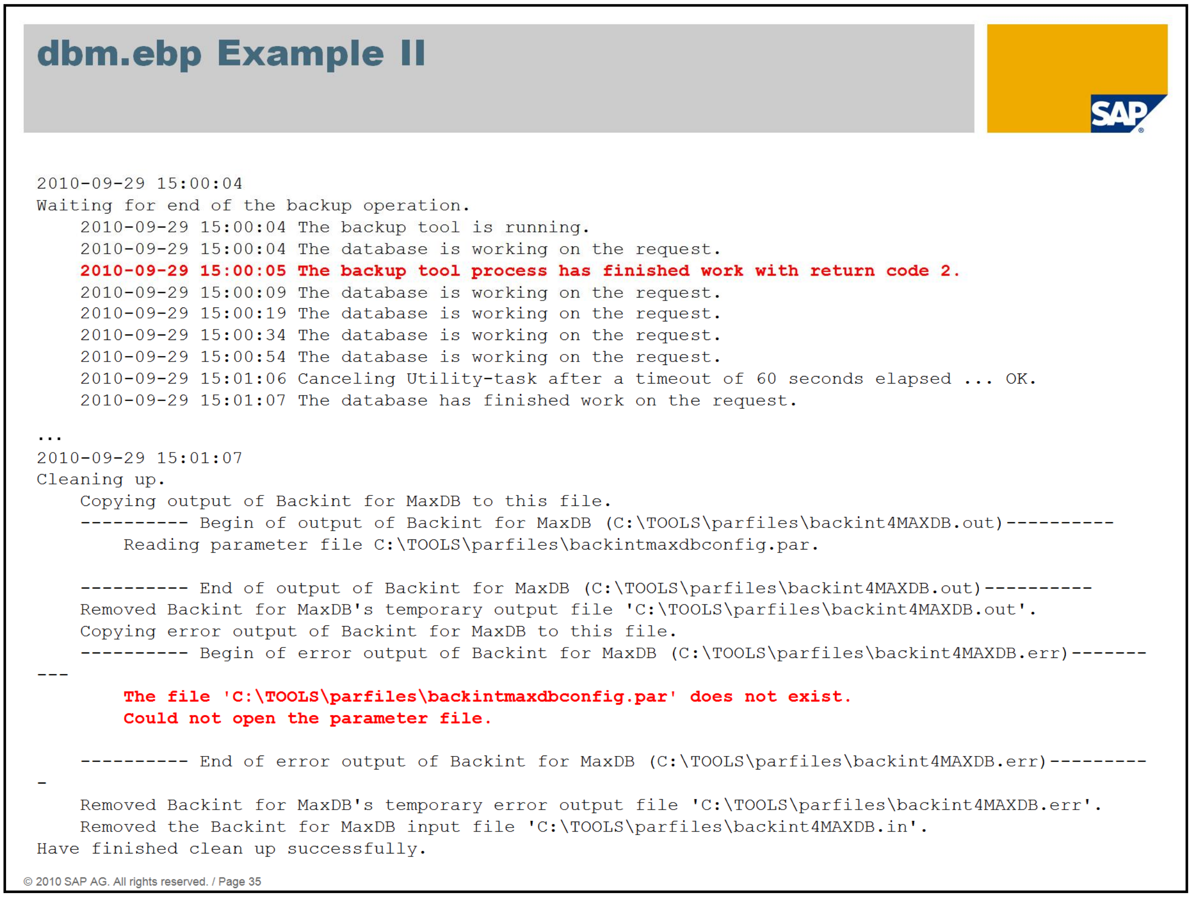
```
2010-09-29 15:00:03
Checking existence and configuration of Backint for MaxDB.
    Using configuration variable 'BSI_ENV' = 'C:\TOOLS\parfiles\bsi.env' as path of the configuration
file of Backint for MaxDB.
    Setting environment variable 'BSI_ENV' for the path of the configuration file of Backint for MaxDB
to configuration value 'C:\TOOLS\parfiles\bsi.env'.
    Reading the Backint for MaxDB configuration file 'C:\TOOLS\parfiles\bsi.env'.
        Found keyword 'BACKINT' with value 'C:\sdb\expertdb\db\bin\backint.exe'.
        Found keyword 'INPUT' with value 'C:\TOOLS\parfiles\backint4MAXDB.in'.
        Found keyword 'OUTPUT' with value 'C:\TOOLS\parfiles\backint4MAXDB.out'.
        Found keyword 'ERROROUTPUT' with value 'C:\TOOLS\parfiles\backint4MAXDB.err'.
        Found keyword 'PARAMETERFILE' with value 'C:\TOOLS\parfiles\backintmaxdbconfig.par'.
        Found keyword 'TIMEOUT_SUCCESS' with value '600'.
        Found keyword 'TIMEOUT_FAILURE' with value '300'.
    Finished reading of the Backint for MaxDB configuration file.
...
2010-09-29 15:00:03
Starting database action for the backup.
    Requesting 'SAVE DATA QUICK TO '\\.\pipe\EXPERTDBpipe' PIPE BLOCKSIZE 8 NO CHECKPOINT MEDIANAME
'DataBackupBackint'' from db-kernel.
The database is working on the request.
...
2010-09-29 15:00:04
Starting Backint for MaxDB.
    Starting Backint for MaxDB process 'C:\sdb\expertdb\db\bin\backint.exe -u EXPERTDB -f backup -t
file -p C:\TOOLS\parfiles\backintmaxdbconfig.par -i C:\TOOLS\parfiles\backint4MAXDB.in -c
>>C:\TOOLS\parfiles\backint4MAXDB.out 2>>C:\TOOLS\parfiles\backint4MAXDB.err'.
    Process was started successfully.
Backint for MaxDB has been started successfully.
```

This is the beginning of file dbm.ebp. You can see that variable BSI_ENV is set to C:\TOOLS\parfiles\bsi.env. Next, the configuration parameters read from this file are listed. In case a parameter is spelled incorrectly, this would be visible here, as unknown keywords are explicitely listed.

In this example, the configuration file is fine.

The backup request was sent to the database successfully and afterwards Backint for MaxDB was started successfully as well.

So far, everything looks fine – however, the log file is continued on the next slide…



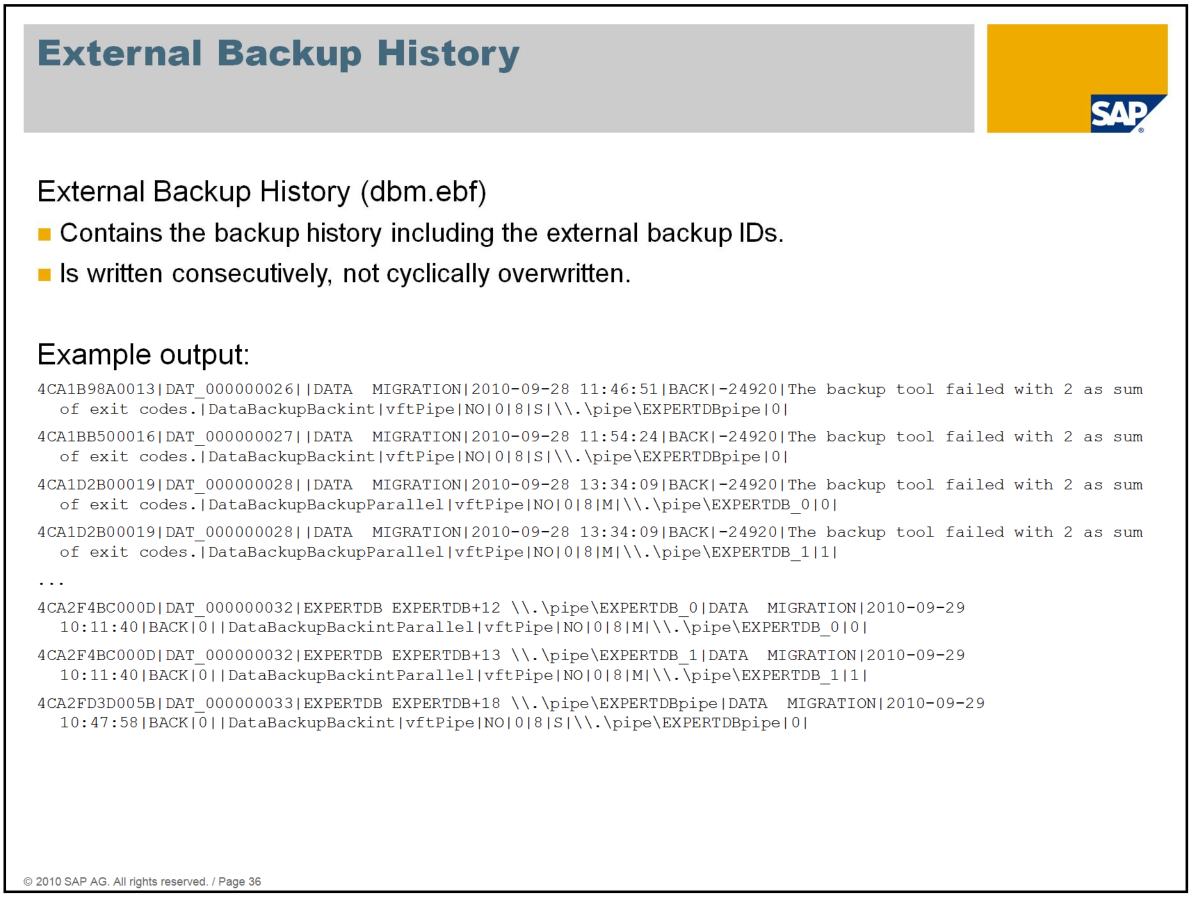
dbm.ebp Example II

```
2010-09-29 15:00:04
Waiting for end of the backup operation.
    2010-09-29 15:00:04 The backup tool is running.
    2010-09-29 15:00:04 The database is working on the request.
    2010-09-29 15:00:05 The backup tool process has finished work with return code 2.
    2010-09-29 15:00:09 The database is working on the request.
    2010-09-29 15:00:19 The database is working on the request.
    2010-09-29 15:00:34 The database is working on the request.
    2010-09-29 15:00:54 The database is working on the request.
    2010-09-29 15:01:06 Canceling Utility-task after a timeout of 60 seconds elapsed ... OK.
    2010-09-29 15:01:07 The database has finished work on the request.
...
2010-09-29 15:01:07
Cleaning up.
    Copying output of Backint for MaxDB to this file.
    ---------- Begin of output of Backint for MaxDB (C:\TOOLS\parfiles\backint4MAXDB.out)----------
        Reading parameter file C:\TOOLS\parfiles\backintmaxdbconfig.par.

    ---------- End of output of Backint for MaxDB (C:\TOOLS\parfiles\backint4MAXDB.out)----------
    Removed Backint for MaxDB's temporary output file 'C:\TOOLS\parfiles\backint4MAXDB.out'.
    Copying error output of Backint for MaxDB to this file.
    ---------- Begin of error output of Backint for MaxDB (C:\TOOLS\parfiles\backint4MAXDB.err)-------
---
        The file 'C:\TOOLS\parfiles\backintmaxdbconfig.par' does not exist.
        Could not open the parameter file.

    ---------- End of error output of Backint for MaxDB (C:\TOOLS\parfiles\backint4MAXDB.err)---------
-
    Removed Backint for MaxDB's temporary error output file 'C:\TOOLS\parfiles\backint4MAXDB.err'.
    Removed the Backint for MaxDB input file 'C:\TOOLS\parfiles\backint4MAXDB.in'.
Have finished clean up successfully.
```

Once the database kernel and the backup tool are started, the DBMServer determines their state regularly. As you can see, the backup tool failed shortly after it was started, error message „The backup tool process has finished work with return code 2.“ is logged.

As a consequence of that, the database request was cancelled by the DBMServer.

In the output information of Backint for MaxDB you can find the reason for the failure: the parameter file 'C:\TOOLS\parfiles\backintmaxdbconfig.par' specified in the bsi.env file could not be found by Backint for MaxDB. Therefore the tool could not start to work on the backup request.



External Backup History

External Backup History (dbm.ebf)

- Contains the backup history including the external backup IDs.
- Is written consecutively, not cyclically overwritten.

Example output:

```
4CA1B98A0013|DAT_000000026||DATA  MIGRATION|2010-09-28 11:46:51|BACK|-24920|The backup tool failed with 2 as sum
  of exit codes.|DataBackupBackint|vftPipe|NO|0|8|S|\\.\pipe\EXPERTDBpipe|0|
4CA1BB500016|DAT_000000027||DATA  MIGRATION|2010-09-28 11:54:24|BACK|-24920|The backup tool failed with 2 as sum
  of exit codes.|DataBackupBackint|vftPipe|NO|0|8|S|\\.\pipe\EXPERTDBpipe|0|
4CA1D2B00019|DAT_000000028||DATA  MIGRATION|2010-09-28 13:34:09|BACK|-24920|The backup tool failed with 2 as sum
  of exit codes.|DataBackupBackupParallel|vftPipe|NO|0|8|M|\\.\pipe\EXPERTDB_0|0|
4CA1D2B00019|DAT_000000028||DATA  MIGRATION|2010-09-28 13:34:09|BACK|-24920|The backup tool failed with 2 as sum
  of exit codes.|DataBackupBackupParallel|vftPipe|NO|0|8|M|\\.\pipe\EXPERTDB_1|1|
...
4CA2F4BC000D|DAT_000000032|EXPERTDB EXPERTDB+12 \\.\pipe\EXPERTDB_0|DATA  MIGRATION|2010-09-29
  10:11:40|BACK|0||DataBackupBackintParallel|vftPipe|NO|0|8|M|\\.\pipe\EXPERTDB_0|0|
4CA2F4BC000D|DAT_000000032|EXPERTDB EXPERTDB+13 \\.\pipe\EXPERTDB_1|DATA  MIGRATION|2010-09-29
  10:11:40|BACK|0||DataBackupBackintParallel|vftPipe|NO|0|8|M|\\.\pipe\EXPERTDB_1|1|
4CA2FD3D005B|DAT_000000033|EXPERTDB EXPERTDB+18 \\.\pipe\EXPERTDBpipe|DATA  MIGRATION|2010-09-29
  10:47:58|BACK|0||DataBackupBackint|vftPipe|NO|0|8|S|\\.\pipe\EXPERTDBpipe|0|
```

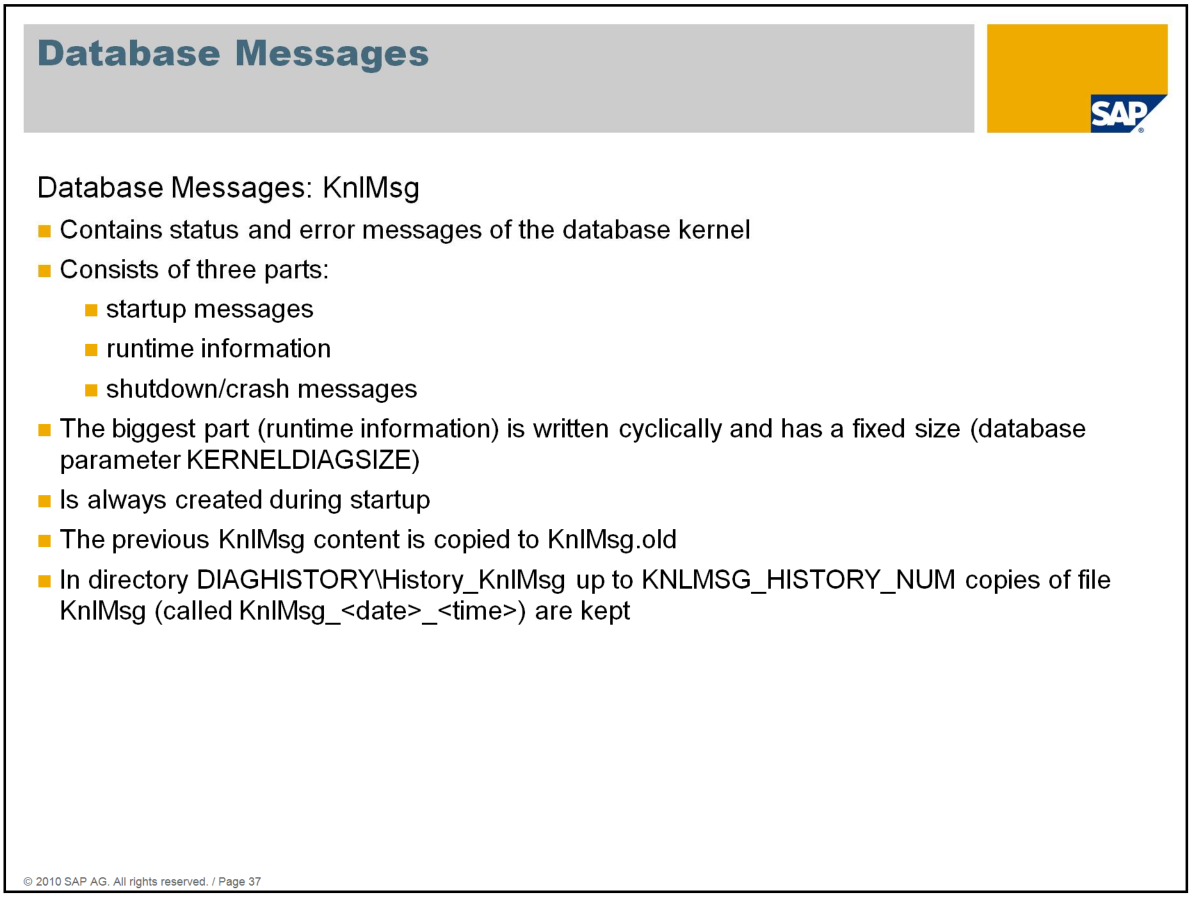
The file dbm.ebf contains the backup history, the backup ID, external backup IDs and error messages. This file is written consecutively and is NOT cyclically overwritten, so that the entire backup history is available for support.

If a backup tool was able to backup successfully, but could not determine the external backup ID, the backup is entered as failed in the backup history.

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

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

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

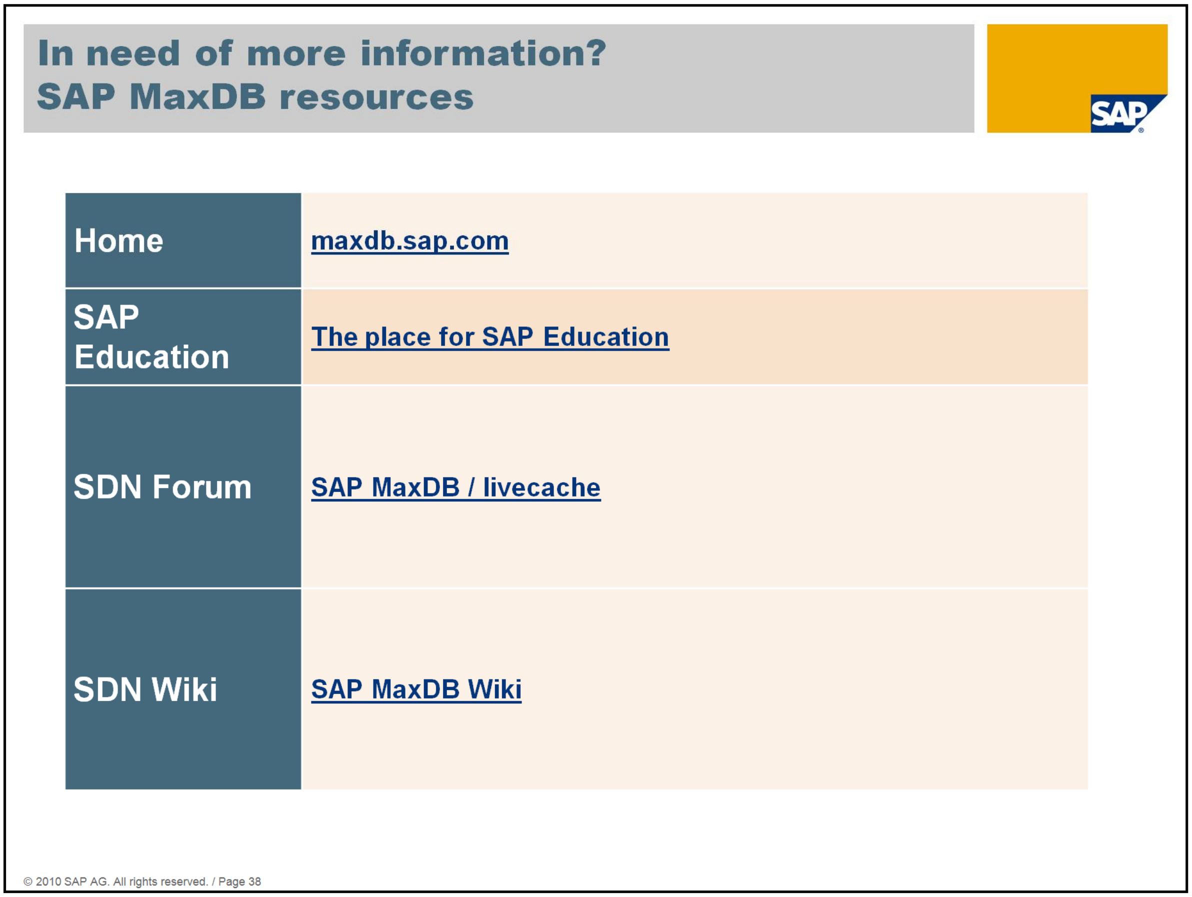


Database Messages

Database Messages: KnlMsg

- Contains status and error messages of the database kernel
- Consists of three parts:
  - startup messages
  - runtime information
  - shutdown/crash messages
- The biggest part (runtime information) is written cyclically and has a fixed size (database parameter KERNELDIAGSIZE)
- Is always created during startup
- The previous KnlMsg content is copied to KnlMsg.old
- In directory DIAGHISTORY\History_KnlMsg up to KNLMSG_HISTORY_NUM copies of file KnlMsg (called KnlMsg_<date>_<time>) are kept

In case the cause for the backup or recovery failure is not the communication with the external backup tool or problems of the external backup tool, but in the actual processing of the data by the database kernel, the database messages file should be checked for more detailed information regarding the problem.



In need of more information? SAP MaxDB resources

Home	maxdb.sap.com
SAP Education	The place for SAP Education
SDN Forum	SAP MaxDB / livecache
SDN Wiki	SAP MaxDB Wiki

In case you‘re ever in need of more information on any kind of subject on SAP MaxDB (or liveCache), please direct your search towards:

- The SAP MaxDB site: maxdb.sap.com. This site contains a lot of information, ranging from our official documentation to the recordings of previous Expert Sessions
- Next is the official SAP Education site: it contains MANY offers for all kind of courses on any SAP topics, including for example, the ADM515 administration course on SAP MaxDB and the UMEW60, concentrating on SAP MaxDB monitoring and optimization.
- Then, we have the heavily used SAP MaxDB forum. In case of questions on SAP MaxDB products, please register and join the Community!
- Lastly, we have our also equally well visited Wiki pages. We’ve added a lot of information here that might interest any SAP MaxDB DBA, including a documentation on tools like x_cons and a Support Guide.

Further information on the topic of external backup tools can also be found in SAP Note 822240 (FAQ: MaxDB and external backup tools).

Thank You!
Bye, Bye – And Remember Next Session

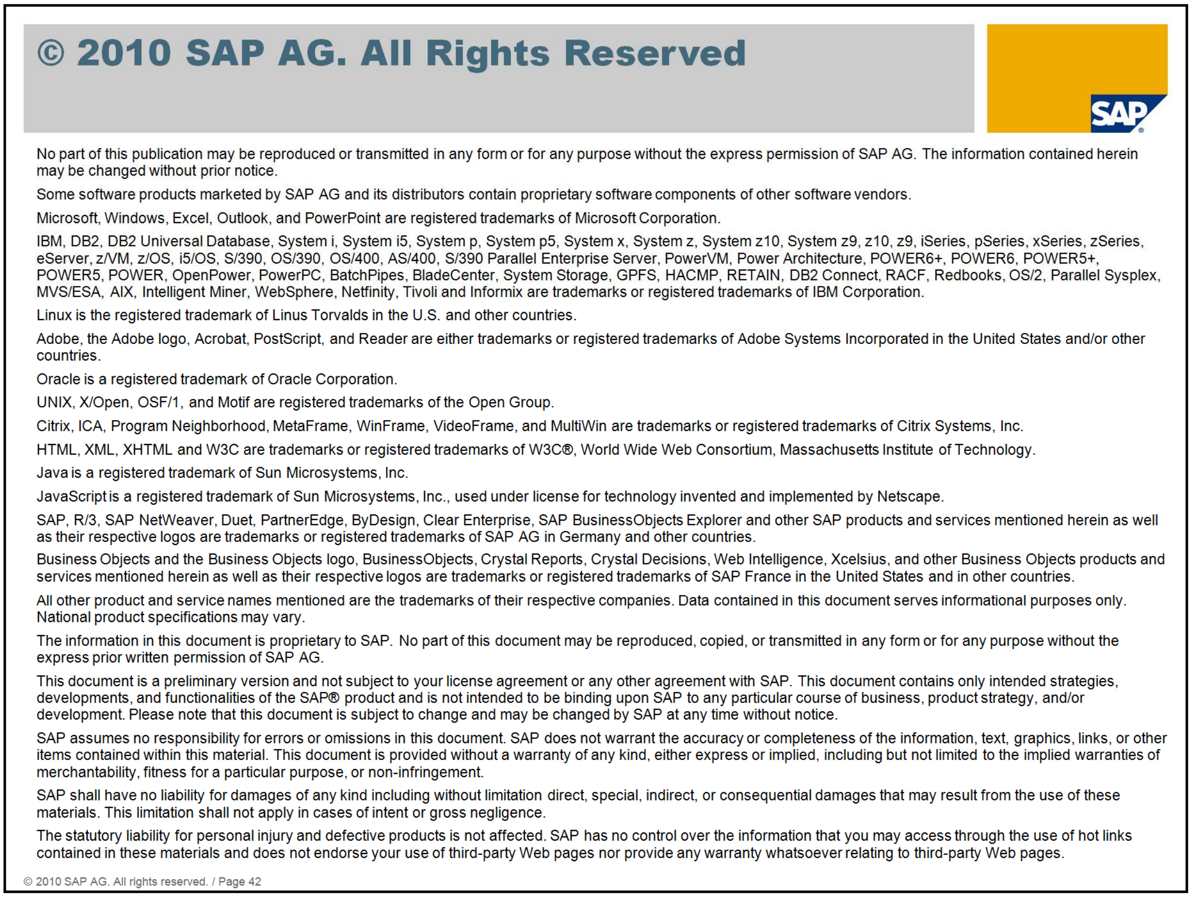
November 9, 2010	**Session 14:** Tracing: SQLDBC, ODBC & VTRACE

Questions and Answers

Disclaimer

This presentation reflects current planning.

**Contents may be changed
without prior notice, and are in no way
binding upon SAP.**



© 2010 SAP AG. All Rights Reserved

This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP to any particular course of business, product strategy, and/or development. Please note that this document is subject to change and may be changed by SAP at any time without notice.

SAP assumes no responsibility for errors or omissions in this document. SAP does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. This limitation shall not apply in cases of intent or gross negligence.

The statutory liability for personal injury and defective products is not affected. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third-party Web pages nor provide any warranty whatsoever relating to third-party Web pages.