TADM51

SAP Netweaver AS DB Operation (Oracle)

PARTICIPANT HANDBOOK
INSTRUCTOR-LED TRAINING

Course Version: 10
Course Duration: 5 Day(s)
Material Number: 50117656

SAP Copyrights and Trademarks

About This Handbook

This handbook is intended to complement the instructor-led presentation of this course, and serve as a source of reference. It is not suitable for self-study.

Typographic Conventions

American English is the standard used in this handbook.

The following typographic conventions are also used.

This information is displayed in the instructor's presentation	
Demonstration	
Procedure	
Warning or Caution	
Hint	
Related or Additional Information	
Facilitated Discussion	
User interface control	*Example text*
Window title	*Example text*

Contents

xi Course Overview

1 Unit 1: Database Overview

2 Lesson: Reviewing Database Architecture
23 Exercise 1: Analyze Oracle Environment Variables
27 Lesson: Connecting to the Database
41 Exercise 2: Establish Connection to the Database
47 Lesson: Using Database Administration Tools
69 Exercise 3: Use Database Administration Tools
73 Lesson: Administrating Oracle Instances
87 Exercise 4: Change Oracle Parameters

101 Unit 2: Backup, Restore, and Recovery

102 Lesson: Reviewing a Backup Strategy
113 Exercise 5: Create a Backup Strategy
116 Lesson: Reviewing Backup Tools
133 Exercise 6: Use Backup Tools
153 Lesson: Performing Backups
169 Exercise 7: Perform Backups
173 Lesson: Performing Restore and Recovery
193 Exercise 8: Perform Restore and Recovery to the System
205 Lesson: Working with Advanced Backup Techniques

219 Unit 3: Monitors and Tools

220 Lesson: Introducing Oracle Data Management
235 Exercise 9: Manage Oracle Tablespace Using SAP Tools
239 Lesson: Managing Database System Check
255 Exercise 10: Perform Database System Checks
261 Lesson: Explaining Computing Center Management System (CCMS) Alert Monitor

269 Unit 4: Space Management

270 Lesson: Administering Tablespaces
285 Exercise 11: Performing Tablespace Administration
292 Lesson: Performing Reorganization of Tables
313 Exercise 12: Work with Table Reorganization
318 Lesson: Housekeeping and Troubleshooting
327 Exercise 13: Perform Housekeeping and Troubleshooting

335	**Unit 5:**	**Oracle Cache Management**
336		Lesson: Introducing Oracle System Global Area
351		Exercise 14: Activate Dynamic System Global Area
356		Lesson: Introducing Automatic Oracle Program Global Area
365		Exercise 15: Activate Automatic PGA
373	**Unit 6:**	**Monitoring of the Database Instance**
374		Lesson: Working in the DBA Cockpit Performance Monitor
376		Lesson: Using the DBA Cockpit to Monitor the Database
385		Exercise 16: Navigate in the DBA Cockpit Performance Monitor
393	**Unit 7:**	**Application Design Analysis**
394		Lesson: Understanding the Impact of Expensive SQL Statements
397		Lesson: Using SM50/SM66 to Find Expensive SQL Statements
399		Lesson: Using ST03N/STAD to Find Expensive SQL Statements
402		Lesson: Using ST04 to Find Expensive SQL Statements
411		Exercise 17: Use ST04
415		Lesson: Using SQL Trace to Find Expensive SQL Statements
421		Exercise 18: Use the SQL Trace to Find Expensive SQL Statements
425		Lesson: Monitoring Exclusive Lock Waits
435		Exercise 19: Monitor Exclusive Lock Waits
445	**Unit 8:**	**Index Management and Optimization**
446		Lesson: Utilizing Indexes
460		Lesson: Creating an Index
473		Exercise 20: Create an Index
483	**Unit 9:**	**Cost-Based Optimizer**
484		Lesson: Updating Statistics
502		Lesson: Detecting Problems with Optimizer Statistics
509		Exercise 21: Detect Problems with Database Statistics
513		Exercise 22: Use the Cost-Based Optimizer
523	**Unit 10:**	**Analysis of Physical and Logical Layout**
524		Lesson: Defragmenting Indexes
535		Exercise 23: Detect and Resolve Fragmentation in Indexes
540		Lesson: Working with I/O Contention

547 Unit 11: **Memory Configuration Analysis**

548 Lesson: Describing Data Buffer Utilization
557 Exercise 24: Estimate Data Buffer Utilization
560 Lesson: Analyzing the Efficiency of Shared Pool
565 Exercise 25: Estimate the Efficiency of the Shared Buffer Pool
568 Lesson: Monitoring the Automatic Program Global Area
577 Exercise 26: Monitor the Automatic PGA

583 Unit 12: **Wait Interface and Automatic Workload Repository**

584 Lesson: Observing Oracle Wait Interface
613 Exercise 27: Monitor the Oracle Wait Interfaces
616 Lesson: Reviewing the Automatic Workload Repository and Histories
629 Exercise 28: Use the Automatic Workload Repository

Course Overview

TARGET AUDIENCE

This course is intended for the following audiences:

- Support Consultant
- Technology Consultant
- Systems Architect
- System Administrator

UNIT 1 Database Overview

Lesson 1

Reviewing Database Architecture 2

Exercise 1: Analyze Oracle Environment Variables 23

Lesson 2

Connecting to the Database 27

Exercise 2: Establish Connection to the Database 41

Lesson 3

Using Database Administration Tools 47

Exercise 3: Use Database Administration Tools 69

Lesson 4

Administrating Oracle Instances 73

Exercise 4: Change Oracle Parameters 87

UNIT OBJECTIVES

- Analyze Oracle environment variables
- Establish connection to the database
- Use database administration tools
- Change Oracle parameters

Unit 1
Lesson 1

Reviewing Database Architecture

LESSON OVERVIEW
This lesson explains the Oracle architecture, which helps you to understand the administration and working of restore and recovery.

Business Example

You want to install an SAP system with an Oracle database in your company. Before the installation, you need to plan and prepare the disk layout on the server.

There are general recommendations in the installation guide for the disk layout, but in practice, the distribution of database components on different disks is a compromise between security, optimal disk usage, and performance.

You need the background to design a good disk layout according to your requirements. For this reason, you require the following knowledge:

- An understanding of the architecture and the main components of an Oracle database
- An understanding of the basic concepts of the Oracle database
- An understanding of the file structure of an Oracle database in an SAP system

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Analyze Oracle environment variables

Database Architecture

An Oracle server is a relational database management system (RDBMS); that is, a server component that manages a relational database model.

An RDBMS is able to perform the following functions:

- Manages large amounts of data in a multiuser environment so that many users can concurrently access the same data
- Maintains relationships between data
- Provides secure access to data using user authorization concept
- Recovers data automatically to the most recent consistent status after a system failure
- Delivers high performance for processing data requests

In an SAP system, the database administrator (DBA) is the only interactive user connected to the database server. Application data processing is typically performed by work processes of SAP instances in the role of database clients.

Database Terminology

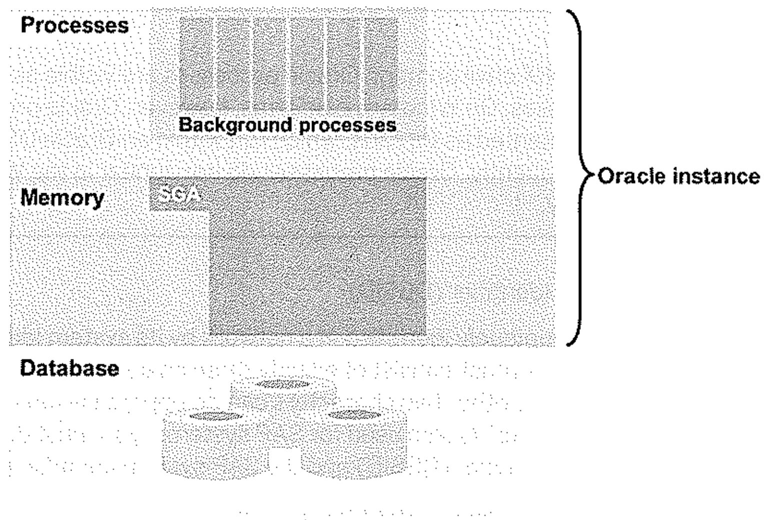


Figure 1: Database Terminology

The basic Oracle database terminologies are as follows:

- Database

  An Oracle database is a collection of data, logically treated as a unit. The data is physically stored in one or several files. Oracle manages data in logical units called tablespaces. A database object, such as a table, is always created in a particular tablespace. A tablespace consists of one or more files.

- Instance

  Because the database is only a passive part of a database server, some processes and memory structures are needed to access the data and manage the database. The combination of Oracle (background) processes and memory buffers is called an Oracle instance.

  Every running Oracle database is linked to an Oracle instance. Moreover, every Oracle database needs its own instance.

> Hint:
> Using real application clusters (RAC), two or more instances can serve one database.

- System Global Area (SGA)

  When an Oracle instance starts, a shared memory region called the SGA is allocated. The SGA allocated by an Oracle instance can be accessed only by the processes of this instance. This means that each instance has its own SGA. The SGA contains copies of data and control information for the corresponding Oracle instance. When the instance stops, the SGA is de-allocated.

- Processes

When an Oracle instance starts, Oracle background processes are started. When an instance shuts down, the processes stop.

> Hint:
> In a UNIX environment, Oracle processes are visible as operating system (OS) processes. On Windows platforms, however, Oracle processes are implemented as threads, which run within an Oracle OS process oracle.exe. Consequently, Oracle processes do not appear in the Windows OS process display.

- Database System Identifier (DBSID)

  Every database is uniquely identified in a network by its system identifier. On SAP systems, the system identifier must consist of exactly three characters, the first of which must be an uppercase letter. The other two characters can be uppercase letters or digits. Because the term "system identifier" is also used for SAP systems, you must distinguish between the database system identifier (DBSID) and the SAP system identifier (SAPSID).

Oracle Instance and Database – Architecture Overview

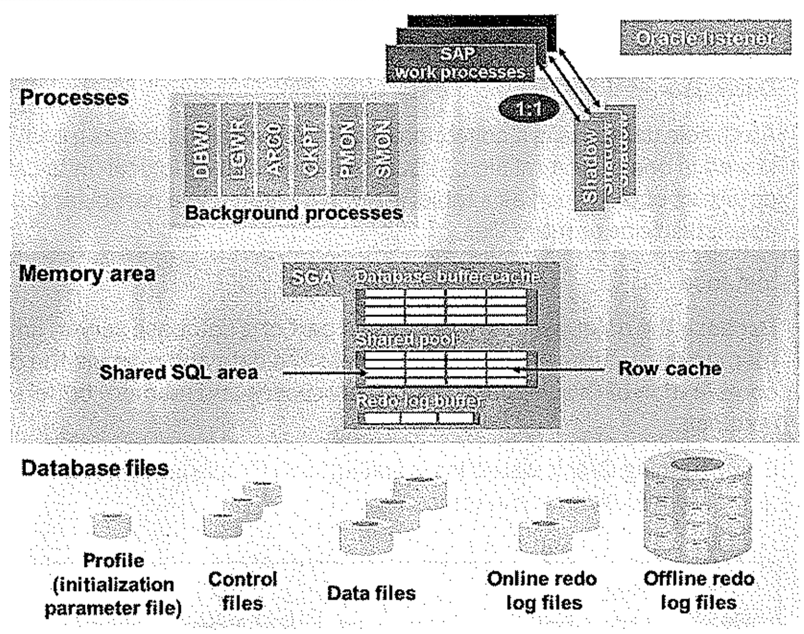


Figure 2: Architecture Overview

After an Oracle instance starts, a special process called the listener, allows the database clients and the instance to communicate with each other.

> Note:
> The listener process is not part of an Oracle instance; it is part of networking processes that work with Oracle.

In SAP installations, dedicated servers are used. When a work process makes a request to connect to the database, the listener creates a dedicated server process and establishes an appropriate connection.

Characteristics of a server process are as follows:

- A shadow process is the separate server process created on behalf of each work process (typically, for each user process).
- To handle database requests from several SAP system users, a work process communicates with its corresponding shadow process.

When a work process loses its connection with the database system, the work process automatically reconnects to the database system when the database server is available again and a database request is to be processed.

Oracle background processes perform various tasks required for the database to function properly

Databases are stored in data files on disks. To accelerate read and write access to data, the files are cached in the database buffer cache in the SGA.

The Oracle database management system (DBMS) holds the executable Structured Query Language (SQL) statements in the shared SQL area (also called the shared cursor cache), which is part of the shared pool allocated in SGA. Another part of the shared pool, called the row cache, temporarily stores Oracle data dictionary information.

Caching of Data

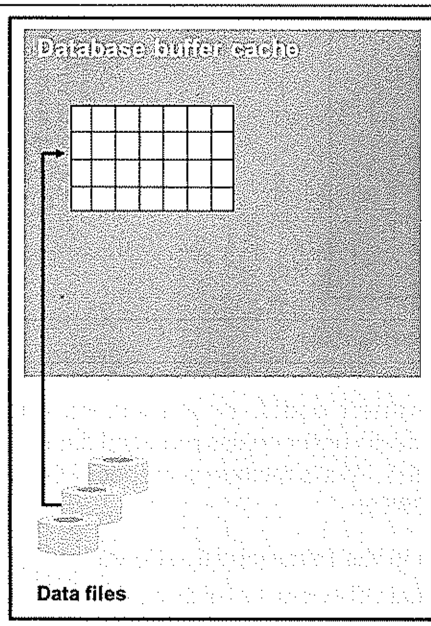


Figure 3: Oracle Architecture – Caching of Data

Databases are stored in data files on hard disks. However, data is never processed on the disks themselves. Regardless of whether a database client only needs to read some data or modify it, the associated shadow process copies the data from the disk to the database buffer cache in the SGA (if the data is not already in the database buffer cache).

Data is always cached if the data is accessed for the first time after an Oracle instance starts. But because all users concurrently connected to the instance share access to the database buffer cache, any user can reuse copies of data read from data files into the buffer cache.

The smallest logical unit that Oracle uses for copying data between data files and the buffer cache, and for managing data in the cache, is the data block.

You can choose the size of Oracle data block while creating a database. In SAP installations, however, the data block size is always 8 kB. For performance reasons, the physical allocation unit size on disks on which Oracle files are stored must also be 8 kB.

Oracle always keeps the most recently used data blocks in the buffer cache. Depending on the size of the buffer cache, Oracle must sometimes overwrite the least recently used (LRU) data blocks in the buffer cache.

Writing of Modified Data

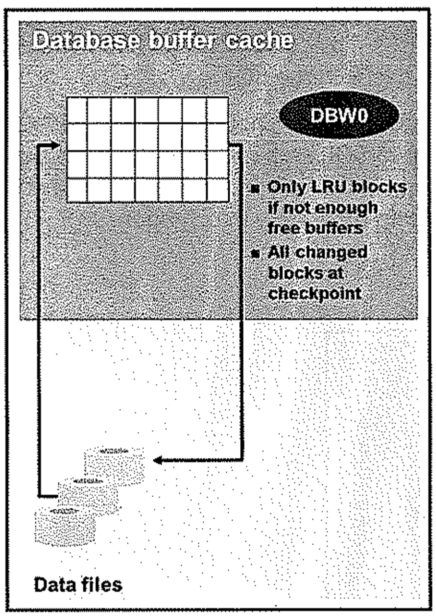


Figure 4: Oracle Architecture – Writing of Modified Data

Any changes to Oracle data, such as an insert, update, or deletion, are always performed in the buffer cache. An Oracle shadow process never copies modified data blocks ("dirty blocks") from the buffer cache to the disk. This is the task of a special Oracle background process called the database writer (DBW0).

The DBW0 writes dirty blocks to disk in the following situations:

- Buffers in the buffer cache that contain dirty blocks (dirty buffers) cannot be reused until these blocks are copied back to disk. When a shadow process needs to copy data from disk to the buffer cache, it first scans the cache for non-modified, reusable buffers. If the number of scanned buffers reaches a certain threshold, the shadow process signals the database writer to start writing some of the modified blocks to disk. The database writer then copies those dirty blocks that are on the list of least recently used blocks (LSU list), thus making them available.

- At specific times, the database writer writes all modified buffers in the SGA to data files. This event is called checkpoint, and the checkpoint process (CKPT) triggers the database writer to perform the process.

Using the concept of deferred writes, rather than immediate writes, improves efficiency because, in many cases, the database writer performs several changes on same block before the block is copied to a disk. Also, the database writer performs multiblock writes in a batch(ed) style to increase IO efficiency.

Logging of Modifications

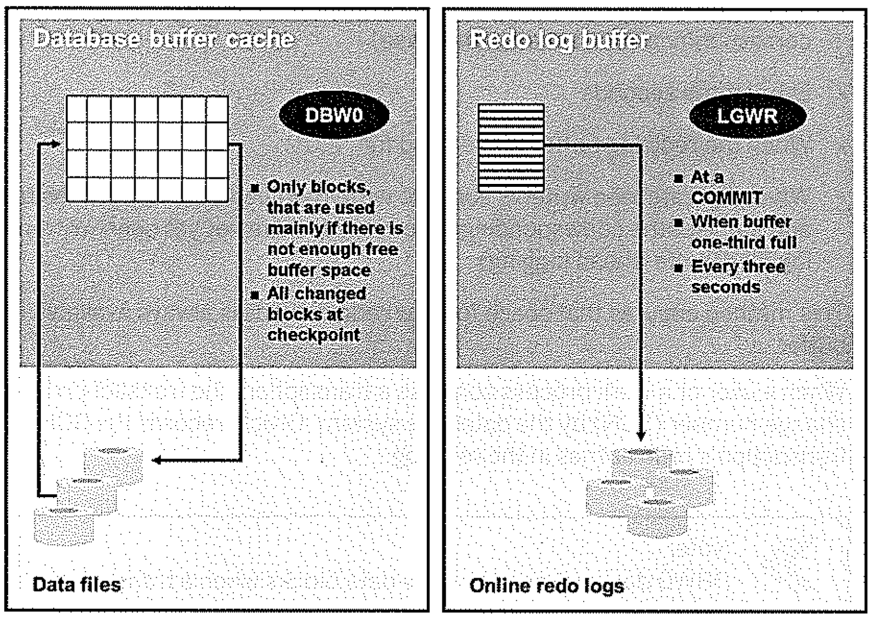


Figure 5: Oracle Architecture – Logging of Modifications

Based on deferred writes, a mechanism is needed to prevent data loss and avoid data inconsistencies, in case of a failure of any system component, such as disk, an Oracle instance, or a server.

Typically, each RDBMS logs all data changes in a log area. The log is then written to disk at appropriate times, typically after the commit of a database transaction, so that all data block changes are logged.

A database transaction is a logical unit of work (LUW) for a database server. An LUW is always treated as an atomic unit, which means it must either be processed completely or not processed at all.

To achieve data consistency and read consistency, Oracle maintains redo entries for roll forward or redo recovery (for example, after a crash), and maintains undo entries to roll back uncommitted transactions.

Redo Entries

Redo entries contain the information necessary to reconstruct, redo, or roll forward changes made to the database by SQL statements within a committed transaction. Redo entries contain the new values of the modified data, also called "after images".

Parallel to changes made in data blocks, Oracle shadow processes write redo entries into the redo log buffer. The redo log buffer is a circular buffer in the SGA that temporarily records all uncommitted and committed changes made to the database. The Oracle background process log writer (LGWR) then writes contiguous portions of the redo log buffer sequentially to an online redo log file or group of files on disk.

The online redo log consists of four or more online redo log files. Redo entries in the online redo log can be used for the database recovery, if necessary.

The LGWR writes entries from the redo log buffer to the online redo log at the following times:

- When any transaction commits (the LGWR also writes a special commit record for the corresponding transaction in this case)
- Every three seconds
- When the redo log buffer is one-third full
- When the database writer is about to write modified buffers from the block buffer to disk and some of the corresponding redo records have not yet been written to the online redo log files (write-ahead logging)

This method ensures that space for new redo records is always available in the redo log buffer.

When a user or a work process commits a transaction, the transaction is assigned a system change number (SCN) by the database system. Oracle records the SCN, along with the transaction's redo entries in the redo log.

> Note:
> DBW0 does not need to write data blocks when a transaction commits, because Oracle uses write-ahead logging.

Undo Entries

Undo entries contain the information necessary to undo, or roll back, any changes to data blocks that have been performed by SQL statements which have not yet been committed. Undo entries contain the old values of the modified data, also called "before images".

Oracle stores undo information (old values of modified data), called "before images" in a special undo segment that is separate from the redo log.

The Oracle undo space consists either of an undo tablespace or of rollback segments. An undo tablespace solution is called automatic undo management, and only the undo tablespaces must be created. The rollback segments are called manual undo management, and rollback segments must be allocated in a tablespace and managed.

The undo information of a transaction is retained in the undo space at least until the end of the transaction. The undo information can be overwritten only after the transaction has been committed.

During database recovery, Oracle performs the following steps:

1. Applies all changes recorded in the redo log, which includes the recovery of changes in the undo space
2. Uses the undo information to roll back any uncommitted transactions

Moreover, Oracle can use the undo entries for other purposes, including the reading of snapshots of consistent data (accessing "before images" of data blocks changed in uncommitted transactions).

Log Switch

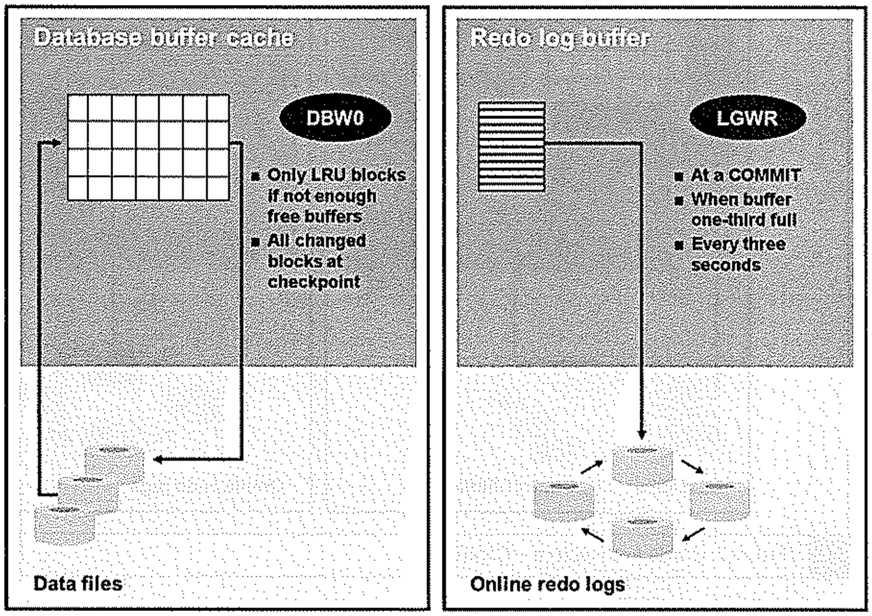


Figure 6: Oracle Architecture – Log Switch

Oracle redo log files have a fixed size (typically 50 MB on SAP systems). The redo log files do not dynamically grow when more space is needed for redo entries. Instead, when the current online redo log file becomes full, the log writer process closes the file and starts writing into the next one. This procedure is called a log switch.

The characteristics of redo log files are as follows:

- The predefined collection of online redo log files (four files in the example shown in the figure) is used in a cycle.
- At every log switch, Oracle increases the log sequence number (LSN). Through the LSN, Oracle automatically creates a sequential numbering of redo logs.
- The online redo log file into which the LGWR is currently writing is called the current online redo log file.

Control Files

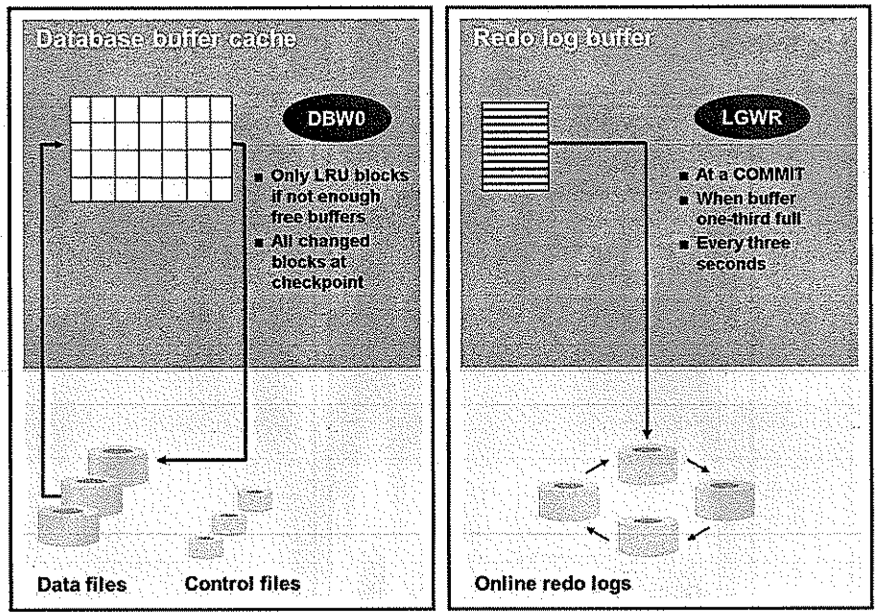


Figure 7: Oracle Architecture – Control Files

Every Oracle database has a control file, which is a small binary file necessary for the database to start and operate successfully.

A control file contains entries that specify the physical structure and state of a database, such as tablespace information, names and locations of data files and redo log files, and the current LSN.

If the physical structure of the database changes (for example, when creating new data files or redo log files), Oracle updates the control file automatically.

The characteristics of control files are as follows:

- Only Oracle can change the control files. No DBA or any other user can edit the control file directly.
- After opening the database, the control file must be available for writing. If the control file is not accessible, the database cannot function properly.
- Oracle control files can be mirrored for security reasons. Several copies can be stored at different locations, and Oracle updates them at the same time. In SAP installations, three copies of the control file are stored. Each copy must be created on physically separate disks.

Checkpoints

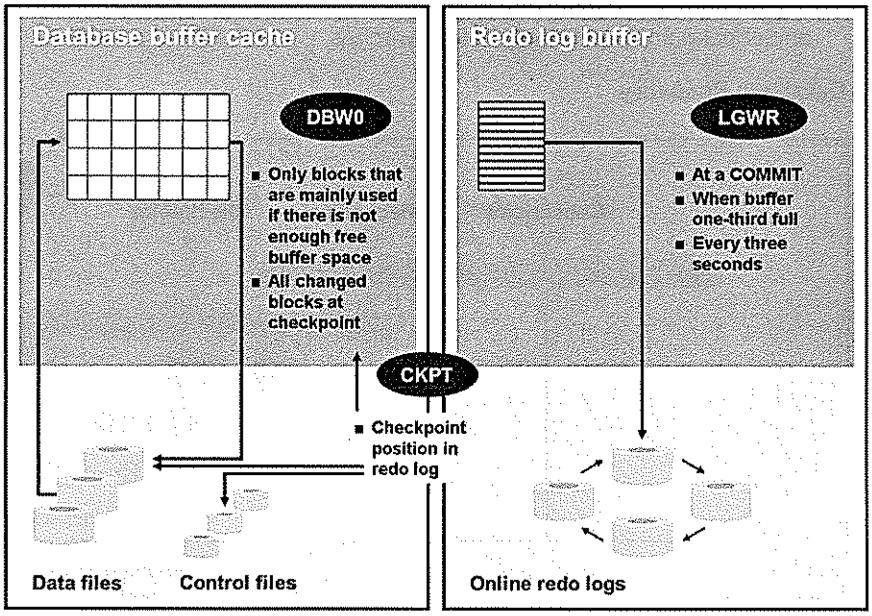


Figure 8: Oracle Architecture – Checkpoints

The checkpoint is the point at which the database writer writes all changed buffers in the buffer cache to the data files. Checkpoint is also a position in the redo log.

To activate the DBWO, the CKPT performs the following steps:

1. The DBWO receives a signal at specific times from the background CKTP process to perform this action.
2. DBWO copies all buffers that are dirty at that moment to disk. Before DBWO finishes this task, other blocks in the buffer cache can become dirty.
3. When the checkpoint event finishes, the oldest dirty buffer in the buffer cache determines a point in the redo log and from this time, recovery must begin if a crash occurs. This log position is also called a checkpoint.

In addition to activate the DBWO, the CKTP also performs the following tasks:

- Writes checkpoint information to the data file header
- Writes information about the checkpoint position in the online redo log into the control file

The CKTP does not write blocks to disk, because this is the task of DBWO.

The information about the checkpoint position in the online redo log in the control file is needed for instance recovery. This checkpoint position information tells Oracle that all redo entries recorded before the checkpoints are not necessary for database recovery because they are already written to data files.

The frequency of checkpoints is one of the factors that influences the time required for the instance to recover from a failure. The less frequent the checkpoints, the more time the instance needs for recovery.

The characteristics of a checkpoint are as follows:

- A checkpoint always occurs at a log switch.
- Frequency of checkpoints can be specified with help of Oracle parameters. In SAP installations, these Oracle parameters have values such that they are effectively not used, and checkpoints occur only at log switches.

Database Recovery

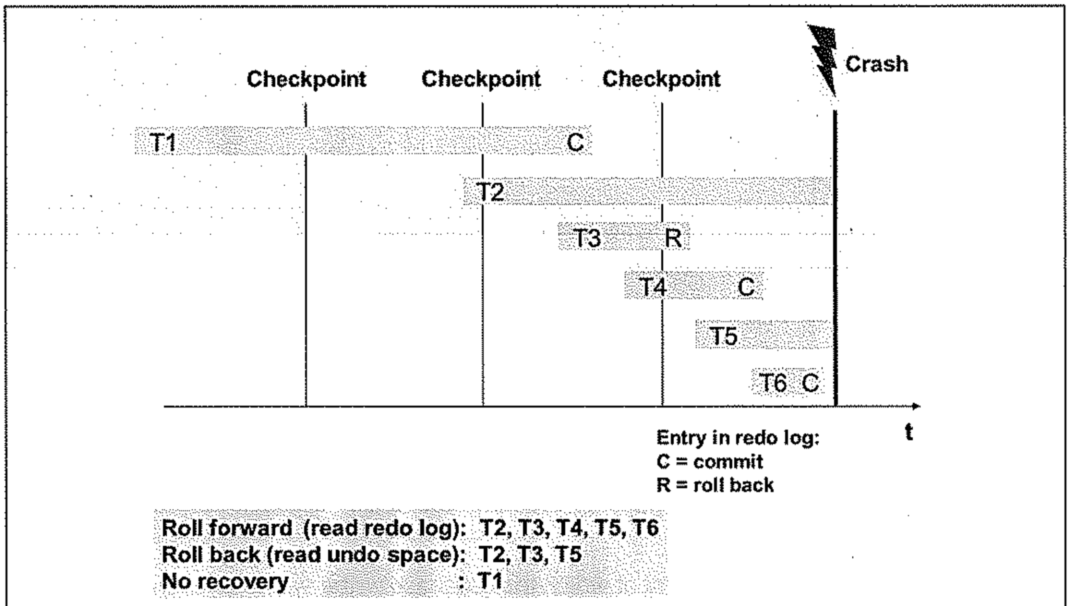


Figure 9: Oracle Architecture – Database Recovery

Online redo logs play an important role when starting an Oracle instance and opening the database, especially after a crash or when the instance does not shut down cleanly.

In such situations, Oracle recognizes that the database did not shut down properly, then automatically initiates database recovery, also called instance recovery.

Automatic recovery at restart consists of the following phases:

- Starting at the checkpoint position, redo entries are read from the online redo log and transactions are reprocessed (roll forward). This includes the roll forward of changes in the undo space.
- For every transaction that was either uncommitted at the time of the crash or rolled back explicitly before the crash (so that there is no commit entry for it in the redo log), a rollback is performed with the help of "before images" read from the undo space. Oracle ensures that this is always possible because Oracle never deletes undo entries of open transactions from the undo space.

The result is a consistent database containing only changes committed before the crash.

The following transactions are used in the example in the figure 9:

- Transaction T1 is not relevant for redo or undo because it was committed at the time of the last checkpoint. Changes to Transaction T1 are written to disk at the last checkpoint.
- Transactions T2, T3, T4, T5, and T6 are redone because they caused changes in the database after the last checkpoint. However, among these, only the changes to T4 and T6 are committed, which means only these changes are persistent.

- Transactions T2, T3, and T5 are rolled back.

Redo Log Mirroring

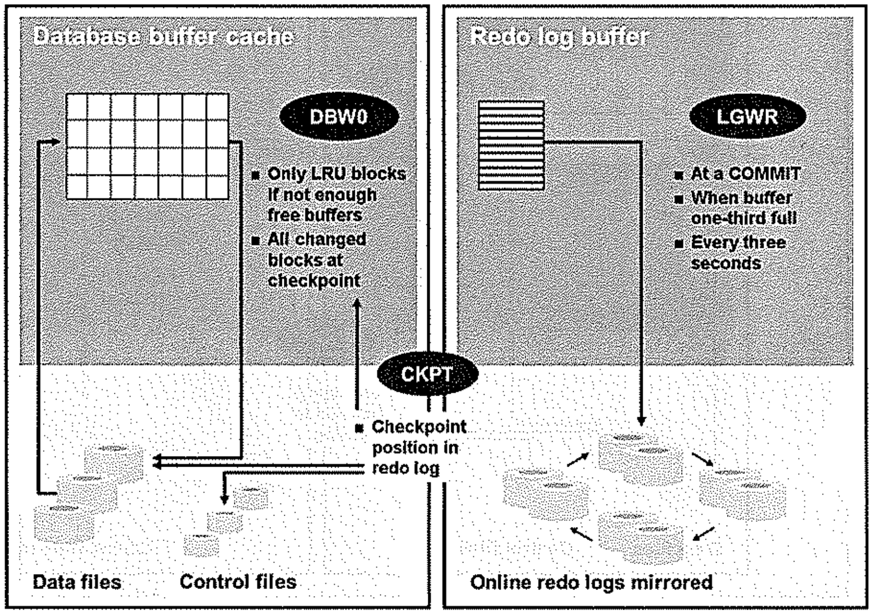


Figure 10: Oracle Architecture – Redo Log Mirroring

From a data security point of view, the online redo logs are one of the most critical areas in an Oracle server. If you lose the online redo logs in a crash, a complete recovery of the database is not possible and the result is the loss of data.

> Caution:
> Online redo logs must always be mirrored; that is, two or more copies of each redo log must be maintained on different disks.

Oracle can mirror online redo logs. This feature is used in SAP installations by default, so that there is no need for a software solution or a redundant array of independent disks (RAID). From the data security point of view, it does not matter which solution you choose. Even a combination of both Oracle and RAID mirroring is feasible to minimize the risk of losing an online redo log.

Archiving

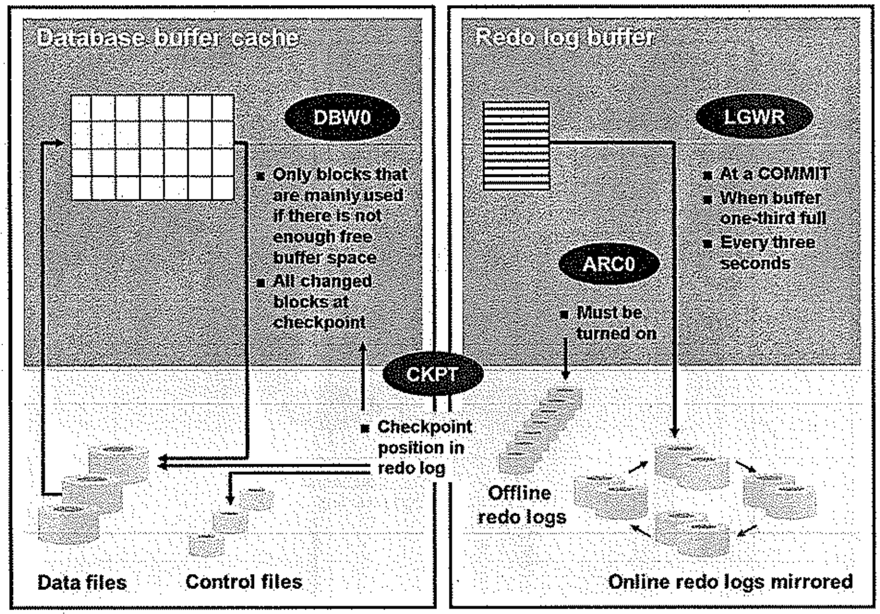


Figure 11: Oracle Architecture – Archiving

Because the online redo log is limited in size and cannot grow automatically, Oracle must overwrite old redo entries before it can write new ones.

Only the oldest redo entries up to the checkpoint position in the log, which corresponds to data changes that have already been written to data files, can be overwritten. This ensures that automatic instance recovery after a crash is always possible.

However, when you must restore data files after a disk crash and recover them manually (usually to the state of the files at the point of the crash), you need both a database backup and all the redo information written after the database backup. In an SAP system, log switches occur every few minutes so that online redo log files are reused very frequently. To prevent loss of redo information, the information must be copied from online redo log files to a safe location before overwriting. This is the task of a special Oracle background process called archiver (ARC0).

Archiving must be explicitly activated by turning on the ARCHIVELOG mode of the database.

When the ARCHIVELOG mode of the database is turned on, the following steps occur:

1. The archiver process starts automatically.
2. The archiver process copies a newly written online redo log file (after a corresponding switch to the next online redo log file) to an offline redo log file. Overwriting old redo log entries in online redo logs is not allowed before the entries are copied to offline redo logs.
3. Once an offline redo log file is successfully created as a copy, the corresponding online redo log file is released to be overwritten with new log information. The directory where offline redo log files are created can be specified through an Oracle parameter.

> Caution:
> Archiving must be activated in productive systems. Moreover, offline redo log files must be stored on a mirrored disk to prevent loss of redo information. A RAID system can be used for this purpose.

In an SAP system, the activation of archiving is the default setting. The SAP tool BRSPACE supports deactivation of archiving by changing the database log mode to NOARCHIVELOG (required, for example, during a system upgrade).

> Caution:
> If you lose a disk containing offline redo logs and data files after a crash, complete recovery is no longer possible. Therefore, offline redo logs and data files must be stored on different disks.

Other Background Processes

The system monitor (SMON) and the process monitor (PMON) are also background processes that always run in an Oracle instance.

The SMON performs the following functions:

- Performs recovery at instance startup, if necessary
- Writes alert log information if any other instance process fails
- Cleans up temporary segments that are no longer in use

The PMON performs the following functions:

- Monitors shadow processes
- Rolls back its non-committed transaction, stops the corresponding shadow process, and frees resources that the process is using, if a client process crashes

Oracle Directory Structure in SAP

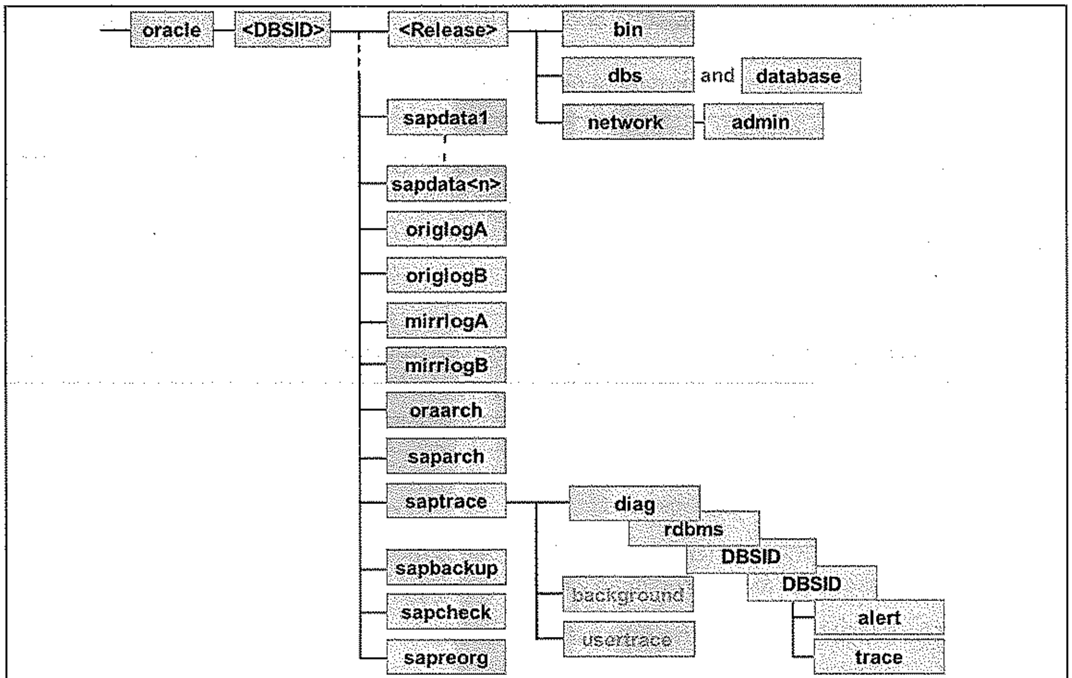


Figure 12: Oracle Directory Structure on SAP Systems

The Oracle directory and file names are standardized in SAP environments. Directories are always created with similar structure and naming conventions during installation. You cannot change the structure and you must observe all naming conventions. SAP tools for Oracle administration depend on these standardized structures and naming conventions.

Various parts of the Oracle directories and files must be physically separated from each other for performance and data security reasons. On UNIX systems, the Oracle directories appear as a tree structure because the file systems created on the physical disks are mounted on directories. On Windows there are several \oracle\<DBSID> folders on different disks with different drive letters.

Hint:
On UNIX, the <Release> subdirectory under /oracle/<DBSID> also contains information on whether you use a 32-bit or 64-bit version of Oracle (for example, /oracle/<DBSID>/112_64 for a 64-bit Oracle 11.2).

Oracle Directories and Files on SAP Systems

Directory	Contents	File name examples
<Release>		
bin	Oracle executables	
dbs or database	SAP and Oracle profiles	init<DBSID>.ora, spfile<DBSID>.ora, init<DBSID>.sap
Network		
admin	Listener or client configuration files	listener.ora, tnsnames.ora
sapdata1 . . sapdata<n>	Data files	<DBSID>.DATA1, SYSTEM.DATA1 ROLL.DATA1, ctrl<DBSID>.dbf
origlogA	Online redo log files	log_g11m1.dbf, log_g13m1.dbf
origlogB	Online redo log files	log_g12m1.dbf, log_g14m1.dbf
mirrlogA	Online redo log files	log_g11m2.dbf, log_g13m2.dbf
mirrlogB	Online redo log files	log_g12m2.dbf, log_g14m2.dbf
oraarch	Offline redo log files	<DBSID>arch1_<LSN>.dbf
saparch		arch<DBSID>.log
saptrace		
diag		
rdbms		
<dbsid>		
<dbsid>		
alert	Oracle alert file (XML) Oracle alert file	log.xml alert_<DBSID>.log
trace	Trace file of server processes Trace file of background processes	<DBSID>_ora_<PID>.trc <DBSID>_<process>_<PID>.trc
sapbackup	BRBACKUP / BRRESTORE / BRRECOVER logs	
sapcheck	BRCONNECT logs	
sapreorg	BRSPACE logs, default compression directory	

Figure 13: Oracle Directories and Files on SAP Systems

The Oracle directory and some examples of file names on SAP systems are as follows:

- `dbs` (on UNIX) or `database` (on Windows)

  The Oracle profile `init<DBSID>.ora` or `spfile<DBSID>.ora` holds the Oracle instance configuration parameters.

  The profile `init<DBSID>.sap` holds configuration parameters for administration tools BR*Tools.

- `sapdata<n>`

  Contains the data files of the tablespaces.

- `origlogA/B` and `mirrlogA/B`

  Online redo log files reside in the `origlog` and `mirrlog` directories.

  The log file numbers 1 and 3 and their mirrors reside in `origlogA` and `mirrlogA` and log file numbers 2 and 4 and their mirrors reside in `origlogB` and `mirrlogB` respectively.

- `oraarch`

  Offline redo log files are written to the `oraarch` directory and their names are specified with the help of Oracle instance configuration parameters (an example of a name is `<DBSID>arch1_<LSN>.dbf`).

- `saptrace`

  Oracle dump files are written in the `saptrace` directory. The Oracle alert log `alert_<DBSID>.log` occurs in the directory `/oracle/<DBSID>/saptrace/diag/rdbms/<DBSID>/<DBSID>/trace`.

- `saparch`

  Logs written by the SAP tool BRARCHIVE are stored in the `saparch` directory.

- `sapbackup`

  Logs written by the SAP tools BRBACKUP, BRRESTORE, and BRRECOVER are stored in the `sapbackup` directory.

- `sapreorg`

  BRSPACE creates logs for its different functions in this directory.

- `sapcheck`

  BRSPACE creates logs for its different functions in this directory.

Oracle Directories and Environment Variables

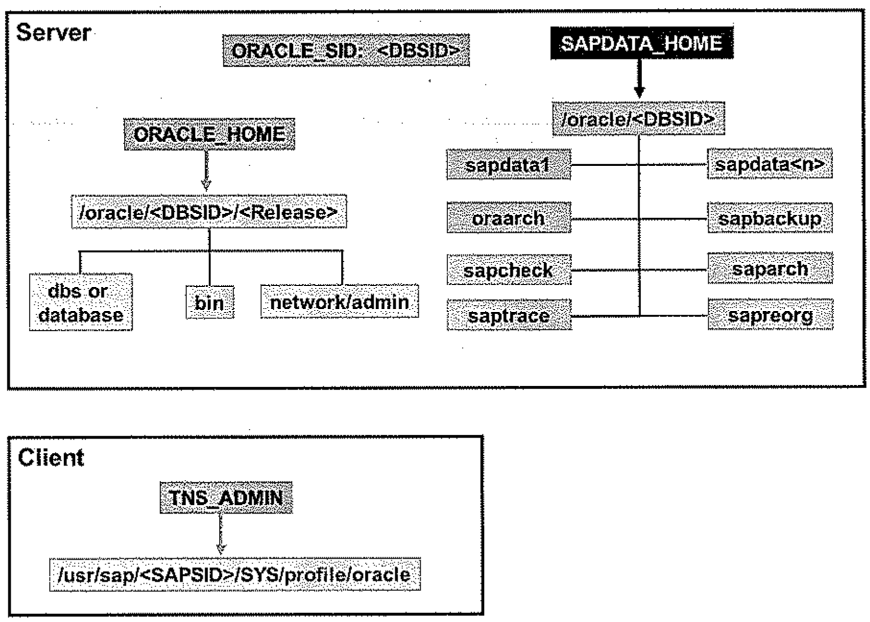


Figure 14: Oracle Directories and Environment Variables

On the database server, the environment variables `ORACLE_SID`, `ORACLE_HOME`, and `SAPDATA_HOME` must always be set for the user `<sapsid>adm` on UNIX and Windows and for the user `ora<dbsid>` on a UNIX platform.

The environment variables available on the database server are as follows:

- `ORACLE_SID`

  This is the system ID of the database instance (DBSID).

- `ORACLE_HOME`

  This is the home directory of the Oracle software and it points to the directory that contains the subdirectories `bin`, `dbs` (or `database`), and `network`. This means that the Oracle profile `init<DBSID>.ora` or `spfile<DBSID>.ora` is always located in `$ORACLE_HOME/dbs` on UNIX, and in `%ORACLE_HOME%\database` on Windows.

- `SAPDATA_HOME`

This points to the directory in which the database files are stored.

> Hint:
> The location of the control files and of the offline redo logs is configured in the Oracle profile `init<DBSID>.ora` ; the location of all other files (data files, online redo logs, and so on) is stored in the database itself. Therefore, `SAPDATA_HOME` is mainly used by BR*Tools to offer suitable directories, for example, when new tablespaces or data files need to be created.

- Other variables

  There are also other variables you can set if the corresponding directories do not have any subdirectories of `SAPDATA_HOME`. This is often the case on Windows systems because of the different drive letters, which are `SAPARCH`, `SAPBACKUP`, `SAPCHECK`, `SAPREORG`.

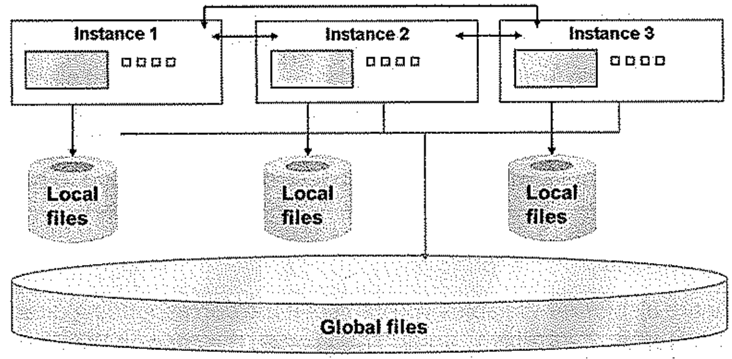
On an Oracle client (especially on every SAP application server), the variable `TNS_ADMIN` must also be set so that the Instant Client finds the Oracle Net Services profile files (sqlnet.ora, tnsnames.ora, and listener.ora). `TNS_ADMIN` is set to the path `/usr/sap/<SAPSID>SYS/profile/oracle`.

In a UNIX environment, the environment variables `ORA_NLS10` are also set for the user `ora<dbsid>`. The default value for `ORA_NLS10` is `$ORACLE_HOME/nls/data`. Since `ORACLE_HOME` is set, `ORA_NLS10` does not need to be set.

For the user `<sapsid>adm`, the environment variable `ORA_NLS10` is not set or must not be set. The Oracle instant client downloads the National Language Support (NLS) data from a dynamic library (NLS library), which is stored in the Instant Client directory.

For information on how to set this variable correctly for Oracle 10g and Oracle 11g, see SAP Note 830578. For earlier Oracle releases, see SAP Note 180430 and the other SAP Notes referenced there.

Oracle RAC



File type	Global	Local
Data files	x	
Control files	x	
spfile<DBSID>.ora / init<DBSID>.ora	x	
Online redo logs		x
Archive logs		x
Alert log		x

Figure 15: RAC

Install your SAP system in an Oracle RAC environment for the following reasons:

- To improve performance
- To increase throughput
- To deliver high availability at the same time

RAC overcomes the restrictions of normal failover solutions with the following instances:

- Concurrent processing
- Load balancing
- Fast and reliable detection of a node or network failure
- Fast recovery

Although you use just one active database instance with a standard cluster solution, RAC makes it possible to use several instances simultaneously. All active instances execute transactions against a shared database.

To provide data consistency and data integrity, RAC coordinates the access of each instance to the shared data as follows:

- The Transport Network Service (TNS) listener files provide automated load balancing across all nodes in the cluster.
- The load balancing feature automatically adjusts for cluster configuration changes. For example, if you add a node to your cluster database, Oracle updates all the listener files in the cluster with the new node's listener information.

There is no need to make code changes to deploy applications on RAC if the applications run on single instance Oracle configurations. No adjustments to the SAP application are needed for RACs.

RACs require that all nodes have simultaneous access to the shared disks to give the instances concurrent access to the database. The implementation of the shared disk subsystem is based on the OS you choose; you can either use a cluster file system or place the files on raw devices. However, cluster file systems greatly simplify the installation and administration of RACs.

Apart from using a shared database, RACs coordinate the buffer caches of multiple instances on different nodes. This optimizes performance and expands the effective memory to be nearly equal to the sum of all memory in your cluster database.

RACs also support all Oracle backup and archiving features that are available in the single-instance Oracle databases. This includes both online and offline backups of either an entire database or individual tablespaces.

Migration to Oracle RACs is relatively easy, because the unloading and loading of data is unnecessary.

Unit 1
Exercise 1

Analyze Oracle Environment Variables

Business Example

You log on to the database server and call Oracle tools. Unexpected error messages appear, possibly because environment variables are not set or set incorrectly. You want to know what is causing these error messages.

Change Oracle environment variables and observe the consequences.

1. Log on to the database server with the logon information provided by your instructor.
2. Check the environment variables set, especially Oracle variables starting with `ORA`.
3. Change variable `ORACLE_HOME` to any other path and call the Oracle tool TNSPING to ping your <DBSID>. Check the output and explain the reason for any unexpected output.

Unit 1
Solution 1

Analyze Oracle Environment Variables

Business Example

You log on to the database server and call Oracle tools. Unexpected error messages appear, possibly because environment variables are not set or set incorrectly. You want to know what is causing these error messages.

Change Oracle environment variables and observe the consequences.

1. Log on to the database server with the logon information provided by your instructor.
2. Check the environment variables set, especially Oracle variables starting with `ORA`.
   a) Call the `set` command to display environment variables.

   > Note:
   > Paths, drive letters, and system IDs can be different than in the following output.

```
G:\oracle\T99>set
...
ORACLE_HOME=g:\oracle\DEV\112
ORACLE_SID=T99
...
SAPARCH=G:\oracle\DEV\saparch
SAPBACKUP=G:\oracle\DEV\sapbackup
SAPCHECK=G:\oracle\DEV\sapcheck
SAPDATA_HOME=G:\oracle\T99
SAPEXE=G:\usr\sap\DEV\SYS\exe\run
SAPLOCALHOST=twdf9999
SAPREORG=G:\oracle\DEV\sapreorg
SAPTRACE=G:\oracle\DEV\saptrace
```

3. Change variable `ORACLE_HOME` to any other path and call the Oracle tool TNSPING to ping your <DBSID>. Check the output and explain the reason for any unexpected output.
   a) Call the `TNSPING` command without making any changes and check whether you can access your database.

```
D:\oracle\T99>tnsping T99

TNS Ping Utility for 64-bit Windows: Version 11.2.0.1.0 -
Production on 31-MAI-2011 15:09:57

Copyright (c) 1997, 2010, Oracle.  All rights reserved.

Used parameter files:
D:\oracle\DEV\112\network\admin\sqlnet.ora
```

```
Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS =
(PROTOCOL = TCP)(HOST = TWDF1825)(PORT = 1527))) (CONNECT_DATA =
(SERVICE_NAME = T99)))OK (40 msec)
```

b) Change the variable `ORACLE_HOME` and call the `TNSPING` command.

The following results display:

```
D:\oracle\T99>set oracle_home=D:\

D:\oracle\T99>tnsping T99

TNS Ping Utility for 64-bit Windows: Version 11.2.0.1.0 -
Production on 31-MAY-2011 15:11:34

Copyright (c) 1997, 2010, Oracle.  All rights reserved.

Message 3511 not found; No message file for product=NETWORK,
facility=TNSTNS-03505: Message 3505 not found; No message file
for product=NETWORK, facility=TNS
```

c) Oracle works internally with message numbers, whose message texts are stored in a subdirectory of `%ORACLE_HOME%`. If `ORACLE_HOME` is not set correctly, texts cannot be displayed.

Although TNSPING does not display a useful reason for the unexpected output, SQLPLUS displays as follows:

```
G:\oracle\T99>sqlplus /nolog
Error 6 initializing SQL*Plus
Message file spl&lt;lang>.msb not found
SP2-0750: You may need to set ORACLE_HOME to your Oracle
software
directory
```

d) Change the `ORACLE_HOME` variable back to your original value as follows:

```
G:\oracle\T99>set ORACLE_HOME=d:\oracle\DEV\112
```

LESSON SUMMARY

You should now be able to:

- Analyze Oracle environment variables

Unit 1
Lesson 2

Connecting to the Database

LESSON OVERVIEW

This lesson explains how SAP connects to an Oracle database over a network.

Business Example

A database administrator (DBA) changed the password of the default database user SAP<SCHEMA-ID> using Oracle commands. Afterward, the SAP system could not be started. You need to know how to correctly change passwords. For this reason, you require the following knowledge:

- An understanding of the default operating system (OS) and database users
- An understanding of Oracle communication over a network (NET Services)
- An understanding of the function of the Oracle listener
- An understanding of how to start and stop the Oracle listener

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Establish connection to the database

Database Connectivity

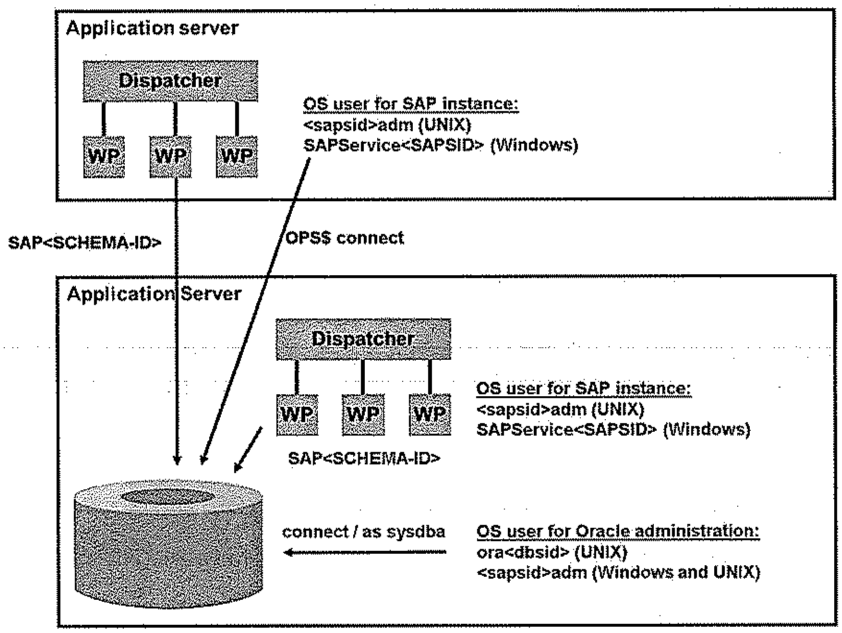


Figure 16: OS and Database Users

To safeguard your SAP system, you must control user access at the following levels:

- OS
- Database
- SAP system

Database users are required in the SAP environment in the following scenarios:

- The SAP system connects to the database during operation.
- The DBAs connect to the database to perform administrative actions.

Oracle System Privileges

Oracle System Privileges: Examples

Privilege name	Operations authorized
CREATE SESSION	Connect to the database
CREATE TABLESPACE	Create a new tablespace for the database
ALTER SYSTEM	Issue ALTER SYSTEM statements
...	

Special System Privileges

Privilege name	Operations authorized
SYSDBA	STARTUP, SHUTDOWN, CREATE DATABASE, ARCHIVELOG, BACKUP, RECOVERY, ...
SYSOPER	As for SYSDBA but without CREATE DATABASE and without the ability to look at user data

Figure 17: Oracle System Privileges

Operations in Oracle are controlled by system privileges and object privileges.

One or two special system privileges are required for database administration.

Object privileges protect access at the object level so that you can set the right to send queries to tables or views (for example, SELECT) and perform Data Manipulation Language (DML) operations (for example, INSERT, UPDATE, and DELETE).

System privileges restrict actions performed by database users on the instance or database level. There are over 100 system privileges in Oracle.

The special system privileges SYSDBA and SYSOPER can be thought of as types of connections. In an SAP system, you use OS authentication to connect to Oracle with the privileges SYSDBA or SYSOPER.

SYSDBA and SYSOPER give you special administrative privileges and allow access to a database instance, even when the database is not open, enabling you to perform certain database operations for which authorizations cannot otherwise be granted. Control of these access privileges is totally outside of the database.

OS Users and Groups

OS Users and Groups in an SAP System with Oracle

UNIX environment

OS user	Oracle-relevant OS group	Privileges in Oracle
ora<dbsid>	dba oper	Full administration of all instances Restricted administration of all instances
<sapsid>adm	dba oper	Full administration of all instances Restricted administration of all instances

Windows 2000/2003 environment

OS user	Oracle-relevant OS group	Privileges in Oracle
<sapsid>adm	ORA_<DBSID>_DBA ORA_<DBSID>_OPER ORA_DBA	Full administration of instance Restricted administration of instance Full administration of all instances
SAPService<SAPSID>	ORA_<DBSID>_DBA ORA_<DBSID>_OPER	Full administration of instance Restricted admin of the instance ORA_DBA Full administration of all instances

Figure 18: OS Users and Groups in an SAP System with Oracle

In SAP systems with Oracle, special OS users are created during installation.

The special OS users have privileges for administration and maintenance of the Oracle database at the following levels:

- Special OS users can access the Oracle instance directories and files and call database administration tools at the OS level.
- Special OS users can connect to the Oracle instance with special database users, and either perform administrative work or maintain SAP objects and data in the database.

Oracle can move the database security mechanism to the OS level by using certain mappings between OS users and database users, or by using mappings between OS groups and system privileges.

Examples of the mappings that Oracle uses are as follows:

- Members of the OS group `dba` (UNIX) or `ORA_DBA` or `ORA_<DBSID>_DBA` (Windows) can connect to the Oracle instance with system privilege SYSDBA and perform administrative work there.
- Members of the OS group `oper` (UNIX) or `ORA_<DBSID>_OPER` (Windows) can connect to the Oracle instance with system privilege SYSOPER and perform corresponding administrative work there.

> Hint:
> A member of `ORA_DBA` can connect to and administer any Oracle instance installed on the corresponding host.

Oracle Database Users

Standard Database Users in Oracle

User name	Default password	Description
SYS	*CHANGE_ON_INSTALL	Owner of the database's data dictionary tables and views; can perform database administration; has privileges to access and modify all database tables and data.
SYSTEM	*MANAGER	Can perform database administration; has privileges to access all database tables and data, but cannot modify data dictionary tables

*** Starting with SAP Web AS 6.40 the password for the database users are defined during installation.**

Figure 19: Standard Database Users in Oracle

Every Oracle database contains two administrative user accounts, SYS and SYSTEM, which are automatically created during installation and assigned the database role DBA.

SYS is the user with the most privileges in an Oracle database.

The SYS user has the following characteristics:

- All tables and views of the database data dictionary are stored in the SYS schema. These tables and views are important for operating Oracle. Therefore, a user or DBA must never modify them. Do not create tables in the schema of the SYS user.
- SYS users are granted additional privileges over those of the DBA role, and can access and modify all data in the database.

> Note:
> A schema is a collection of database objects belonging to a user as owner. A schema always has the same name as the owner.

SYSTEM is a username defined by Oracle for the creation of additional internal tables and views that display administrative information. Although, SYSTEM can access all database tables, it has no privilege to change Oracle data dictionary tables.

In an SAP installation, the SYSTEM user has the following characteristics:

- SYSTEM is additionally assigned the database role SAPDBA to allow BR*Tools access to certain tables of the SYS schema.
- SYSTEM is the default user when SAP tools are called for Oracle administration to create a connection to the database.

> Hint:
> Oracle user and role names, as well as user passwords, are not case sensitive unless you use them as strings enclosed in quotation marks.

Database Users in Oracle Created by SAP

Database Users in Oracle Created by SAP

OS-platform independent

User name	Default password	Description
SAP<SCHEMA-ID> or SAPR3	SAP	All SAP objects in Oracle are created here belong to the schema of this user No privileges for administration of the database Assigned role: SAPCONN

UNIX environment

User name	Description
OPS$<SAPSID>ADM	Assigned role: SAPDBA (but not DBA)
OPS$ORA<DBSID>	Assigned role: SAPDBA (but not DBA)

Windows 2000/2003 environment

User name	Description
OPS$<DOMAIN>\<SAPSID>ADM	Assigned role: SAPDBA (but not DBA)
OPS$<DOMAIN>\SAPSERVICE<SAPSID>	Assigned role: SAPDBA (but not DBA)

Operating System Authentication

Oracle parameters

REMOTE_OS_AUTHENT=TRUE **OS_AUTHENT_PREFIX=OPS$**

Operating system user	Oracle database user
<USERNAME>	OPS$<USERNAME>

OPS$ users have no passwords, they are identified externally.

Figure 20: Database Users in Oracle Created by SAP

The SAP installation always creates the Oracle user SAP<SCHEMA-ID> (or SAPR3 up to SAP Basis4.6D), where SCHEMA-ID is, in most cases, identical with the SAP system identifier (SID). All tables and indexes of the corresponding SAP system belong to the schema of this database user. However, SAP<SCHEMA-ID> does not have privileges to perform administrative tasks on the database and it is not assigned the database roles DBA or SAPDBA.

> Caution:
> During the installation of an Oracle database for an SAP system, you are asked to specify passwords for the users SYS, SYSTEM, and SAP<SCHEMA-ID>. If you do not enter anything, the system assigns users default passwords. In this case, change the passwords after the installation completes. If you do not change the passwords, your system is not sufficiently secure.

Other users created in the Oracle database by SAP make use of an Oracle feature called OS authentication. If the user `OPS$<USERNAME>` is defined as identified externally at the database level, it has no password.

The OS user <USERNAME> can connect to the database without authentication when the following Oracle parameters are set:

- REMOTE_OS_AUTHENT=TRUE

  Allows remote OS authentication for OS users on UNIX systems with an OPS$ user on any computer in the network from which you can access the database.

- OS_AUTHENT_PREFIX=OPS$

These are the default values of the parameters in an SAP system; normally, you do not have to change them.

On a Windows platform, the <USERNAME> used in the definition of the OPS$ user includes the name of the Windows domain from which the OS user originates (or the host name, if it is a local user). The remote OS authentication is performed by the domain information.

> Hint:
> As of Oracle 11g, the profile parameters `REMOTE_OS_AUTHENT` and `OS_AUTHENT_PREFIX` have changed, from the perspective database security. With `REMOTE_OS_AUTHENT=TRUE`, you see one of the following messages when starting the instance:
>
> - `ORA-32004: obsolete or deprecated parameter(s) specified for RDBMS instance`
> - `ORA-32006: REMOTE_OS_AUTHENT initialization parameter has been deprecated`
>
> In an SAP environment, SAP BR*Tools and the SAP application use the OPS$-connect mechanism, which requires the `REMOTE_OS_AUTHENT=TRUE` parameter setting. Therefore, on UNIX systems, you must ignore the error messages when starting the database. SAP replaces the OPS$ connect mechanism in long term with another secure logon mechanism.

Oracle Database Roles

Privileges are Grouped and Granted to Users Through Database Roles

Predefined Oracle database roles

Role	Description
DBA	Contains all system and object privileges needed for administration of the database
SELECT_CATALOG_ROLE	Provides the SELECT privilege on objects in the data dictionary.
...	

Database roles created by SAP

Role	Description
SAPDBA	Contains system privileges, such as ALTER SYSTEM,ALTER DATABASE, and ALTER TABLESPACE ,and object privileges enabling the assigned user to access certain tables required for database administration actions performed with SAP tools (BR*Tools)
SAPCONN	Specially-defined database role containing all database authorizations required by the SAP application (ABAP and Java stack) Independent of SAP release
...	

Figure 21: Oracle Database Roles

Within the database, system and object privileges can be pooled to database roles. If you assign a database role to a database user, the user is granted all privileges included in the role.

There are only few predefined database roles in Oracle, the most important of which is DBA. The DBA role contains the privileges carrying the ADMIN OPTION flag. However, the DBA role does not enable you to fully administer the Oracle instance (for example, to start and shut down the instance); therefore, you also need the system privilege SYSDBA or SYSOPER.

You can create any number of database roles in an Oracle database. The SAP installation creates two additional roles, called SAPDBA and SAPCONN.

Based on object privileges, the SAPDBA role allows a user to access certain tables used by SAP tools for database administration. Examples of such tables are DBCSTATC, SDBAD, DBSTATHORA, and so on. The SAPDBA role is assigned to the OPS$ user at installation. If the role assignment is accidentally deleted, you can reassign the SAPDBA role to the user using the SQL script `sapdba_role.sql` (see SAP Note 134592).

Up to, and including, Oracle Release 10.1, the CONNECT role is assigned to user SAP<Schema ID> or SAPR3. This Oracle standard role comprises a large number of database authorizations. For security reasons, as of Release 10.2, Oracle has restricted the CONNECT role to the CREATE SESSION privilege. As of Release 10.2, only application-specific database roles (SAPDBA, SAPCONN) are used for SAP database users.

When installing an SAP system with Oracle Database 10g Release 2 (10.2), the SAPCONN role is automatically assigned to the SAP database users. When you upgrade the database, you must explicitly create the SAPCONN role. The `sapconn_role.sql` script is used for this purpose, with installation instructions in the SAPEXE directory. You can also find installation instructions for the sapconn_role.sql script in the appendix to SAP Note 834917.

Hint:
You can use the SAPCONN role as an option with Oracle 9i. However, the SAPCONN role is a requirement as of Oracle 10.2.

Note:
As of Oracle Database 10g Release 2 (10.2), the limit for FAILED_LOGIN_ATTEMPTS for the DEFAULT user profile is 10. Prior to Oracle Database 10g Release 2 (10.2), the limit for FAILED_LOGIN_ATTEMPTS for the DEFAULT user profile was UNLIMITED. After an account is locked, SAP application work processes can no longer log on to the Oracle database, which can result in SAP system downtime.

To avoid the risk of downtime, you can revert the FAILED_LOGIN_ATTEMPT for the DEFAULT user profile to the less secure limit UNLIMITED.

As of Oracle 11g Release 2 (11.2), the default limit for PASSWORD_LIFE_TIME for the DEFAULT profile is 180 days; prior to Oracle 11g, the default was UNLIMITED.

For normal database users, forcing password changes makes sense from a database security perspective, but not necessarily, for an application account.

Therefore, consider using the database user profile SAPUPROF for SAP application users.

The advantages of using the SAPUPROF user profile are as follows:

- Dedicated user profile SAPUPROF
- Adaptation to the SAP application
- Customization, such as adding a password verification function, is possible
- No modification of the DEFAULT user profile is required
- No easing of Oracle's 'Secure by Default' configuration
- Greater security

For more information, see the OSS Notes 1519872, 1522952, and 951167.

Connecting to Oracle

Connection to Oracle and User Authentication

Connect request	Description
connect <DB_user>/<PW>[@<DBSID>]	This request is independent of OS user. Oracle performs user authentication.
connect /[@<DBSID>]	OS authentication: OS user is connected to the database as the corresponding OPS$ user.
connect <DB_user>/<PW>[@<DBSID>] AS SYSDBA or connect /[@<DBSID>] AS SYSDBA	OS authentication: OS user is connected to the database as the SYS user with administrative privilege of SYSDBA. OS user depends on membership in OS group (dba on UNIX).
connect <DB_user>/<PW>[@<DBSID>] AS SYSOPER or connect /[@<DBSID>] AS SYSOPER	OS authentication: OS user is connected to the database with the PUBLIC schema with the administrative privilege of SYSOPER. OS user depends on membership in OS group (oper on UNIX).

Figure 22: Connection to Oracle and User Authentication

> Note:
> The connect identifier @<DBSID> specifying the database ID is usually not necessary in an SAP environment, where database system identifier (DBSID) is known from the environment variable ORACLE_SID.

When establishing a connection to an Oracle instance, the user authentication used by Oracle depends on the type of connection request sent by the client.

The keyword CONNECT is used within the Oracle client tool SQL*Plus for interactive connections to Oracle. The connect requests described in the SQL*Plus tool can be used by other client applications, for example, by work processes of SAP instances through the database shared library for Oracle.

An OS user can successfully perform `connect /` only if the corresponding OPS$ user exists in the Oracle database.

An OS user can successfully perform `connect / {AS SYSDBA | AS SYSOPER}` only if the user is a member of the corresponding OS group (dba or oper on UNIX and ORA_DBA, ORA_<SID>_DBA, or ORA_<SID>_OPER on Windows). If the user is not a member of the OS group, the system refuses the connection request.

Connection AS SYSDBA gives you full Oracle instance and database administration privileges through system privilege SYSDBA and the privileges assigned to the SYS user, which is always used in this case.

When you connect to the Oracle database with AS SYSOPER, you are not authenticated as an explicit database user, but as the schema PUBLIC.

Connections with AS SYSDBA or AS SYSOPER replace CONNECT INTERNAL, which are used up to Oracle 8.x.

Security – SAP<SCHEMA-ID> Password

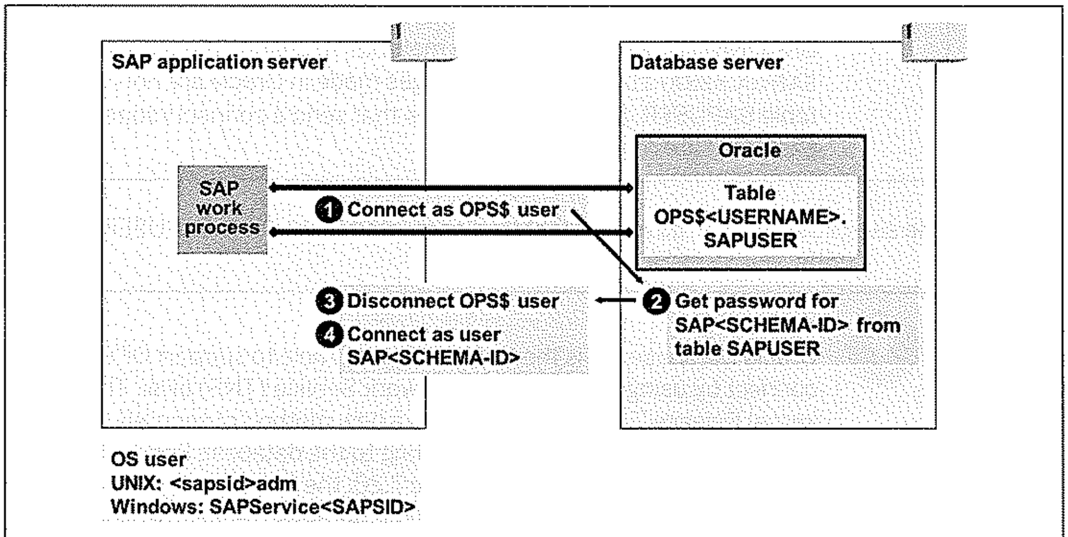


Figure 23: Work Process Connection to Oracle

SAP work processes run at the OS level in the context of user `<sapsid>adm` (UNIX) or `SAPService<SAPSID>` (Windows). The work processes must connect to Oracle with the username `<SAPSCHEMA-ID>` (or `SAPR3` up to SAP Basis 4.6D). For this reason, the Oracle user must be protected by a password. SAP has implemented a mechanism for work processes that enables the users to find this password.

This work process connection mechanism is based on storing the password of user `SAP<SCHEMA-ID>` not only in an Oracle system table, but also in a special table called SAPUSER, which is created in the schema of user `OPS$<SAPSID>ADM` on UNIX or `OPS$<DOMAIN>\<SAPSID>ADM` on Windows. On Windows, this table is also accessible by the user `OPS$<DOMAIN>\SAPService<SAPSID>`.

When a work process starts and tries to connect to the Oracle database, the SAP application performs the following steps:

1. The work process logs on to the database as its corresponding OPS$ user with OS authentication.
2. The work process sets a SELECT statement in the table SAPUSER and reads the password of `SAP<SCHEMA-ID>`.
3. The work process disconnects from Oracle.
4. The work process connects with username `SAP<SCHEMA-ID>` and retrieves the password from the table SAPUSER.

If any of the steps cannot be successfully performed, the connection cannot be established, the work process stops, and an error is reported.

> Caution:
> You can only change the `SAP<SCHEMA-ID>` password with the SAP tool BRCONNECT. Do not use Oracle methods to change the password of `SAP<SCHEMA-ID>`. Password changes made in Oracle are not reflected in the table SAPUSER, and work processes that send a connect request with the wrong password are refused by Oracle.

The `SAP<SCHEMA-ID>` password is stored encrypted in the table SAPUSER.

> Hint:
> Oracle Database 11g is the last version of Oracle to support OPS$ remote connect by using the Transport Network Service (TNS) alias. Therefore, as of kernel Release 7.20, SAP introduces a new method of securely storing the database password and for connecting to the database Secure Storage in File System (SSFS). With kernel Release 7.20, the encrypted password for the SAP database user is no longer stored in the database, but in the file system.
>
> For further information, see OSS Note 1622837.

Net Services

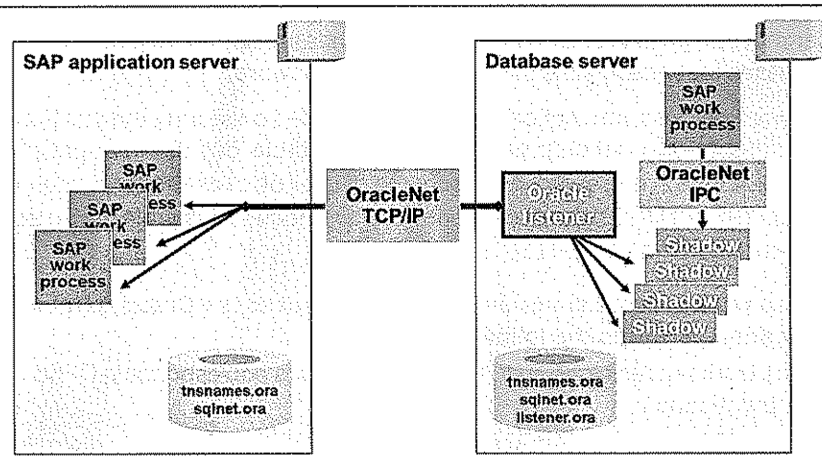


Figure 24: OracleNet Basics

If an Oracle client, such as an SAP instance, is running on a computer other than the database server, SAP work processes and their dedicated shadow processes communicate over the network. A communication protocol, Transmission Control Protocol/Internet Protocol (TCP/IP), is used and a software layer called OracleNet (Oracle Network Services) is used. The work processes of an SAP instance configured on the database server use the interprocess communication (IPC) protocol to communicate with dedicated shadow processes running on the same server.

OracleNet resides both on the client and on the Oracle database server.

OracleNet establishes and maintains the connection between the client application and server, and manages the exchange of messages between the client application and server, using standard protocols such as TCP/IP.

There is a special process called OracleNet Listener on the server which monitors incoming connection requests. When the listener receives a client request for a network session with the database server, and the client information matches the listener information, the listener forwards the requests to the server. Once a connection is established, the client and Oracle database server communicate directly with one another.

The listener is configured with a protocol address. Only clients configured with the same protocol address can send connection requests to the listener. The OS files used for this purpose are stored in the directory `$ORACLE_HOME/network/admin`.

The following OS files are used to send connection requests to the listener:

- `listener.ora`

  `listener.ora` configures the listener and is only used on the database host. This file is read when the listener is started. The configuration information specified in this file determines OracleNet settings, such as the network protocol to be used, host name, port, and the default tracing information. `listener.ora` must contain all Oracle system IDs and protocol addresses for which the listener must accept connection requests.

- `tnsnames.ora`

  `tnsnames.ora` contains a list of service names for all databases that you can access in the network.

- `sqlnet.ora`

  `sqlnet.ora` can contain client-side information, such as a client domain to append to unqualified service names or net service names, or optional diagnostic parameters used for client tracing and logging.

Oracle Listener

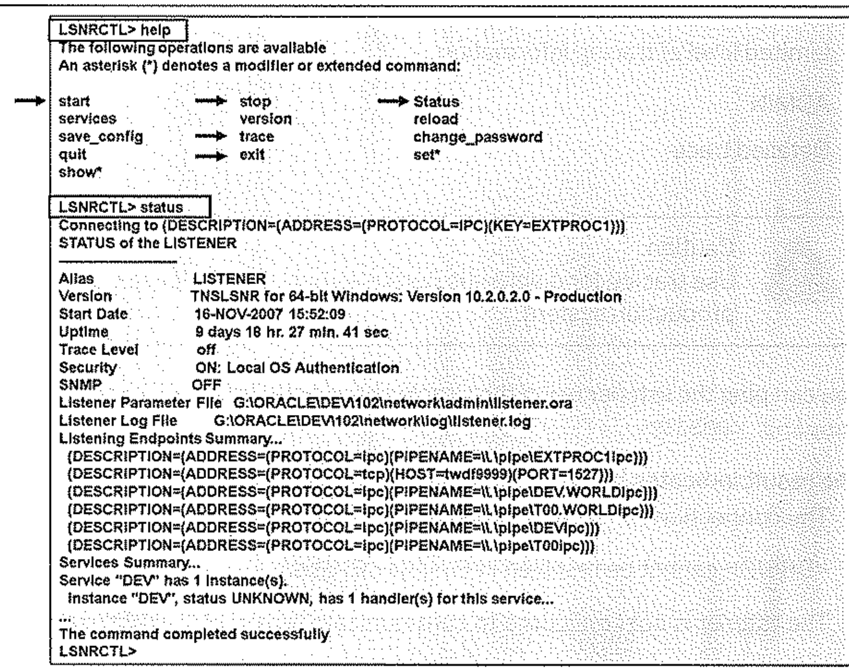


Figure 25: Managing the Listener – LSNRCTL

For OracleNet to accept connections on the database server, the listener must be running.

The Oracle utility lsnrctl is used to perform the following actions:

- Starting and stoping the listener
- Checking the status of OracleNet connections

When the listener starts, the following services start:

- In a UNIX environment, the process *tnslsnr* is started.
- In a Windows environment, the service Oracle<DBSID><Release>TNSListener is started.

If several Oracle instances are installed on one host, there is usually one listener process running on the host serving all active Oracle instances.

The command line tool LSNRCTL controls the Oracle listener.

To return a list of available commands, enter **`help`** if the lsnrctl command prompt appears.

The command lsnrctl status displays the following information:

- OracleNet version
- Listener program start time
- Location of parameter
- Listener log files

Database server listener tracing can be enabled by setting trace level information in the `listener.ora` file or by turning it on through the program LSNRCTL.

Valid options for listener tracing are as follows:

- NOT SET

  No tracing (default)
- USER

  Limited level of tracing information
- ADMIN

  Detailed trace

> Caution:
> Use tracing for diagnostic purposes only. Do not leave tracing on indefinitely in a production system.

You can ping the listener using the Oracle command TNSPING. To ping the listener, use `tnsping <DBSID>`.

The result indicates if any of the following problems occur:

- The connection can be established.
- <DBSID> cannot be resolved in `tnsnames.ora`.
- The listener is not configured to communicate with <DBSID> through `listener.ora`.
- The listener is not running on the database server.

Unit 1
Exercise 2

Establish Connection to the Database

Business Example

You need to connect to the Oracle database with SYSDBA privileges, but the connection does not work.

Connect to the database with different methods without specifying a password.

1. Call `sqlplus /nolog` to start SQLPLUS without automatically connecting to the database. Then use the connect command with different arguments to connect to the database. First perform `connect /`. Does the connection work? Explain why

2. List the users registered in the database and show the content of the table *SAPUSER*.

3. List the database roles created by SAP.

4. (Optional) List the tables in the schema of the user `SAP<Schema-ID>`.

5. Use `connect /@<SID+1>` where <SID+1> is your SID plus one (for example, if your SID is T05, use `connect /@T06`).

6. Use `connect /@<SID+1> as sysdba` where <SID+1> is your SID plus one (for example, if your SID is T05, use `connect /@T06 as sysdba`).

Unit 1
Solution 2

Establish Connection to the Database

Business Example

You need to connect to the Oracle database with SYSDBA privileges, but the connection does not work.

Connect to the database with different methods without specifying a password.

1. Call `sqlplus /nolog` to start SQLPLUS without automatically connecting to the database. Then use the connect command with different arguments to connect to the database. First perform `connect /`. Does the connection work? Explain why

   a) Call the `connect /` command.

   The OPS$ connect message displays as shown below. The OPS$ connect works because you are logged on to the OS as user `<sapsid>adm`, and database user `OPS$<DOMAIN>/<SAPSID>ADM` exists.

```
D:\oracle\T99>sqlplus /nolog

SQL*Plus: Release 11.2.0.1.0 Production on Di Mai 31 15:20:28
2011

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

SQL> connect /
Connected.
SQL>
```

   The OPS$ user can only log on to a running database. The following ORA error messages indicate that the database is not running:

```
ORA-01034: ORACLE not available
ORA-27101: shared memory realm does not exist
```

   b) If both ORA messages appear, you must start the database first. Use `connect / as sysdba` to connect with SYSDBA privileges. This connection works because the OS user you are logged on to belongs to the OS group `ORA_<DBSID>_DBA`.

   Call the `startup` command. This starts the database.

```
SQL> connect / as sysdba
Connected.
SQL> startup
ORACLE instance started.

Total System Global Area  133644288 bytes
Fixed Size                  2173360 bytes
Variable Size             109053520 bytes
Database Buffers           16777216 bytes
Redo Buffers                5640192 bytes
Database mounted.
```

```
Database opened.
SQL>
```

c) Call the `connect /` command.

```
SQL> connect /
Connected.
SQL>
```

After starting the database, you can now log on with `connect /`.

2. List the users registered in the database and show the content of the table *SAPUSER*.

   a) Use the following select statement to show the content of the table *DBA_USERS* when connected as OPS$ user:

```
SQL> connect /
Connected.
SQL> select username from dba_users;

USERNAME
------------------------------
SYSTEM
SYS
ORACLE_OCM
WMSYS
OPS$TWDF1825\T99ADM
DBSNMP
DIP
OUTLN
OPS$TWDF1825\SAPSERVICET99
SAPT99
APPQOSSYS

11 rows selected.

SQL>
```

   b) Use the following select statement to show the content of the *SAPUSER* table when connected as OPS$ user:

```
SQL> connect /
Connected.
SQL> select * from sapuser;

USERID
---------------------------------------

PASSWD
---------------------------------------

SAPT99-CRYPT
V01/0010ZctvSB67Wv3u6Yw=

SQL>
```

3. List the database roles created by SAP.

   a) Use the following select statement to list the database roles created by SAP when connected as OPS$ user:

```
SQL> connect /
Connected.
SQL> select role from dba_roles where role like'SAP%';
```

```
ROLE
------------------------------
SAPCONN
SAPDBA

SQL>
```

4. (Optional) List the tables in the schema of the user `SAP<Schema-ID>`.

   a) Use the following select statement to list the tables in the schema of the user `SAP<Schema-ID>` when connected as OPS$-user:

```
SQL> connect /
Connected.
SQL> select table_name from dba_tables where owner='SAPT99';

TABLE_NAME
------------------------------
DBABARL
DBABD
DBABL
DBADFL
DBAERR
DBAEXTL
DBAFID
DBAGRP
DBAML
DBAOBJL
DBAOPTL

TABLE_NAME
------------------------------
DBAPHAL
DBAREOL
DBASPAL
DBATID
DBATL
DBATRIAL
DBCHECKORA
DBMSGORA
DBSTAIHORA
DBSTATC
DBSTATHORA

TABLE_NAME
------------------------------
DBSTATIORA
DBSTATTORA
SDBAD
SDBAH

26 rows selected.

SQL>
```

5. Use `connect /@<SID+1>` where <SID+1> is your SID plus one (for example, if your SID is T05, use `connect /@T06`).

   a) Call the `connect /@<SID+1>` command.

```
SQL> connect /@T06
ERROR:
```

```
ORA-01017: invalid username/password; logon denied

Warning: You are no longer connected to ORACLE.
SQL>
```

b) This is the OPS$ connect again. This time, it does not work because on database T06, no user `OPS$<DOMAIN>/T05ADM` exists, only `OPS$<DOMAIN>/T06ADM` exists (see the example given in step 5a).

6. Use `connect /@<SID+1> as sysdba` where <SID+1> is your SID plus one (for example, if your SID is T05, use `connect /@T06 as sysdba`).

   a) Call the `connect /@<SID+1> as sysdba` command.

   ```
   SQL> connect /@T06 as sysdba
   ERROR:
   ORA-01031: insufficient privileges
   ```

   b) The connect does not work this time because on database T06, your OS user `t05adm` does not belong to OS group `ORA_T06_DBA` (see the example given in step 5a).

LESSON SUMMARY

You should now be able to:

- Establish connection to the database

Unit 1
Lesson 3

Using Database Administration Tools

LESSON OVERVIEW

This lesson introduces the Oracle administration tool SQL*Plus. You will learn which SAP tools are available for database administration and how they are used.

Business Example

A database administrator (DBA) typically writes scripts to simplify the work of monitoring and managing databases and performing backups. SAP BR*Tools provides an intuitive, interactive interface for performing these tasks on SAP systems that use an Oracle database. As a DBA, you need to learn more about SAP BR*Tools so you can work more efficiently. For this reason, you require the following knowledge:

- An understanding of the main tools for administration of the Oracle database
- An understanding of the basic functions of SQL*Plus
- An understanding of how to use SAP tools for administration of the Oracle database – BR*Tools
- An understanding of the usage of the SAP Computing Center Management System (CCMS) for the administration of Oracle databases

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Use database administration tools

Database Administration Tools

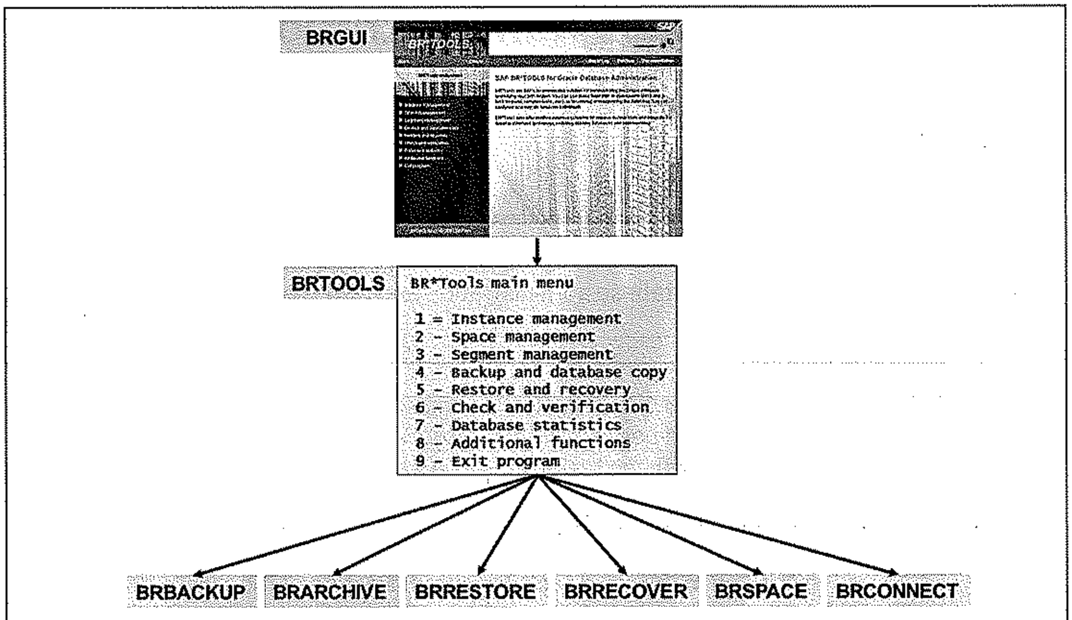


Figure 26: Hierarchy of SAP Tools for Oracle Administration

With the Oracle tool SQL*Plus, you can perform the following actions:

- Start and stop the database
- Log on to the database
- Perform database administration
- Enter SQL*Plus commands to configure an SQL*Plus environment
- Enter, edit, store, retrieve, and run Structured Query Language (SQL) commands and SQL scripts
- Redirect the output of query results to text files

You can also perform Oracle database administration in an SAP system at the operating system (OS) level with a set of SAP administration tools. These tools are called BR*Tools, and they are installed automatically on the database server in the directory `/usr/sap/<SID>/SYS/exe/run`.

> Caution:
> Although the SAP administration tool SAPDBA is still available for Oracle 9i, SAP strongly recommends not using the SAPDBA functions because these functions are no longer being developed. Oracle administration must now be performed exclusively with BR*Tools. BR*Tools can be used for all SAP Releases, provided you are using Oracle 9i or higher.

BRTOOLS is provided as a character-based user interface for the functional programs BRBACKUP, BRARCHIVE, BRRESTORE, BRRECOVER, BRSPACE, and BRCONNECT. Although BRTOOLS allows the DBA to manage the database through menus using a simple

Telnet connection or on a console, the graphical user interface (GUI) of the BRGUI program offers more convenient database administration.

BRGUI is a Java-based GUI for BRTOOLS. BRGUI itself does not offer any logic for database administration.

BRGUI performs the following functions:

- Calls BRTOOLS
- Displays BRTOOLS menus
- Forwards selections and mouse clicks to BRTOOLS

Using a remote shell, you can run BRGUI on the Administrator Workbench by calling BRTOOLS on the remote database server.

BR*Tools

The Oracle tool used at the OS level is SQL*Plus. It is an interactive and batch query tool providing access to the Oracle relational database management system (RDBMS) through a command line. It is installed with every Oracle server or client installation.

The following table describes the SAP tools for Oracle administration (BR*Tools):

Name	Description
BRBACKUP	Backup of data files, control files, and online redo log files
BRARCHIVE	Backup of offline redo log files
BRRESTORE	Restore of data files, control files, and offline redo log files
BRRECOVER	Interactive parent tool for database restore and recovery
BRCONNECT	Database administration, which includes database check, update of statics, changing user password, and so on
BRSPACE	Database administration, which includes instance management, space management, and reorganization
BRTOOLS	Interactive tool for calling other tools through menus
BRGUI	GUI for BRTOOLS

When the DBA performs a certain administrative task, the DBA can use the interactive, menu-driven program BRTOOLS, instead of calling a tool such as BRRESTORE, BRBACKUP, or BRCONNECT directly, which for some functions requires a detailed specification of the corresponding function and options. Through nested menus, BRTOOLS lets you choose the action and all the necessary options you need (including logon information).

Menu Structure of BR*Tools

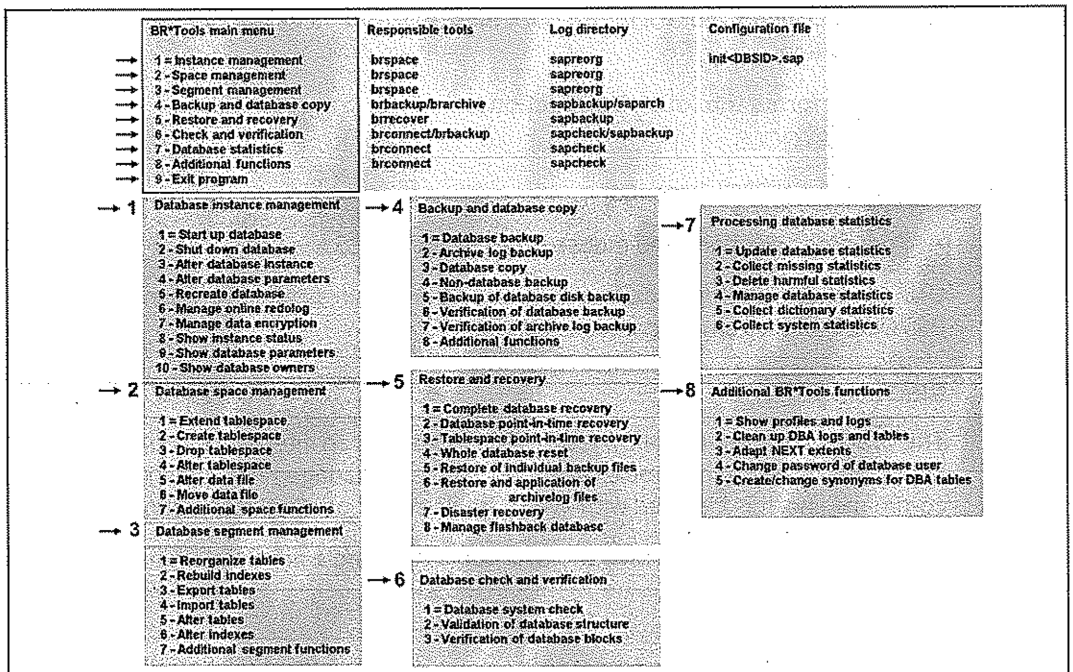


Figure 27: Menu Structure of BR*Tools

The figure provides an overview of the menu structure.

> Hint:
> Do not confuse the names BR*Tools and BRTOOLS. They have the following differences:
>
> - BR*Tools is the program package containing BRBACKUP, BRARCHIVE, BRRESTORE, BRRECOVER, BRSPACE, BRCONNECT, and BRTOOLS.
> - BRTOOLS is the interactive program that displays menus from which the other BR* programs are called.

BR*Tools – Program Types

The following table describes the program types in BR*Tools:

Type	Program Names	Description
Functional programs	BRBACKUP, BRARCHIVE, BRRESTORE, BRRECOVER, BRSPACE, and BRCONNECT	These programs perform actions on the database.
Help programs	BRTOOLS and BRCONNECT	These programs are called by other programs. All functional programs can be called interactively.

Type	Program Names	Description
Batch programs	BRBACKUP, BRARCHIEVE, BRRESTORE, and BRCONNECT	These tools do not have their own menus.
Interactive programs	BRRECOVER (besides BRTOOLS itself)	These programs offer their own menus.

Within BR*Tools, you can differentiate among the following programs:

- Functional programs

  These programs perform administrative actions on the database. The functional programs are BRBACKUP, BRARCHIVE, BRRESTORE, BRRECOVER, BRSPACE, and BRCONNECT.

- Help programs

  BRTOOLS and BRCONNECT are help programs.

  They help in the following ways:

  - From BRTOOLS, all functional programs can be called interactively. BRTOOLS offers menus to select the action, enter parameters and options, and then call the functional programs with the corresponding options and parameters.
  - BRTOOLS and BRCONNECT are also called by other tools. They perform certain actions during backup and restore (BRTOOLS as a help tool for BRBACKUP, BRARCHIVE, and BRRESTORE) and monitor the database during a backup (BRCONNECT, internally called by BRBACKUP).

  > Hint:
  > In previous releases, BRTOOLS and BRCONNECT were delivered only with this internal help functionality. If you get an error message instead of a menu when calling BRTOOLS, you have a BR*Tools version older than 6.20 and you need to get the newest BR*Tools.

- Batch programs

  The tools BRBACKUP, BRARCHIVE, BRRESTORE, and BRCONNECT do not have built in menus. Their functionality is selected exclusively by options. To use these tools interactively, call BRTOOLS and select actions, options, and parameters from the menu provided by BRTOOLS.

  > Hint:
  > For batch programs, specify all required options in BRTOOLS when directly called from the command line. If any required option is not specified, the system displays an error message.

- Interactive programs

  The functional programs BRRECOVER and BRSPACE are (as is BRTOOLS) interactive tools. Their functionality is mainly menu-driven.

You can also call BRRECOVER and BRSPACE directly from the command line as follows:

- When no option is specified, the menu of the default function (BRSPACE: *Show database information*) and the default recovery type (BRRECOVER: *Complete database recovery*) is shown.
- When a function (BRSPACE) or recovery type (BRRECOVER) is specified on the command line, the corresponding menu is shown and further input is performed from the menus.
- When further options are specified but they are not complete, you must select missing options and parameters from a menu.
- To force an interactive program to run in batch mode, call the interactive program with the option `-c force`. The interactive program then starts in batch mode. If mandatory options are not specified or any option or parameter is wrong, the tool ends with an error message.

> Caution:
> Be careful, because the default values for all optional parameters are selected when you perform the corresponding action.

BR*Tools – Mode Types

When calling any space management functionality of BRSPACE from BRTOOLS or BRGUI (*Space management* menu), select a function (for example, *Extend tablespace*), and then choose either the main menu mode or the quick mode.

- main menu mode

  If you do not specify anything in the menu called `BRSPACE options for <function>` (except specifying another BRSPACE profile and/or username and password, if required) and choose *Continue*, BRSPACE displays the main menu of the corresponding function. You can then select all required options and parameters from BRSPACE menus.

- quick mode

  In the `BRSPACE options` menu, if you specify anything for `<function>`, BRSPACE appends this to all command line options for all database objects previously identified.

  In this case, BRSPACE skips the main menu of the corresponding function if one object is specified in the BRTOOLS menu and also the object selection menu. You can then directly enter function options in the BRTOOLS input menu.

> Hint:
> Use the quick mode only if you already know the objects for which you want to perform the function. If you want to select objects from a list, or see a list of possible entries for parameters or options, use the main menu mode. Not all BRSPACE and BRRECOVER options can be set from BRTOOLS. All options can be set in menus provided by the interactive tools.

Caution:
Before executing an action, BRTOOLS displays the complete command line of the batch program it calls. You can use this command line as a reference to call this command later in batch. Use caution when changing the command line, as changes can result in errors.

BRSPACE displays the SQL command it executes before performing the operation.

For the following reasons, it is recommended that only Oracle experts use the BRSPACE command or a changed version of the BRSPACE command:

- In many cases, BRSPACE performs additional actions before and after executing the SQL command. Therefore, do not enter the command directly in SQL*Plus.
- BRSPACE performs several checks before creating the SQL command shown. If you change the command, you are responsible for the action performed by the changed command.

Common Command Line Options of BR*Tools

The following table represents the common command line options of BR*Tools:

<table>
<tr><th>Tool</th><th>Option</th><th>Meaning</th></tr>
<tr><td>All</td><td>-h|-help</td><td>List of all possible options and functions</td></tr>
<tr><td>BRSPACE, BRCONNECT</td><td>-h|-help <function></td><td>List of all possible options of <function></td></tr>
<tr><td>All</td><td>-c|-confirm</td><td>No confirmation required in unattended mode, attended mode is standard</td></tr>
<tr><td>BRSPACE, BRRECOVER</td><td>-c|-confirm force</td><td>Interactive tools to stop at menus even with -c|confirm; option force used to run them in batch</td></tr>
<tr><td>All</td><td>-u|-user [<name>[/<pw>]]</td><td>Username and password for database connection</td></tr>
</table>

If you run any BR*Tools tool with the option `-h|-help`, you get a list of all possible options. For BRSPACE and BRCONNECT, this includes a list of all program functions.

If you only need a listing of all options for a particular function of BRSPACE or BRCONNECT, call the tool with `-h|-help` and the corresponding function name, for example, `brspace -h tscreate`.

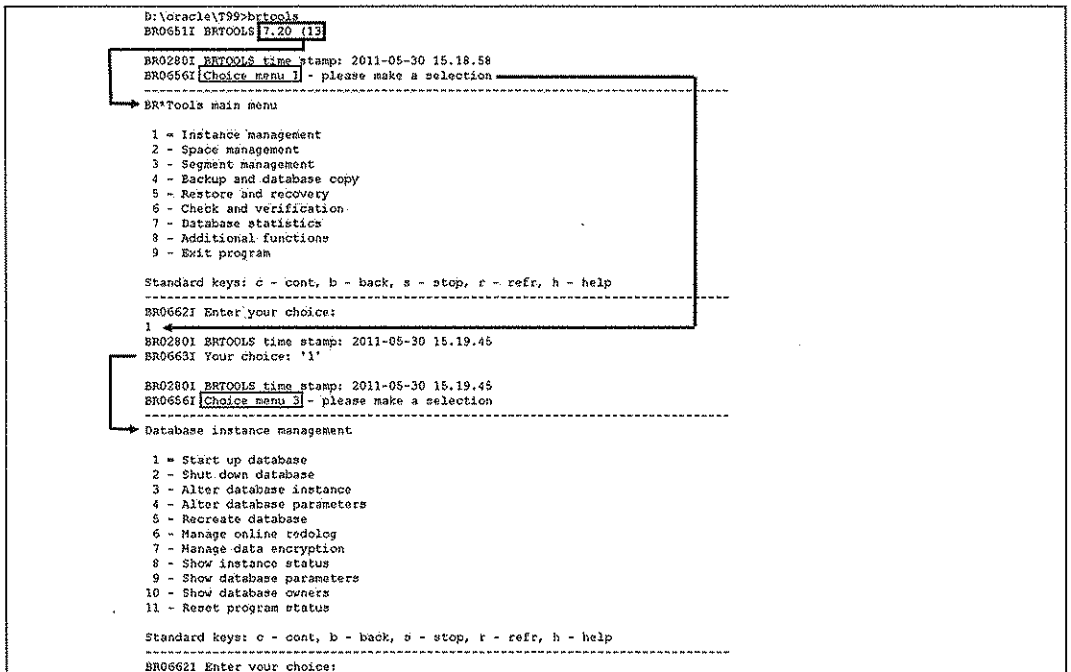
All tools run by their default settings in the attended mode, which means that the user must confirm every single step. To avoid this, start a tool with the option `-c|-confirm`. The interactive tools BRRECOVER and BRSPACE stop at menus, even when started with -c|-

`confirm`. To run the tools in a batch, use the option `-c | -confirm force`, which suppresses all confirmation messages and accepts the default input value in menus.

To avoid entering passwords interactively or appearing on the command line, BR*Tools uses the OPS$ user to connect to the database. To use a different user for the database connect, you can specify the option `-u | -user`.

> Hint:
> BRSPACE and BRRECOVER always make a CONNECT / AS SYSDBA because their actions require SYSDBA privilege. The username or password specified with the option `-u | -user` is only used for other BR*Tools called by BRSPACE or BRRECOVER, or for BRBACKUP, BRARCHIVE, and BRCONNECT called through BRTOOLS menus. Therefore, BRSPACE and BRRECOVER are called from an OS user belonging to the DBA group (<sapsid>adm on Windows and ora<dbsid> on UNIX) on the database server.

Working with BRTOOLS – Menu Structure



```
D:\oracle\T99>brtools
BR0651I BRTOOLS 7.20 (13

BR0280I BRTOOLS time stamp: 2011-05-30 15.18.58
BR0656I Choice menu 1 - please make a selection
-------------------------------------------------------------------------------
BR*Tools main menu

 1 = Instance management
 2 - Space management
 3 - Segment management
 4 - Backup and database copy
 5 - Restore and recovery
 6 - Check and verification
 7 - Database statistics
 8 - Additional functions
 9 - Exit program

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-------------------------------------------------------------------------------
BR0662I Enter your choice:
1
BR0280I BRTOOLS time stamp: 2011-05-30 15.19.45
BR0663I Your choice: '1'

BR0280I BRTOOLS time stamp: 2011-05-30 15.19.45
BR0656I Choice menu 3 - please make a selection
-------------------------------------------------------------------------------
Database instance management

 1 = Start up database
 2 - Shut down database
 3 - Alter database instance
 4 - Alter database parameters
 5 - Recreate database
 6 - Manage online redolog
 7 - Manage data encryption
 8 - Show instance status
 9 - Show database parameters
10 - Show database owners
11 - Reset program status

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-------------------------------------------------------------------------------
BR0662I Enter your choice:
```

Figure 28: Working with BRTOOLS – Menu Structure

After you start BRTOOLS, you must choose a type of administrative activity from the main menu.

The main menu and all menus on the next level are called choice menus. You can make an independent choice from the menu in any sequence. You can repeat the choice as often as necessary.

Other types of menus are as follows:

- Control menu

The steps presented by the menu must be processed in the given sequence, for example, during restore and recovery.

- Input menu

  The system suggests values of required parameters or options and these values can be modified.

- List menu

  The system lists several items from which you can select one or more entries, for example, one database backup from several available backups during restore.

Standard keys, used in all types of menus, are represented in the following table:

Standard Keys	Meaning
c - cont	Continues to the next menu or program step
b - back	Goes back to the previous menu or program step
s - stop	Cancels the active program
r - refresh	Refreshes the screen and makes some plausibility checks
h - help	Calls context-specific help

> Hint:
> When entering BRSPACE from BRTOOLS using quick mode, going back (using the b – back key) deactivates quick mode and you can reach the main menu of the corresponding function.

Altering a Database User's Password

For security reasons, you must regularly change the passwords of standard database users, as well as the passwords of SAP user SAPR3 or SAP<SCHEMA-ID>.

Do not change the password of database user SAP<SCHEMA-ID> (or SAPR3) with Oracle methods because the password must be maintained in the *SAPUSER* table. Instead, use the chpass function of BRCONNECT, because it changes the password both in the Oracle system table and in *SAPUSER*.

To change the password of a database user, call BRTOOLS or BRGUI and choose *Additional functions → Change password of SAP user*.

The following example shows how to change the password of an SAP owner, using BRCONNECT:

```
D:\oracle\T99>brtools
BR0651I BRTOOLS 7.20 (13)

BR0280I BRTOOLS time stamp: 2011-05-30 15.46.08
BR0656I Choice menu 1 - please make a selection
-----------------------------------------------------------------------
BR*Tools main menu

 1 = Instance management
 2 - Space management
```



```
 3 - Segment management
 4 - Backup and database copy
 5 - Restore and recovery
 6 - Check and verification
 7 - Database statistics
 8 - Additional functions
 9 - Exit program

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-----------------------------------------------------------------------
BR0662I Enter your choice:
8
BR0280I BRTOOLS time stamp: 2011-05-30 15.46.12
BR0663I Your choice: '8'

BR0280I BRTOOLS time stamp: 2011-05-30 15.46.12
BR0656I Choice menu 2 - please make a selection
-----------------------------------------------------------------------
Additional BR*Tools functions

 1 = Show profiles and logs
 2 - Clean up DBA logs and tables
 3 - Adapt NEXT extents
 4 - Change password of database user
 5 - Create/change synonyms for DBA tables
 6 - Reset program status

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-----------------------------------------------------------------------
BR0662I Enter your choice:
4
BR0280I BRTOOLS time stamp: 2011-05-30 15.46.16
BR0663I Your choice: '4'

BR0280I BRTOOLS time stamp: 2011-05-30 15.46.16
BR0657I Input menu 59 - please enter/check input values
-----------------------------------------------------------------------
BRCONNECT options for changing password of database user

 1 - BRCONNECT profile (profile) .............. [initT99.sap]
 2 - Database user/password (user) ............ [/]
 3 ~ Database owner to change password (owner) . []
 4 ~ Log file name (log) ...................... []
 5 - Message language (language) .............. [E]
 6 - BRCONNECT command line (command) ......... [-p initT99.sap -l E -
f chpass]


Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-----------------------------------------------------------------------
BR0662I Enter your choice:
3
BR0280I BRTOOLS time stamp: 2011-05-30 15.46.27
BR0663I Your choice: '3'
BR0681I Enter string value for "owner" []:
SAPT99
BR0280I BRTOOLS time stamp: 2011-05-30 15.46.31
BR0683I New value for "owner": 'SAPT99'

BR0280I BRTOOLS time stamp: 2011-05-30 15.46.31
BR0657I Input menu 59 - please enter/check input values
-----------------------------------------------------------------------
BRCONNECT options for changing password of database user

 1 - BRCONNECT profile (profile) .............. [initT99.sap]
```



```
 2 - Database user/password (user) ............. [/]
 3 ~ Database owner to change password (owner) . [SAPT99]
 4 ~ Log file name (log) ....................... []
 5 - Message language (language) ............... [E]
 6 - BRCONNECT command line (command) .......... [-p initT99.sap -l E -
f chpass -o SAPT99]

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-------------------------------------------------------------------------
BR0662I Enter your choice:
c
BR0280I BRTOOLS time stamp: 2011-05-30 15.46.33
BR0663I Your choice: 'c'
BR0259I Program execution will be continued...

BR0291I BRCONNECT will be started with options '-p initT99.sap -l E -f
chpass -o SAPT99'

BR0280I BRTOOLS time stamp: 2011-05-30 15.46.33
BR0670I Enter 'c[ont]' to continue, 'b[ack]' to go back, 's[top]' to
abort:
c
BR0280I BRTOOLS time stamp: 2011-05-30 15.46.34
BR0257I Your reply: 'c'
BR0259I Program execution will be continued...

#########################################################################

BR0801I BRCONNECT 7.20 (13)
BR0280I BRCONNECT time stamp: 2011-05-30 15.46.37
BR1048I Enter new password for database user 'SAPT99' (max. 30
characters):

BR0280I BRCONNECT time stamp: 2011-05-30 15.46.42
BR1048I Reenter new password for database user 'SAPT99' (max. 30
characters):

BR0280I BRCONNECT time stamp: 2011-05-30 15.46.44
BR0829I Password changed successfully in database for user SAPT99
BR0830I Password changed successfully in table OPS
$TWDF1825\T99ADM.SAPUSER for user SAPT99

BR0280I BRCONNECT time stamp: 2011-05-30 15.46.44
BR0802I BRCONNECT completed successfully

#########################################################################

BR0292I Execution of BRCONNECT finished with return code 0

BR0280I BRTOOLS time stamp: 2011-05-30 15.46.44
```

The option `-u|-user` defines the username and password used by the SAP tool to log on to the database. This user must be defined in the database and have at least SYSOPER authorization. The option `-o|-owner` specifies the user for whom the password must be changed. You can use this function to change the password of any database user.

Menu Symbols in BR*Tools

Special symbols are used in BR*Tools menus at the beginning of every line. The symbols help DBAs orient themselves better in the menus and use the tools more effectively.

The following table the menu symbols in BR*Tools:

Character Interface	Used in Menu Types	Meaning
+	Control Choice	The action is completed
Information message	Display	• An error message is logged to the detail log in the form: BRxxxxI. • This message is used for all rows in the display menu.
Warning message	Not used	An error message is written to the detail log in the form: BRxxxxW.
Error message	Not used	An error message is written to the detail log in the form: BRxxxxE.
–	Control Choice List	This can be chosen or executed now.
*	Control Input	• This cannot be chosen or executed now. • Display entry, no input possible.
–	Input	You can change this parameter.
~	Input	You can change this optional parameter or reset its value to null (use a single space for this in the character interface).
stop	All menus	This cancels the active program.
help	All menus	This calls context-specific help.
back	All menus	Go back to the previous menu or program step.
continue yes	All menus	Continue to the next menu or program step.
no	Not used	Skips the following actions and go to the next program step; this action is recorded as BR0676I in the detail log.
=	Control Choice List	This is the initial default choice; it is automatically selected if you choose *Continue*.

Character Interface	Used in Menu Types	Meaning
?	Input	You must enter a value for this parameter.
#	Control Input	You cannot execute this action or change this parameter.

SQL*Plus

Start SQL*Plus by calling the program SQLPLUS. The system prompts you to enter a username and password to connect to your default database.

The following example shows how to call the program and connect to the database:

```
D:\oracle\T99>sqlplus

SQL*Plus: Release 11.2.0.1.0 Production on Mo Mai 30 15:49:30 2011

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

Enter user-name: sapt99
Enter password:<password not shown on screen>

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit
Production
With the Partitioning, OLAP, Data Mining and Real Application Testing
options

SQL>
```

If you do not want to enter the username and password interactively, you can perform either of the following steps:

- Enter the username/password as the first parameter when calling SQLPLUS. For example, `sqlplus system/manager` where `system` is the username and `manager` is the password.
- Call SQLPLUS with the option `/nolog`. In this case, the first SQL*Plus command must be a `connect username/password` command. The following example shows how the interactive logon is suppressed and the `CONNECT` command is necessary to enter SQL*Plus commands.

The following example shows how the interactive logon is suppressed and the CONNECT command is necessary to enter SQL*Plus commands:

```
D:\oracle\T99>sqlplus /nolog

SQL*Plus: Release 11.2.0.1.0 Production on Mo Mai 30 15:51:08 2011

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

SQL> archive log list;
ORA-03114: not connected to ORACLE
SQL> connect / as sysdba
Connected.
SQL> archive log list;
Database log mode              Archive Mode
Automatic archival             Enabled
```

```
Archive destination            D:\oracle\T99\oraarch\T99arch
Oldest online log sequence     104
Next log sequence to archive   107
Current log sequence           107
SQL>
```

The general syntax to specify username and password on the connection is `username/password[@DBSID]`. By default, SQL*Plus connects to the database defined in the environment variable `%ORACLE_HOME%` (Windows) or `$ORACLE_HOME` (UNIX). To connect to another database, enter the <SID> directly after the password without a blank, delimited by the @ (at) sign. To connect to the database with a user having the SYSDBA privilege, but without specifying a password, do a `CONNECT / AS SYSDBA`.

Customizing SQL*Plus

SQL*Plus maintains system variables that allow you to set up a particular environment for a SQL*Plus session.

You can change the system variables with the SET command as follows:

```
SET <VARIABLE> <VALUE>
```

You can list system variables with SHOW command as follows:

```
SHOW <VARIABLE>
```

```
SHOW ALL
```

The following table represents the most commonly used SQL*Plus system variables:

Variable	Value	Description
`PAGESIZE`	integer	Displays number of lines per page (after this number of lines, a new header line is displayed).
`PAUSE`	`ON|OFF`	If `ON`, the `PAUSE` variable stops the output after `PAGESIZE` lines and continues with *Return*.
`HEADING`	`ON|OFF`	Enables and disables the output of column headings.
`LINESIZE`	integer	Displays width for data. After displaying the number of characteristics, the output line is wrapped or truncated, depending on the value of `WRAP`.
`WRAP`	`ON|OFF`	If set to `ON`, the variable wraps the lines after `LINESIZE` characters. If the variable is set to `OFF`, the line is truncated.

Variable	Value	Description
SPOOL	<filename>|OFF	The SPOOL variable copies the output of all subsequent commands and their output to <filename>, until set OFF.
AUTO[COMMIT]	ON|OFF	If set to ON, pending changes are automatically committed to the database after each successful INSERT, UPDATE, or DELETE statement. If set to OFF, such changes must be committed manually.
TERMOUT	ON|OFF	If set to OFF, the variable suppresses terminal output (only works in scripts).

For a complete list, refer to the Oracle documentation.

> Hint:
> If AUTOCOMMIT is OFF (the default setting), data modifications performed from SQL*Plus are committed in the database only when you execute the COMMIT command in the same session or disconnect from the database.

Although SQL*Plus variables can be changed dynamically in a SQL*Plus session, a common method for changing SQL*Plus variables is to create a script file containing a SET command for each variable you want to set, and then execute the file every time you start SQL*Plus.

To create a script file, perform the following steps:

1. Create an ASCII file with the extension sql, for example, sqlplusenv.sql, in a directory from which you start SQL*Plus, for example, in your home directory.
2. Enter a **SET** command for each relevant variable in this script file.
3. Save the contents.
4. When starting SQL*Plus, enter the script file name as the last option of the sqlplus command, separated by a space, and with the @ sign in front of the file name (you can omit the extension sql): `sqlplus {username[/password]|/}[@<DBSID>] @<script>`.

SAP Database Monitors

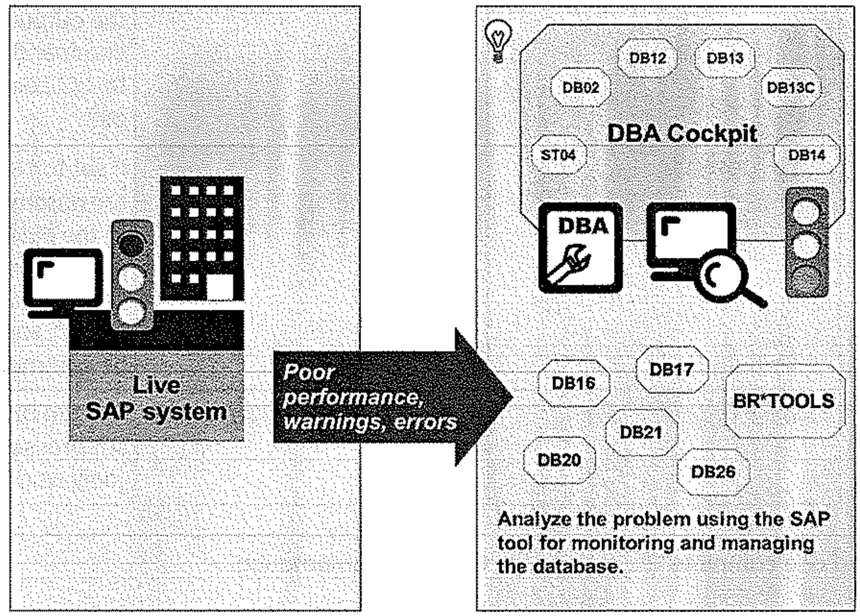


Figure 29: Database Administration and Monitoring in SAP System

To minimize system downtime and improve performance, you must schedule regular administrative jobs, such as backups and database checks, and monitor your database daily. The DBA Cockpit, first delivered with SAP Basis Release 7.00 Support Package 12, forms the central access point for the monitoring and administration of SAP systems with an Oracle database.

The DBA Cockpit replaces various transactions that were previously used for monitoring and administration (see SAP Note 1028624).

These include the following transactions that lead to individual functions in the DBA Cockpit:

- The DBA Planning Calendar and the Central DBA Planning Calendar (transactions `DB13` and `DB13C`) are available for scheduling backups and other administrative jobs in your database system. You can use these transactions to schedule backups and administrative activities locally or centrally for several SAP systems and databases.
- You can use the Backup Log Overview (transaction `DB12`) to display the results of your backups and the status of the archive directory. If all backups are available for a restore and recovery, the Backup Log Overview transaction also contains a function for checking the restore reports.
- The DBA Operations Monitor (transaction `DB14`) checks the status and logs of all database operations, including backup monitoring, updates of the optimizer statistics, and database checks.
- The Database Performance Monitor (transaction `ST04`) displays the most important indicators for Oracle database performance, such as buffer cache quality, statistics of user calls, or number of block reads per SQL statement. The Database Performance Monitor also displays the important indicators of the configuration of the database management

system and the database, either directly through showing current parameter values or with help of V$ views.

- The Tables and Indexes Monitor (transaction DB02) monitors the storage behavior of the database (for example, space statistics showing the history of the database or size and free space in each tablespace) and the status of the database objects (for example, size of each table in kilobytes and blocks or indexes that are defined in the ABAP Dictionary but missing in the database).

These transactions are still available, but the transaction codes are renamed to <PREVIOUS TRANSACTION CODE>OLD. This means that the transaction codes are now ST04OLD, DB02OLD, DB12OLD, DB13OLD, DB14OLD, and DB13COLD. The DBA Cockpit is called using the transaction DBACOCKPIT.

Database Monitoring

In addition to the DBA Cockpit, the following transactions are used to monitor the database:

- DB16

  Use this transaction to view the overview of database checks.

- DB17

  Use this transaction to view and maintain check conditions used by a database system check.

- DB20

  Use this transaction to edit table statistics.

- DB21

  Use this transaction to configure statistics.

- DB26

  Use this transaction to view the database parameter overview with history.

- RZ20

  Use this transaction to start the database alert monitor, which monitors all preset alerts for different areas of the database.

DBA Cockpit

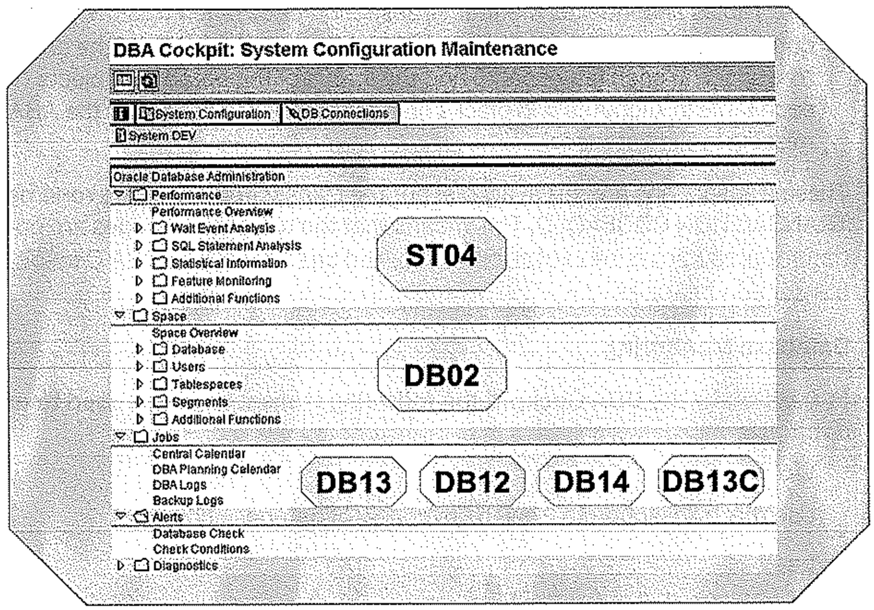


Figure 30: DBA Cockpit

The DBA Cockpit offers a navigation area, which is visible in all functions of the DBA Cockpit.

This navigation area contains a menu tree with the following access points:

- *Performance* (corresponds to the old transaction `ST04`)
- *Space* (corresponds to the old transaction `DB02`)
- *Jobs* (corresponds to the old transactions `DB13`, `DB12`, `DB14`, and `DB13C`)
- *Diagnostics*

The individual functions of the DBA Cockpit occur under these access points, which can each be called by double-clicking.

The following prerequisites are required for monitoring and administration of the local system (that is, the system running on the DBA Cockpit):

- Specific database objects are required for some performance monitors in the DBA Cockpit. These objects are created with a SQL script (see SAP Note 706927 for the script and other information).
- IBR*Tools 700, patch level 24 or higher must be installed for the new planning calendar.
- The Oracle Active Workload Repository must be available (see SAP Note 1028068).

In addition, other corrections must be imported for the DBA Cockpit with Basis 7.00 Support Packages 12 and 13, which are first fixed in Support Package 13 or 14 (see SAP Note 1028624).

Administration and Monitoring of External Databases

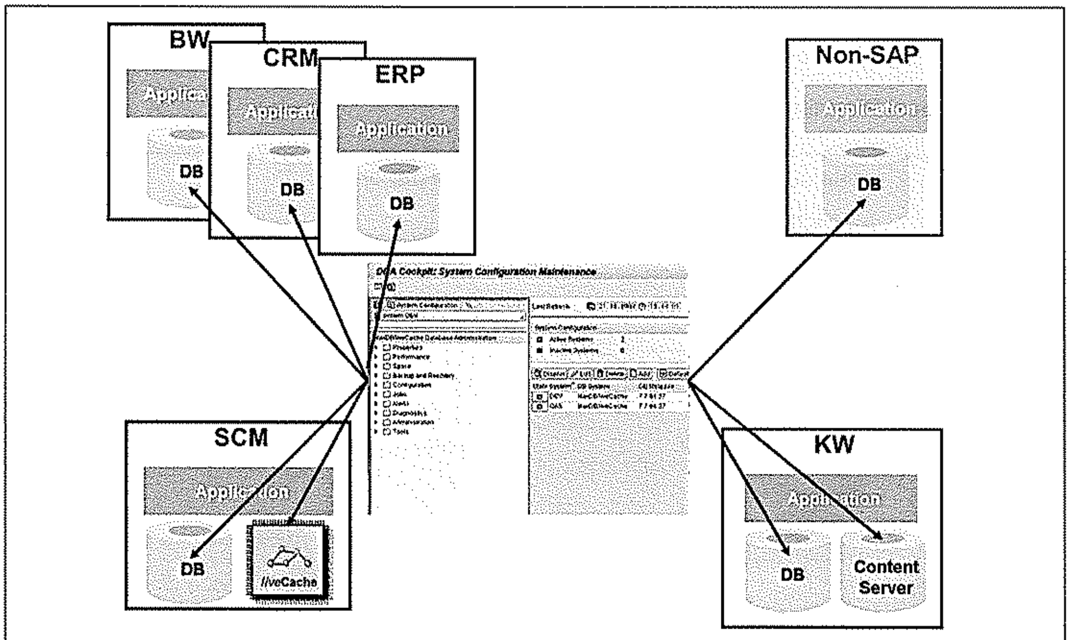


Figure 31: DBA Cockpit – Remote Monitoring

An important feature of the DBA Cockpit is the option of monitoring and administering external databases, including both ABAP and non-ABAP systems.

In addition to Oracle databases, all other SAP-supported database systems (for example, IBM DB2 UDB, IBM DB2 z/OS, IBM DB2 AS/400, MS SQL Server, and MaxDB) can be connected as external databases. To connect to a non-Oracle database, you must install the corresponding database client software and the SAP shared library.

> Hint:
> If the system on which the DBA Cockpit is running is not an Oracle database system and you need to connect external Oracle databases, you must first install the Oracle database client software and the corresponding SAP database shared library.
>
> External Oracle databases are supported as of Oracle 9.2.
>
> To connect external Unicode databases, the system on which the DBA Cockpit is running must be a Unicode system.

A functional secondary database connection is required to connect external databases. In other words, there must be an entry in the *DBCON* table. When adding an external database in the system configuration of the DBA Cockpit, you can create your own DBCON entry.

To connect external Oracle databases, note the other following prerequisites:

- The script must be executed on the external Oracle database (see SAP Note 706927).
- For the planning calendar (transaction DB13) and the corresponding transactions DB12 and DB14, you must configure either an Remote Function Call (RFC) ABAP connection

(only possible for external ABAP systems) or a connection through the SAP gateway or remote shell. For more information on this procedure see SAP Note 1025707.

- A minimum of BR*Tools 700 patch level 24 (Oracle client 10) or BR*Tools 640 patch level 52 (Oracle client 9) must be installed on the external system.

Local Planning Calendar

DBA Cockpit: Jobs → DBA Planning Calendar (DB13)

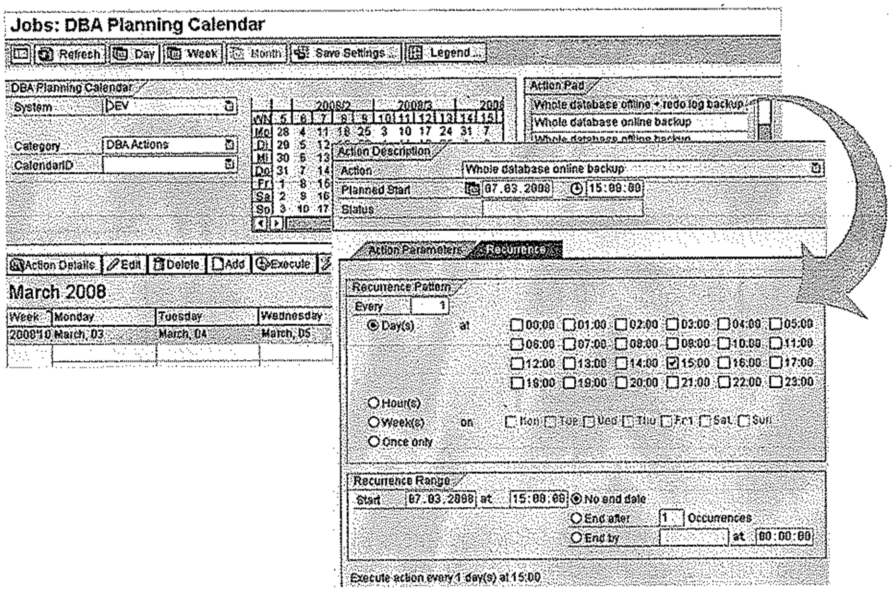


Figure 32: DBA Planning Calendar

To schedule background jobs, choose *Jobs → DBA Planning Calendar* in the DBA Cockpit menu tree (transaction `DB13`).

From the DBA Planning Calendar, you can schedule the following periodic administrative jobs for a database:

- Database check
- Database backups
- Updating the optimizer statistics
- Adapting next extents (pertinent only for dictionary managed tablespaces)

To comply with legal accessibility requirements in interface programs, a new planning calendar has replaced the old planning calendar. However, the old planning calendar is still available through transaction `DB13OLD`. You can also view completed actions in both the old and new planning calendars. Scheduled actions in the old planning calendar also appear as scheduled actions in the new planning calendar. The reverse also applies; that is, actions that are scheduled in the new planning calendar can also be viewed in the old one. Periodically scheduled actions in the new planning calendar with any repeat period other than weekly (for example, hourly or daily), do not appear in the old planning calendar. This can create issues if you are using the old and new planning calendars at the same time. Therefore, it is recommended that you do not use both versions of the planning calendar at the same time.

Even though the old planning calendar can still be used, it now has the status *deprecated*. If you do not want to use the new planning calendar immediately, it is recommended that you switch to it in the medium term. For more information on the new planning calendar, particularly the configuration and administration of external databases, see SAP Note 1025707.

The DBA Planning Calendar is a simple interface to schedule background jobs that perform administrative tasks. These jobs are named *DBA:**. These background jobs use the information in the table *SDBAC* to determine which OS level command must be executed on which server.

The DBA Planning Calendar provides templates for all activities that are recommended to be regularly performed on the database.

Central Planning Calendar

To use the Central DBA Planning Calendar, choose *Jobs → Central Planning Calendar* in the DBA Cockpit menu tree (transaction `DB13C`).

The new central planning calendar is fundamentally different from the old one (which is still available as transaction `DB13COLD`), in terms of the configuration of the systems to be monitored. The new central planning calendar uses the system configuration of the DBA Cockpit to add or remove systems. To migrate the systems from the old central planning calendar to the new one, choose *Administration → Migration DB13C Configuration* menu option.

DBA Operations Monitor

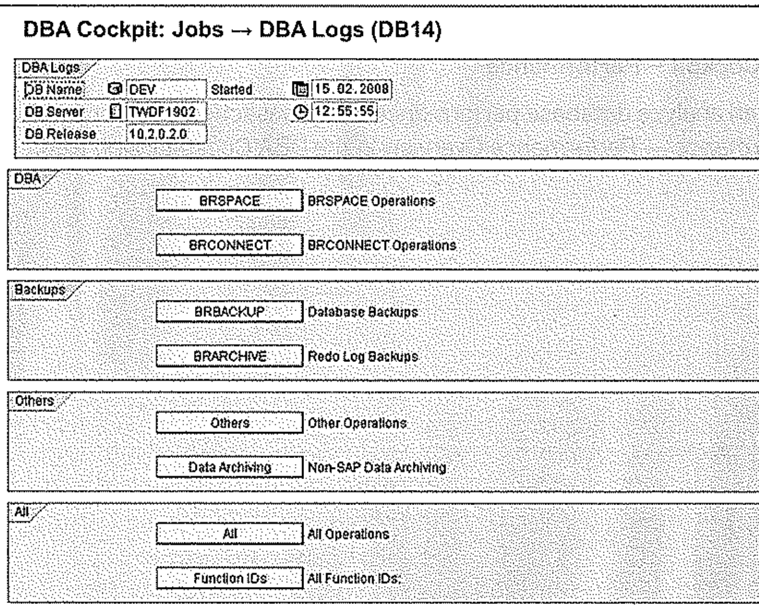


Figure 33: DBA Operations Monitor

In the DBA Operations Monitor, choose *Jobs → DBA Logs* in the DBA Cockpit menu tree (transaction `DB14`) to monitor online database operations and operation runtime.

The DBA Operations Monitor provides an overview of the activities of any of the BR*Tools. To display specific database operations (for example, backup operations), choose the corresponding button. For your daily check, choose *All* to see a list of all activities and their results. The colors indicate if an activity had warnings (yellow = return code 0001) or errors (red = return code larger than 0001).

> Hint:
> To restrict the activity list to a specific period, or to display only the activities with warnings or errors, select the list of corresponding operations and then choose the *Selection criteria* button.

To see the action log of operations, double-click the corresponding line. You can view the detail log by choosing the *Detail Log* button.

To see an overview of the backup logs, choose transaction `DB12` or *Jobs → Backup Logs* in the DBA Cockpit menu tree. The overview shows the results of the data backups and the status of the archive directory. If all backups are available for a restore and recovery, the backup log also contains a function for checking the restore reports.

DBA Cockpit – Space

Choose *Space → Space Overview* in the DBA Cockpit menu tree (transaction `DB02`) to see the functions for monitoring disk space in the database.

This overview contains information about how much disk space the database is using. To ensure that the required data for this overview is available, you must first schedule a background job. Available disk space information about individual tablespaces or table is provided in additional submonitors. Information about the growth of the individual database objects is also provided.

DBA Cockpit – Performance

You can analyze database performance with transaction `ST04` or by choosing *Performance → Performance Overview* in the DBA Cockpit menu tree.

Unit 1
Exercise 3

Use Database Administration Tools

Business Example

You want to learn more about how BR*Tools is used to display information about the database and perform some actions on the database.

Use BR*Tools.

> Note:
> It is recommended that you perform the exercise using BRGUI, but you can also use BRTOOLS directly. For better readability, the provided solutions come from BRTOOLS.

1. Use BR*Tools to display all users of your database.
2. Use BR*Tools to change the password of the user SAP<SCHEMA-ID> to `secret`. Then change the password again to `sap`.

Unit 1
Solution 3

Use Database Administration Tools

Business Example

You want to learn more about how BR*Tools is used to display information about the database and perform some actions on the database.

Use BR*Tools.

> Note:
> It is recommended that you perform the exercise using BRGUI, but you can also use BRTOOLS directly. For better readability, the provided solutions come from BRTOOLS.

1. Use BR*Tools to display all users of your database.
   a) Call BRGUI or BRTOOLS.
   b) Choose *Instance management → Show database owners*.
   c) Choose *Continue* two times.
   d) The *list of database owners* input menu appears with the following information:

```
List of database owners

Pos.  Owner        Id.  Created     Deft-Tsp.       Temp-Tsp
Status

  1 - APPQOSSYS    31  2011-05-27  SYSAUX          PSAPTEMP
EXPIRED &amp; LOCKED
  2 - DBSNMP       30  2011-05-27  SYSAUX          PSAPTEMP
EXPIRED &amp; LOCKED
  3 - DIP          14  2011-05-27  SYSTEM          PSAPTEMP
EXPIRED &amp; LOCKED
  4 - OPS$TWDF1825\SAPSERVICET99  38  2011-05-27  SYSTEM
PSAPTEMP    OPEN
  5 - OPS$TWDF1825\T99ADM  34  2011-05-27  SYSTEM
PSAPTEMP       OPEN
  6 - ORACLE_OCM   21  2011-05-27  SYSTEM          PSAPTEMP
EXPIRED &amp; LOCKED
  7 - OUTLN         9  2011-05-27  SYSTEM          PSAPTEMP
OPEN
  8 - SAPT99       35  2011-05-27  PSAPT99USR      PSAPTEMP
OPEN
  9 - SYS           0  2011-05-27  SYSTEM          PSAPTEMP
OPEN
 10 - SYSTEM        5  2011-05-27  SYSTEM          PSAPTEMP
OPEN
 11 - WMSYS        32  2011-05-27  SYSAUX          PSAPTEMP
EXPIRED &amp; LOCKED
```

2. Use BR*Tools to change the password of the user SAP<SCHEMA-ID> to **`secret`**. Then change the password again to **`sap`**.

   a) Call BRGUI or BRTOOLS and choose *Additional functions → Change password of database user*.

   In the input menu *BRCONNECT options for changing password of database user*, enter the name of user SAP<SCHEMA-ID> in the parameter *Database owner to change password*.

   > Note:
   > If no username is entered in the parameter *Database owner to change password*, the system changes the password for the user SAP<SCHEMAID> by default.

   b) Choose *Continue*.

   c) Enter the following data in the table that appears on the screen:

Field	Value
New database password (password)	**`secret`**
Confirmation of new password (password2)	**`secret`**

   d) Choose *Continue*. The system displays the information *BRCONNECT completed successfully*.

   e) Go back to main menu.

   f) Choose *Additional functions → Change password of database user*.

   g) Choose *Continue*.

   h) Enter the new password as given in the following table:

Field	Value
New database password (password)	**`sap`**
Confirmation of new password (password2)	**`sap`**

   i) Choose *Continue*.

LESSON SUMMARY

You should now be able to:

- Use database administration tools

Unit 1
Lesson 4

Administrating Oracle Instances

LESSON OVERVIEW

This lesson explains how to start and stop an Oracle instance and change Oracle parameters. The lesson also explains where diagnosis files are stored.

Business Example

Your SAP EarlyWatch report recommends that you change some Oracle parameters. Your next maintenance window for the SAP system, which allows a restart of the database, is in two weeks. Therefore, you want to change the parameters dynamically. For this reason, you require the following knowledge:

- An understanding of how to change parameters for initialization
- An understanding of how to start and stop the Oracle instance
- An understanding of how to identify and monitor diagnosis files

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Change Oracle parameters

Oracle Instances

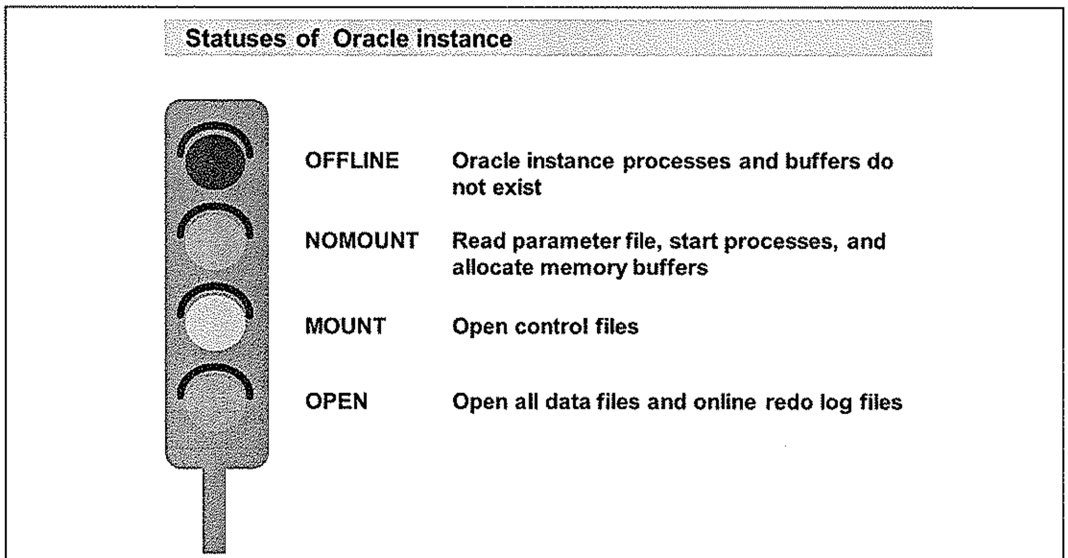


Figure 34: Starting an Oracle Instance

When an Oracle database is started, it goes through the following phases:

1. NOMOUNT

   In the NOMOUNT phase, the SAP database system opens the parameter file and evaluates and the database instance starts. Operating system (OS) resources are allocated using the configuration information stored in the parameter file.

2. MOUNT

   In the MOUNT phase, the control files of the database are opened, using the value of parameter `CONTROL_FILES` from the parameter file and evaluated. The system reads the information about the physical structure of the database. Although data files and log files are not yet opened, parts of the data dictionary are loaded; therefore, V$ views are available. If any of the files specified by the `CONTROL_FILES` initialization parameter does not exist or cannot be opened when you attempt to mount a database, Oracle returns an error message and does not mount the database.

3. OPEN

   In the OPEN phase, the database system opens all the remaining files of the database system. An instance recovery is performed during opening the database, if necessary. Valid database users can connect to the database only when it is open. If any of the data files or redo log files is not available or cannot be opened when attempting to open a database, Oracle returns an error message and does not open the database.

To start the database, start BRTOOLS or BRGUI and choose *Instance management → Start up database*. To start the database in batch, use `brspace -c force -f dbstart -s <state>`.

> Hint:
> If the database is in any state other than shutdown when starting the database with BRSPACE, the startup fails unless you specify the `-f | -force` option for the `dbstart` function. In this case, the database shuts down before starting in the requested state.

The NOMOUNT and MOUNT instance states are needed for the following special administrative tasks:

- The NOMOUNT state is necessary for creating a database and re-creating lost control files.
- The MOUNT state is necessary for database recovery, changing the ARCHIVELOG mode, renaming (moving) data files, and adding, dropping, or renaming online redo log files.

> Hint:
> BRCONNECT also offers menus and options to start up and shut down the database. The use of this functionality of BRCONNECT is not recommended. Instead, use BRSPACE because it writes log files of its actions.

Shutdown Modes

To stop the database, start BRTOOLS or BRGUI and choose *Instance management → Shut down database*. To stop the database in batch, use `brspace -c force -f dbshut -m <mode>`.

The following shutdown modes are possible:

- NORMAL

  With this mode, no new connections are allowed. Oracle waits for all currently connected users to disconnect from the database. Only when the last user disconnects (in the SAP system, all work processes are stopped), Oracle shuts down the database, that means, it closes all files, dismounts the database, and shuts down the instance.

- TRANSACTIONAL

  With this mode, no new connections or transactions are allowed. Oracle waits for all open transactions to finish and then disconnects all database connections and shuts down the database.

- IMMEDIATE

  With this mode, no new connections or transactions are allowed. The process monitor (PMON) process ends all user sessions and performs a rollback of any open transactions. Then, the database shuts down.

- ABORT

  With this mode, no new connections or transactions are allowed. All client SQL statements currently being processed are terminated, without rolling back open transactions. User connections are disconnected and Oracle processes are stopped.

When any of the first three methods is used, the database shuts down in a consistent state and no instance recovery is required at the next restart. After SHUTDOWN ABORT, the database data can be inconsistent because of aborted transactions and the database requires instance recovery at the next startup, which is performed automatically. Therefore, only use this method in exceptional cases, for example, when an Oracle background process terminates abnormally.

> Note:
> NORMAL is the default mode for the Oracle command SHUTDOWN, whereas the IMMEDIATE is the default mode for BRSPACE.

While Oracle commands SHUTDOWN IMMEDIATE and SHUTDOWN ABORT stop the Oracle instance even if work processes in the SAP system have connections to the database, BRSPACE checks for `SAP<SCHEMA-ID>` or `SAPR3` connections and does not continue if any such connections exist. To force the shutdown of the database while SAP is running, use the `-f|-force` option to force BRSPACE to shut down the database.

Initialization Parameters

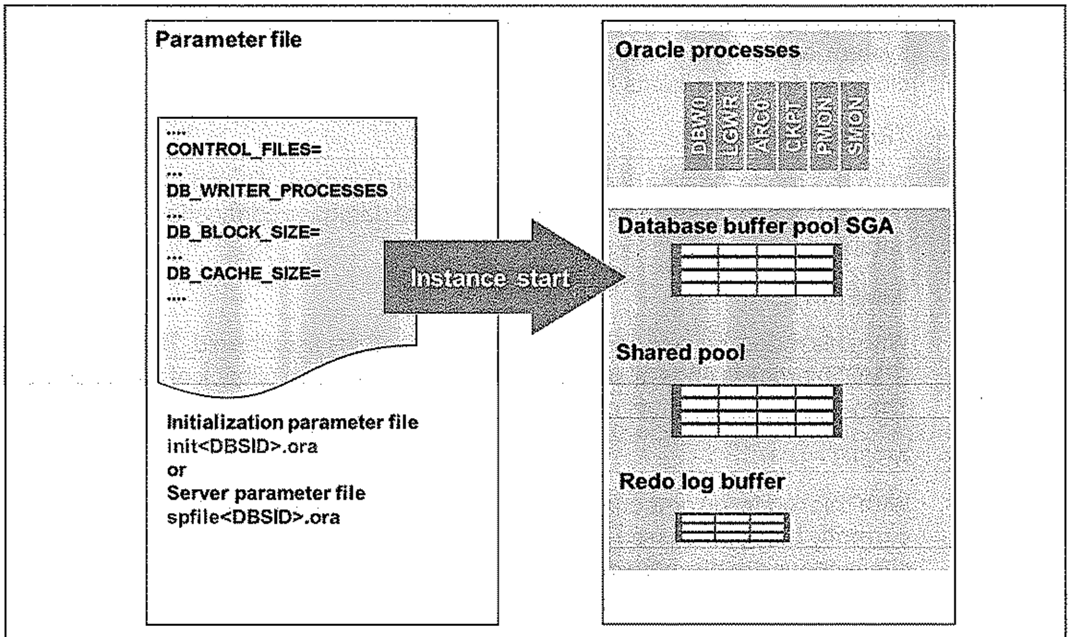


Figure 35: Oracle Instance Profile

The instance for an Oracle database starts using an initialization parameter file. Parameters contained in this file configure the system global area (SGA) and Oracle background processes.

Examples of instance parameters are as follows:

- `CONTROL_FILES`

  The `CONTROL_FILES` parameter specifies one or more names of control files. It is recommended that you keep at least three copies on different disk areas.

- `DB_WRITER_PROCESSES`

  The `DB_WRITER_PROCESSES` parameter specifies the initial number of database writer processes for an instance.

- `DB_BLOCK_SIZE`

  The `DB_BLOCK_SIZE` parameter determines the size of the database blocks.

- `DB_CACHE_SIZE`

  The `DB_CACHE_SIZE` parameter specifies the size of the buffer pool.

For more information about individual parameters, see SAP Note 1289199.

Oracle has traditionally stored initialization parameters in a text initialization parameter file, which is usually called as `init<DBSID>.ora`. Starting with Oracle 9i, you can choose to maintain initialization parameters in a binary server parameter file (logically called SPFILE, the name is `spfile<DBSID>.ora` or `spfile.ora`). The `spfile.ora` is a server-side initialization parameter file, maintained on the machine where the Oracle database server runs.

Oracle allows you to set, change, or delete (restore to default) the values of some initialization parameters dynamically. If you use the traditional initialization parameter file `init<DBSID>.ora`, such a change affects only the currently running instance because there is no mechanism to automatically update initialization parameters on disk. You must update initialization parameters manually and then restart the instance. Use a server parameter file to overcome this limitation because the initialization parameters stored in a server parameter file are persistent. This means that any changes made to the parameters while an instance is running can persist across the instance shutdown and startup.

SPFILE is fully supported by BR*Tools, including parameter maintenance, as of SAP Release 6.40. It is recommended that you use the SPFILE as part of the database upgrade to Oracle 9i.

> Caution:
> If you use SPFILE, do not make parameter changes on the Oracle level because this only changes parameter values in the SPFILE itself. With a parameter change in the SPFILE using BR*Tools, a new `init<DBSID>.ora` is generated automatically from the SPFILE so that the contents of the SPFILE and `init<DBSID>.ora` remain consistent. Certain SAP transactions (such as `DB02` or `ST04`) still rely on `init<DBSID>.ora`. If the SPFILE and `init<DBSID>.ora` are not kept consistent, these transactions show a status that is out-of-date in comparison with the SPFILE. If no `init<DBSID>.ora` exists, the status is not displayed. For this reason, it is also not possible to keep an old version of `init<DBSID>.ora` in the default location. If needed, create a copy of `init<DBSID>.ora` at the OS level.

Creation of Server Parameter File (SPFILE)

As of SAP Release 6.40, the SAP installation tools create the SPFILE. You can also create the server parameter file in SQL*Plus.

It is recommended that you create and store the SPFILE in the default location on the database server, which is the same as the default directory for `init<DBSID>.ora` (`$ORACLE_HOME/dbs` on UNIX and `%ORACLE_HOME%\database` on Windows). This makes administration of your database easier. For example, when the Oracle instance starts, it assumes this default location to read the parameter file.

> Hint:
> When an Oracle instance starts without the specification of a special profile (parameter file), it searches the default location for a file with one of the following names in the order:
>
> 1. `spfile<DBSID>.ora`
> 2. `spfile.ora`
> 3. `init<DBSID>.ora`
>
> The search is finished when one of the files is found and the instance starts using that profile, which is normally an SPFILE. If no SPFILE exists in the default directory, the instance is started with the standard initialization file. This means you can use the init file as a kind of backup for your SPFILE; this is one more reason to keep the init file consistent with the SPFILE.

You can create an SPFILE from standard parameter file `init<DBSID>.ora` using the `CREATE SPFILE` command, regardless of the instance state (the database does not need to be started to issue this statement).

The following commands show how to create an SPFILE:

```
SQL>CONNECT / AS SYSDBA

SQL>CREATE SPFILE FROM PFILE;
```

With the command `SQL>CREATE SPFILE FROM PFILE`, the system creates an SPFILE at the default location (platform-specific) under the default name. You must restart the instance so that the newly created server parameter file is used.

If a server parameter file of the same name already exists on the server, it is overwritten with the new information. If the instance is running and it has already been started using a server parameter file, an error is raised when you attempt to re-create the same server parameter file that is currently in use.

You cannot create an SPFILE from the current system status of a started instance. An SPFILE is always created from an existing Oracle parameter file.

You can have several SPFILEs on a computer, but only one of these can be active at any given time. When an instance is to be started, the DBA can use the `SPFILE` parameter to specify a specific server parameter file, possibly with a name different from the default SPFILE, to be used.

> Caution:
> To start the database using a different SPFILE than the default is not recommended for a standard operation. Use this only in emergency situations.

To view the current parameters set in the profile, start BRTOOLS or BRGUI and choose *Additional functions → Show profiles and logs → Oracle profile*. To view all Oracle parameters, start BRTOOLS or BRGUI and choose *Instance management → Show database parameters*. The column *Modif.* shows whether the parameter can be changed dynamically.

> Caution:
> You can open the binary server parameters file with a text editor and view its text, however, do not manually edit it. Doing so corrupts the file. You will not be able to start your instance, and if the instance runs, it can fail.

> Note:
> For more information about server parameter file, see SAP Note 601157.

Maintenance of Oracle Parameters

Oracle parameter classifications are as follows:

- Static

  The parameter modifications become effective after a restart of the instance.

- Dynamic

  Parameter can be modified while the instance is running.

- Deferred

  Parameters can be modified while the instance is running without affecting the sessions that were already opened at the time the statement is executed. The modified parameter only affects the sessions started later.

The scope of parameter changes is as follows:

- SPFILE

  The change is applied only in the Server Parameter File; therefore, the change is effective at the next setup and is persistent. This is the only scope allowed for static parameters.

- MEMORY

  The change is applied only in memory, not in the profile. This change is not persistent.

- BOTH

  The change is applied both in memory and the server parameter file. The effect is immediate and persistent.

BRSPACE supports changes of dynamic parameters in the memory, if the SPFILE is not used.

> Hint:
> For information about individual parameters, including the parameter class, see the Reference manual from the Oracle documentation.

Oracle parameters that require a restart of the Oracle instance after a change are called static parameters. Oracle parameters that can be immediately modified while running in memory are called dynamic parameters.

The classification of static and dynamic Oracle parameters does not depend on the use of a server parameter file.

If no server parameter file is used in an instance, you must create a persistent change to a parameter value by editing the standard initialization file. In this case, BRSPACE supports only modifications of dynamic parameters in memory, because Oracle statement `ALTER SYSTEM`, which is called by BRSPACE, also supports only modifications of dynamic parameters in memory.

Modification of Oracle Parameters

To modify an Oracle parameter, start BRTOOLS or BRGUI and choose *Instance management → Alter database parameters*. If you know which parameter to change, enter it in *Database parameter (parameter)* field and continue. To select the parameter from a list, do not make any changes. Choose *C* to continue and *Alter database parameter* and enter the position number of the parameter (not the name) you want to change from the list of parameters.

The following menu appears after BRTOOLS or BRGUI is started:

```
BR0657I Input menu 212 - please check/enter input values
-----------------------------------------------------------------------
 Options for alter of database parameter 'log_buffer' 1 * Parameter
description (desc) ..... [redo circular buffer size] 2 * Parameter
type (type) ........... [integer] 3 * Current parameter value
(parval) . [2097152] 4 * Value in spfile (spfval) ......... [100000]
5 ? New parameter value (value) ...... [] 6 - Scope for new value
(scope) ...... [spfile] 7 # Database instance (instance) ..... [] 8 ~
Comment on update (comment) ...... [] 9 - SQL command
(command) ........... [alter system set log_buffer = scope = spfile]
Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-----------------------------------------------------------------------
- BR0662I Enter your choice:
```

The information about the parameters is as follows:

- `desc`

  Displays parameter description from the *V$PARAMETER* view.

- `parval`

  Displays the current parameter value. The current parameter value can differ from the value in SPFILE when either the parameter was previously changed with scope MEMORY (only for dynamic parameters) or Oracle rounded the parameter value on startup.

- `spfval`

  Displays the current parameter value in SPFILE.

- `value`

  Displays the new parameter to be set.

- `comment`

  The comment you enter here is stored in the parameter file.

- `scope`

  The scope can be SPFILE, MEMORY, or BOTH. For static parameters, only SPFILE is possible and the parameter is activated when you restart the Oracle instance.

When you enter all the required information and continue, BRSPACE executes the corresponding SQL command to change the profile parameter. In addition, BRSPACE copies

SPFILE to the standard init file and creates an entry in `param<DBSID>.log` in the `sapreorg` directory to maintain a history of the parameter changes.

> Hint:
> Oracle 11g provides the option to save the current parameterization that is valid for an Oracle instance in a new profile file (pfile). You can do this with SQLPLUS by using the command `CREATE PFILE = ... FROM MEMORY`. You can also use the command `Create init.mem from memory` in the BR*Tools menu `Alter database parameters`.

Parameter Recommendations

Only optimum database parameter settings ensure that the database runs with high system performance and without errors. The parameter recommendations depend on which Oracle release and which SAP application you use.

The recommended Oracle releases and corresponding SAP Notes are as follows:

- Oracle 11g

  SAP Note 1431798: Oracle 11.2.0 – Database Parameter Settings

- Oracle 10g

  SAP Note 830576: Parameter recommendations for Oracle 10g

- Oracle 8 and 9i

  SAP Note 124361: Oracle parameters (R/3 Release 4.x or higher, Oracle 8.x/9.x)

- SAP Note 1171650: Automated Oracle DB parameter check

These notes contain SAP recommendations for the best configuration of the Oracle 11g and Oracle 10g databases in SAP environments. These parameter recommendations are subject to change. It is therefore recommended that you check the most recent version of this SAP Note at least once a month and make changes as necessary.

Prior to Oracle 10g, some parameter settings for the Oracle database (for example, for the cost-based optimizer) depended on whether you were using an ERP or a BW-based system. As of Oracle 10g, there are standard parameter settings for all systems, as described in this SAP Note. Any exceptions are mentioned explicitly.

For Oracle databases 8 and 9i, see SAP Note 124361. This SAP Note contains parameter recommendations for R/3 systems with SAP 4.0B or higher that do not use BW functions. This SAP Note also refers to other notes with parameter recommendations for other SAP applications, for example, Business Information Warehouse (BW), customer relationship management (CRM), strategic enterprise management (SEM), and so on.

Parameter Check

Enter SQL Statement | Result

NAME	RECOMMENDATION	I	U	REMARK
*** INFORMATION 1 ***	Command based on Note/Version: 1431798/33			
*** INFORMATION 2 ***	Command last changed: 2011-05-31			
*** INFORMATION 3 ***	Command Execution: 2011-06-15 10:17:51			
*** INFORMATION 4 ***	DB Startup: 2011-06-14 10:46:15			
*** INFORMATION 5 ***	DB SID: DEV			
*** INFORMATION 6 ***	DB Environment: OLTP, ABAP stack, not RAC, not ASM			
*** INFORMATION 7 ***	DB Platform: Microsoft Windows x86 64-bit			
*** INFORMATION 8 ***	DB Patchset: 11.2.0.1.0			
*** INFORMATION 9 ***	DB Optimizer Mergefix: SAP Bundle Patch (released 2010-09-10)			
*** INFORMATION 10 ***	Check "Event": reliable			
*** INFORMATION 11 ***	Check "_fix_control": reliable			
event (10995)	add with value "10995 level 2"			avoid flush shared pool during onli
_fix_control (4728348)	check if default value " " is suitable (set to 4728348:OFF if Winbundle <=10)	1	f	avoid wrong values; note 154767
_b_tree_bitmap_plans	check if default value " " is suitable (set to FALSE if Winbundle <=5)	2	p	
_enable_numa_support	check if default value " " is suitable (set optionally to TRUE after successful test)	2	p	
control_files	check if value "D:\ORACLE\DEV\SAPDATA1\CNTRL\CNTRLDEV.DBF, ..." is suitable			
db_cache_size	check if value "1258291200" is suitable (appropriately set)			
parallel_max_servers	check if value "20" is suitable (Number of DB machine CPU CORES*10)			Max used (gv$resource_limit): 0 (
pga_aggregate_target	check if value "1024000000" is suitable (appropriately set)			Max used MB (gv$pgastat): 289 (
shared_pool_size	check if value "1040187392" is suitable (appropriately set)			Max used MB (dba_hist_sgastat):
_optimizer_better_inlist_costing	check why set but mentioned with other prerequisites/not mentioned in note			
dml_locks	check why set but mentioned with other prerequisites/not mentioned in note			
job_queue_processes	check why set but mentioned with other prerequisites/not mentioned in note			
local_listener	check why set but mentioned with other prerequisites/not mentioned in note			
remote_login_passwordfile	check why set but mentioned with other prerequisites/not mentioned in note			
sga_max_size	check why set but mentioned with other prerequisites/not mentioned in note			

Figure 36: Result of the Automated Parameter Check

You can check the correct parameter settings using the SQL statement provided by SAP Note 1171650 - Automated Oracle DB parameter check. The system compares the current recommendations of the relevant parameter note, depending on the Oracle Release, patch set, and/or system type, with the actual database parameters. The SQL statement can be executed using *SQL Command Editor* of DBA Cockpit or the report *RSORADJV*.

The system automatically checks whether the current parameterization is correct, and issues a recommendation or an `OK` for the following parameters:

- Each official parameter
- Each underscore parameter that is recommended
- Each underscore parameter that is set but not recommended

The parameter recommendations are organized according to the type of recommendation. After the first `OK`, there are only `OKs`. Because there are some checks that either cannot be implemented or are time-consuming to implement in an individual SQL statement, you must manually check the remaining parameters.

The remaining parameters shown in the parameter check are as follows:

- Parameter name
- Recommendation
- Current/recommended value
- Parameter is set in parameter file (yes or no)
- Brief remark (for some parameters)
- Importance and Usage (columns *I* and *U*)

Important and Usage columns help estimate the impact of a wrong setting Column *I* is an indicator of the extent to which the parameter influences the database behavior.

Column *U* provides information about what is influenced by the parameter.

Columns I and U contain following additional information about the parameter classification:

Column	Parameter	Meaning
I	*1*	This parameter corrects errors, such as incorrect values. It also corrects performance values that can impact multiple queries.
I	*2*	This parameter corrects an important functional issue or is related to a performance parameter with an impact on special queries.
I	*" "*	This parameter is not known or a less important parameter.
U	*f*	This parameter is a functional parameter.
U	*p*	This parameter is a performance-related parameter.
U	*" "*	This parameter has no further classification.

The condition that all parameters must be set in accordance with the parameter note is still applicable. The Importance indicator is only an indicator of the urgency. Only parameters that are included in parameter notes are classified. The Importance and Usage indicators cannot be specified for parameters that are not tested by SAP.

Oracle Error and Diagnosis Files

The Oracle database 11g includes fault diagnosability infrastructure. It is designed to help prevent, detect, diagnose, and resolve problems, such as code bugs, metadata corruption, and customer data corruption.

Starting with the Oracle 11g database, diagnostic data, such as the alert log, trace files, and health monitor reports, is stored in a file-based repository called Automatic Diagnostic Repository (ADR).

ADR is a directory structure that is stored outside the database. Therefore, ADR is available for problem diagnosis when the database is down.

The ADR root directory is known as ADR base. Its location of ADR base is set by the `DIAGNOSTIC_DEST` initialization parameter.

If this parameter is omitted or left null, upon startup the database sets `DIAGNOSTIC_DEST` as follows:

- If environment variable `%ORACLE_BASE%` is set, `DIAGNOSTIC_DEST` is set to the directory designated by `%ORACLE_BASE%`.
- If environment variable `%ORACLE_BASE%` is not set, `DIAGNOSTIC_DEST` is set to `%ORACLE_HOME%/log`.

> Hint:
> Parameters `user_dump_dest` and `background_dump_dest` are depreciated in Release 11g.

Within the ADR base, there can be multiple ADR homes. Each ADR home is the root directory for all diagnostic data (traces, dumps, the alert log, and so on) for a particular instance of a particular Oracle product or component. For example, in an Oracle real application cluster (RAC) environment with Oracle Automatic Storage Management (ASM), each database instance, Oracle ASM instance, and listener has an ADR home. ADR homes reside in ADR base subdirectories that are named according to the product or component type.

In an SAP environment, the ADR base is set as follows:

`DIAGNOSTIC_DEST = %SAPDATA_HOME%\SAPTRACE`

The location of each ADR home is given by the following path, which starts at the ADR base directory:

`diag/product_type/product_id/instance_id,`

where `product_id = DB_NAME, instance_id = %ORACLE_SID%`, and `product_type` can be `rdbms tnslsnr, clients`, or `asm`.

The following example shows the ADR home and ADR base for an Oracle database (`product_id = DEV, instance_id = DEV`, and `product_type =rdbms`):

- Location of ADR base = `/oracle/DEV/saptrace`
- Location of ADR home = `/oracle/DEV/saptrace/diag/rdbms/DEV/DEV`

The central file of the fault diagnosability infrastructure is the Oracle Alert Log, which logs significant database events and messages.

Examples of database events and messages are as follows:

- Startup and shutdown
- Non-default values of parameters
- Errors and warnings
- Log switches and checkpoints
- Media recovery
- Creation of tablespaces

The alert log is stored in an XML file and ASCII file.

In the SAP environment, XML and ASCII files are located in the following directories:

- XML

  `/oracle/<DBSID>/saptrace/diag/rdbms/<DBSID>/<DBSID>/alert`

The file name is `log.xml`.

- ASCII

  `/oracle/<DBSID>/saptrace/diag/rdbms/<DBSID>/<DBSID>/trace`

  The file name is `alert_<DBSID>.log`.

Oracle Alert Log

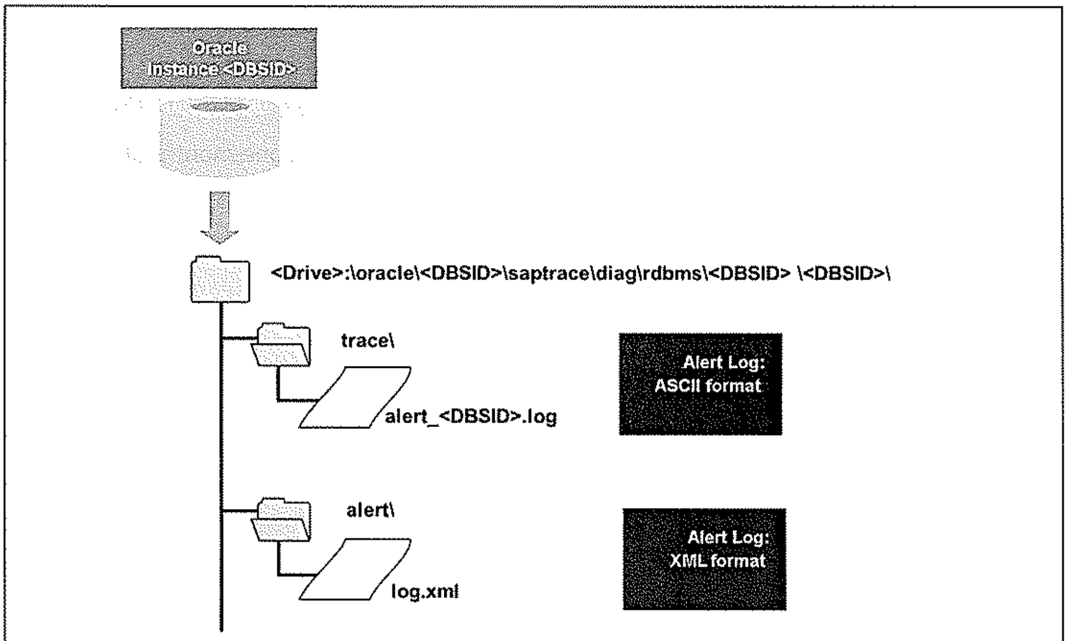


Figure 37: Oracle Alert Log

The evaluation and analysis of the alert log can be performed using either an ASCII editor or Oracle ADR Command Interpreter (ADRCI).

Oracle ADRCI is a command line utility that enables you to perform the following activities:

- Investigate problems
- View health check reports
- Package and upload first-failure diagnostic data to Oracle support

You can also use the utility to view the names of the trace files in the ADR and alert log with XML tags stripped, with and without content filtering. ADRCI has a rich command set that you can use interactively or in scripts.

For more information about ADRCI, see SAP Note 1431751.

> Note:
> Up to Oracle 10g, the location of the alert log is specified by the `BACKGROUND_DUMP_DEST` parameter. The default location is `%SAPDATA_HOME%/saptrace/usertrace`.

> Hint:
> The alert file growth is unlimited; therefore, you can delete it periodically. You can delete the file even when the database instance is running (see SAP Note 786032).

Additionally with Oracle 11g, the retention policy for ADR allows to specify how long the Oracle database must keep the data.

ADR incidents are controlled by the following policies:

- The incident metadata retention policy (default is one year: LONGP_POLICY = 8760 hours)
- The incident files and dumps retention policy (default is one month: SHORTP_POLICY = 720 hours)

These retention times can be controlled using ADRCI.

Unit 1
Exercise 4

Change Oracle Parameters

Business Example

You have received a recommendation from SAP EarlyWatch to change an Oracle parameter.

Set BR*Tools to turn on archiving for your database, create a server parameter file for Oracle from `init<DBSID>.ora`, and change a parameter.

1. Use BR*Tools to check whether archiving is turned on.
2. Use BR*Tools to turn archiving on for your database.
3. Create the server parameter file from `init<DBSID>.ora`.

> Caution:
> After the installation of SAP Web Application Server (AS) prior to 6.40, you must create the server parameter file using SQL*Plus. The installation routine for SAP Web AS 6.40 and higher creates the server parameter file automatically during installation.

4. Change parameter `open_cursors` to **600**. Is it possible to perform this dynamically? Check the setting before and after restarting the database.

> Caution:
> This parameter is set to a value that is not recommended for SAP systems. The database system check will show a warning about the wrong value.

Unit 1
Solution 4

Change Oracle Parameters

Business Example

You have received a recommendation from SAP EarlyWatch to change an Oracle parameter.

Set BR*Tools to turn on archiving for your database, create a server parameter file for Oracle from `init<DBSID>.ora`, and change a parameter.

1. Use BR*Tools to check whether archiving is turned on.
   a) Call BRGUI or BRTOOLS and choose *Instance management → Show instance status.*
   b) Choose *Continue* two times.

      The following menu appears:

      ```
      Information about the status of database instance T99

       1 - Instance number (number) ........... 1
       2 - Instance thread (thread) ........... 1
       3 - Instance status (status) ........... OPEN
       4 - Instance start time (start) ........ 2011-05-31 16.27.08
       5 - Oracle version (version) ........... 11.2.0.1.0
       6 - Database creation time (create) .... 2011-05-27 12.02.35
       7 - Last resetlogs time (resetlogs) .... 2011-05-27 12.02.35
       8 - Archivelog mode (archmode) ......... NOARCHIVELOG
       9 - Archiver status (archiver) ......... STOPPED
      10 - Current redolog sequence (redoseq) . 112
      11 - Current redolog SCN (redoscn) ...... 575099
      12 - Flashback status (flashback) ....... OFF
      13 - Block change tracking (tracking) ... OFF
      14 - Data encryption (encryption) ....... OFF
      15 - Database vault (dbvault) ........... OFF
      16 - Number of SAP connections (sapcon) . 0
      ```

      The output shows that the archiver is started, but the archive log mode is NOARCHIVELOG, which means that archiving is turned off.

2. Use BR*Tools to turn archiving on for your database.
   a) Call BRGUI or BRTOOLS and choose *Instance management → Alter database instance.*
   b) Choose *Continue.*
   c) In *Alter database instance main menu*, choose *Set archivelog mode*.
   d) Choose *Continue.*

      Your database instance stops and is restarted in the mount state, the archive log mode turns on, and the database opens.

3. Create the server parameter file from `init<DBSID>.ora`.

> **Caution:**
> After the installation of SAP Web Application Server (AS) prior to 6.40, you must create the server parameter file using SQL*Plus. The installation routine for SAP Web AS 6.40 and higher creates the server parameter file automatically during installation.

a) Create the SPFILE using the following command:

```
D:\oracle\T99>sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Di Mai 31 16:36:20
2011

Copyright (c) 1982, 2010, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL> create spfile from pfile;

File created.
SQL> exit
```

b) Restart your database using BRGUI or BRTOOLS and choosing *Instance management → Shut down database and then Instance management → Start up database*.

4. Change parameter `open_cursors` to **600**. Is it possible to perform this dynamically? Check the setting before and after restarting the database.

> **Caution:**
> This parameter is set to a value that is not recommended for SAP systems. The database system check will show a warning about the wrong value.

a) Start BRGUI or BRTOOLS and choose *Instance management → Alter database parameters*.

b) To start BRSPACE, choose *Continue* in the next input menu, `BRSPACE options for alter database parameter`.

c) In *BRSPACE options for alter database parameter*, enter the following data:

Parameter	Value
Alter parameter action (action)	**change**
Database parameter (parameter)	**open_cursors**

d) Choose Continue.

e) In the *Options for alter of database parameter 'open_cursors'* input menu, enter **600** in the *New parameter value (value)* field.

The parameter change is done in memory (dynamically) and the server profile because the `both` scope is offered:

```
Options for alter of database parameter 'open_cursors'

 1 * Parameter description (desc) ..... [max # cursors per
session]
 2 * Parameter type (type) ........... [integer]
 3 * Current parameter value (parval) . [800]
 4 * Value in spfile (spfval) ........ [&lt;same>]
 5 - New parameter value (value) ...... [600]
 6 - Scope for new value (scope) ...... [both]
 7 # Database instance (instance) ..... []
 8 ~ Comment on update (comment) ...... []
 9 - SQL command (command) ........... [alter system set
     open_cursors = 600 scope = both]
```

f) To check whether the parameter has changed dynamically as well as permanently, check the parameter setting before and after restarting the database instance. Use BRGUI or BRTOOLS and choose *Instance management → Show database parameters*.

LESSON SUMMARY

You should now be able to:

- Change Oracle parameters

Learning Assessment

1. The system global area (SGA) consists of the database buffer, the redo log buffer, and the ______.

   Choose the correct answer.

   - [ ] A shared pool
   - [ ] B data blocks
   - [ ] C rollback segments
   - [ ] D clusters

2. Which of the following statements is true?

   Choose the correct answers.

   - [ ] A Data files can be mirrored by Oracle.
   - [ ] B Online redo log files can be mirrored by Oracle.
   - [ ] C Control files can be mirrored by Oracle.
   - [ ] D Parameter files can be mirrored by Oracle.

3. Which of the following options are required for minimal security of an Oracle database?

   Choose the correct answers.

   - [ ] A The database must be installed on a cluster.
   - [ ] B The online redo log files must be mirrored.
   - [ ] C Online redo log files and data files must reside on different disks.
   - [ ] D Archiving must be turned on.
   - [ ] E Offline redo log files and data files must reside on different disks.

4. Which environmental variable stores files like init<DBSID>.ora or spfile<DBSID>.ora in its subdirectories?

   Choose the correct answer.

   - [ ] A ORACLE_SID
   - [ ] B ORACLE_HOME
   - [ ] C SAPDATA_HOME
   - [ ] D ORA_NLS10

5. Which users are standard Oracle users created by the Oracle installer?

   Choose the correct answers.

   - [ ] A SYSTEM
   - [ ] B SYS
   - [ ] C SAP<SCHEMA-ID>
   - [ ] D OPS$<HOSTNAME>/<SAPSID>ADM (Windows) and OPS$<SAPSID>ADM (UNIX)
   - [ ] E SYSDBA

6. Which users are standard Oracle users created by the SAP installation tool?

   Choose the correct answers.

   - [ ] A SYSTEM
   - [ ] B SYS
   - [ ] C SAP<SCHEMA-ID>
   - [ ] D OPS$<HOSTNAME>/<SAPSID>ADM (Windows) and OPS$<SAPSID>ADM (UNIX)
   - [ ] E SYSDBA

7. The correct procedure to change the password of database user SAP<SCHEMA-ID> is to use the Oracle command ALTER USER.

   Determine whether this statement is true or false.

   - [ ] True
   - [ ] False

8. The Oracle tool to test the connection to the listener is called ______. To test the connection to the database with SID C11 on host twdf0505, enter tnsping C11 at the operating system (OS) level.

   Choose the correct answer.

   - [ ] A .PING
   - [ ] B lsnrctl start
   - [ ] C TNSPING
   - [ ] D NSLOOKUP

9. Which of the following SAP tools are used for Oracle administration?

   Choose the correct answers.

   - [ ] A BRTOOLS
   - [ ] B BRSPACE
   - [ ] C BRRECOVER
   - [ ] D BRFRULES

10. After creating a server profile from init<DBSID>.ora, you can change Oracle profile parameters at both the OS and database level.

    Determine whether this statement is true or false.

    - [ ] True
    - [ ] False

11. When using the server parameter file, which of the following statements are true?

    Choose the correct answers.

    - [ ] A After changing the parameters using scope BOTH, the database must be restarted.
    - [ ] B After changing the parameters using scope MEMORY, the database must be restarted.
    - [ ] C After changing the parameters using scope SPFILE, the database must be restarted.
    - [ ] D With scope MEMORY, the previous parameter value is used after restart of the database instance.

12. When stopping a database using the _____ shutdown mode, instance processes are stopped immediately and no rollback of open transactions is performed during the shutdown.

Choose the correct answer.

- [ ] A NORMAL
- [ ] B TRANSACTIONAL
- [ ] C IMMEDIATE
- [ ] D ABORT

13. Which of the following shutdown modes leave the database in a consistent state?

Choose the correct answers.

- [ ] A NORMAL
- [ ] B TRANSACTIONAL
- [ ] C IMMEDIATE
- [ ] D ABORT

Learning Assessment - Answers

1. The system global area (SGA) consists of the database buffer, the redo log buffer, and the _____.

   Choose the correct answer.

   - [X] A shared pool
   - [ ] B data blocks
   - [ ] C rollback segments
   - [ ] D clusters

2. Which of the following statements is true?

   Choose the correct answers.

   - [ ] A Data files can be mirrored by Oracle.
   - [X] B Online redo log files can be mirrored by Oracle.
   - [X] C Control files can be mirrored by Oracle.
   - [ ] D Parameter files can be mirrored by Oracle.

3. Which of the following options are required for minimal security of an Oracle database?

   Choose the correct answers.

   - [X] A The database must be installed on a cluster.
   - [ ] B The online redo log files must be mirrored.
   - [ ] C Online redo log files and data files must reside on different disks.
   - [X] D Archiving must be turned on.
   - [X] E Offline redo log files and data files must reside on different disks.

4. Which environmental variable stores files like init<DBSID>.ora or spfile<DBSID>.ora in its subdirectories?

   Choose the correct answer.

   - [ ] A ORACLE_SID
   - [x] B ORACLE_HOME
   - [ ] C SAPDATA_HOME
   - [ ] D ORA_NLS10

5. Which users are standard Oracle users created by the Oracle installer?

   Choose the correct answers.

   - [x] A SYSTEM
   - [x] B SYS
   - [ ] C SAP<SCHEMA-ID>
   - [ ] D OPS$<HOSTNAME>/<SAPSID>ADM (Windows) and OPS$<SAPSID>ADM (UNIX)
   - [ ] E SYSDBA

6. Which users are standard Oracle users created by the SAP installation tool?

   Choose the correct answers.

   - [ ] A SYSTEM
   - [ ] B SYS
   - [x] C SAP<SCHEMA-ID>
   - [x] D OPS$<HOSTNAME>/<SAPSID>ADM (Windows) and OPS$<SAPSID>ADM (UNIX)
   - [ ] E SYSDBA

7. The correct procedure to change the password of database user SAP<SCHEMA-ID> is to use the Oracle command ALTER USER.

   Determine whether this statement is true or false.

   - [ ] True
   - [x] False

8. The Oracle tool to test the connection to the listener is called ______. To test the connection to the database with SID C11 on host twdf0505, enter tnsping C11 at the operating system (OS) level.

   Choose the correct answer.

   - [ ] A .PING
   - [ ] B lsnrctl start
   - [x] C TNSPING
   - [ ] D NSLOOKUP

9. Which of the following SAP tools are used for Oracle administration?

   Choose the correct answers.

   - [x] A BRTOOLS
   - [x] B BRSPACE
   - [x] C BRRECOVER
   - [ ] D BRFRULES

10. After creating a server profile from init<DBSID>.ora, you can change Oracle profile parameters at both the OS and database level.

    Determine whether this statement is true or false.

    - [x] True
    - [ ] False

11. When using the server parameter file, which of the following statements are true?

    Choose the correct answers.

    - [x] A After changing the parameters using scope BOTH, the database must be restarted.
    - [x] B After changing the parameters using scope MEMORY, the database must be restarted.
    - [ ] C After changing the parameters using scope SPFILE, the database must be restarted.
    - [ ] D With scope MEMORY, the previous parameter value is used after restart of the database instance.

12. When stopping a database using the _____ shutdown mode, instance processes are stopped immediately and no rollback of open transactions is performed during the shutdown.

    Choose the correct answer.

    - [ ] A NORMAL
    - [ ] B TRANSACTIONAL
    - [ ] C IMMEDIATE
    - [x] D ABORT

13. Which of the following shutdown modes leave the database in a consistent state?

    Choose the correct answers.

    - [x] A NORMAL
    - [x] B TRANSACTIONAL
    - [x] C IMMEDIATE
    - [ ] D ABORT

UNIT 2 Backup, Restore, and Recovery

Lesson 1

Reviewing a Backup Strategy 102
Exercise 5: Create a Backup Strategy 113

Lesson 2

Reviewing Backup Tools 116
Exercise 6: Use Backup Tools 133

Lesson 3

Performing Backups 153
Exercise 7: Perform Backups 169

Lesson 4

Performing Restore and Recovery 173
Exercise 8: Perform Restore and Recovery to the System 193

Lesson 5

Working with Advanced Backup Techniques 205

UNIT OBJECTIVES

- Create a backup strategy
- Work with backup tools
- Perform backups to the system
- Perform restore and recovery to the system
- Apply advanced backup techniques

Unit 2
Lesson 1

Reviewing a Backup Strategy

LESSON OVERVIEW

This lesson explains how to plan and test the backup strategy before you perform backups. This lesson also describes how to define a backup strategy for different database sizes.

Business Example

Until now, you performed a daily offline backup of the database that starts at 8 p.m., but now that employees of your subsidiaries in Singapore, Boston, and Poland also have access to the SAP system, it must be constantly online. You need a new backup strategy that enables the SAP system to be up 24 hours a day. For this reason, you require the following knowledge:

- An understanding of how to explain the importance of backups
- An understanding of how to list the different backup types (offline, online, partial, and incremental backup)
- An understanding of how to explain the special importance of backups of the archived redo log files
- An understanding of how to define a backup strategy depending on database size, tape capacity, and available time for restore or recovery

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Create a backup strategy

Backup Strategy

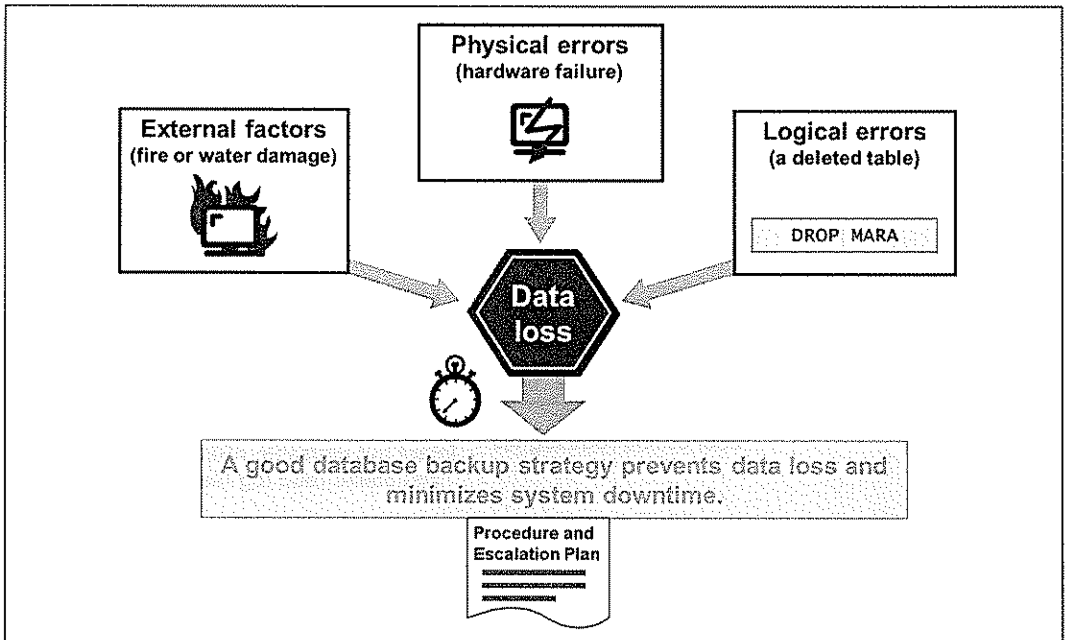


Figure 38: Importance of Backups

Performing a proper database backup requires special actions. Compared to normal backup of files, a database is a collection of files that are dependent on each other. If you lose a single file of the database, simply restoring it is rarely sufficient.

Reasons for Data Loss

Business application data of an SAP system, stored in a relational database, is usually dynamic and requires a comprehensive security strategy. If you do not have a suitable backup strategy, external factors, physical errors, and logical errors can cause system downtime and lead to data loss.

If data is lost due to external factors, such as water damage to your hardware or physical errors, such as hardware failure, you must recover the database up to the point in time when the database crashed. If this complete recovery is possible, only the data of transactions uncommitted at the time of error is lost.

If data is lost due to logical errors, such as unintentional deletion of a table, you must recover the database up to a point in time shortly before the error occurred.

Design your backup strategy as per your company's need. To ensure the availability of your SAP system, your backup strategy must be tested before your SAP system goes live, and again after any changes to your backup strategy.

When planning your backup strategy, take into account the maximum downtime for each of the recovery scenarios.

To ensure that the correct steps are performed for each of the scenarios, create a document containing organizational descriptions of procedures and an escalation plan. The person who restores and recovers a database must understand this document.

Evaluate and implement the most suitable backup type and method for your company. SAP provides tools that support different types of backups, such as online backups, incremental backups with Oracle Recovery Manager (RMAN), and split-mirror backups.

Importance of Redo Log Files

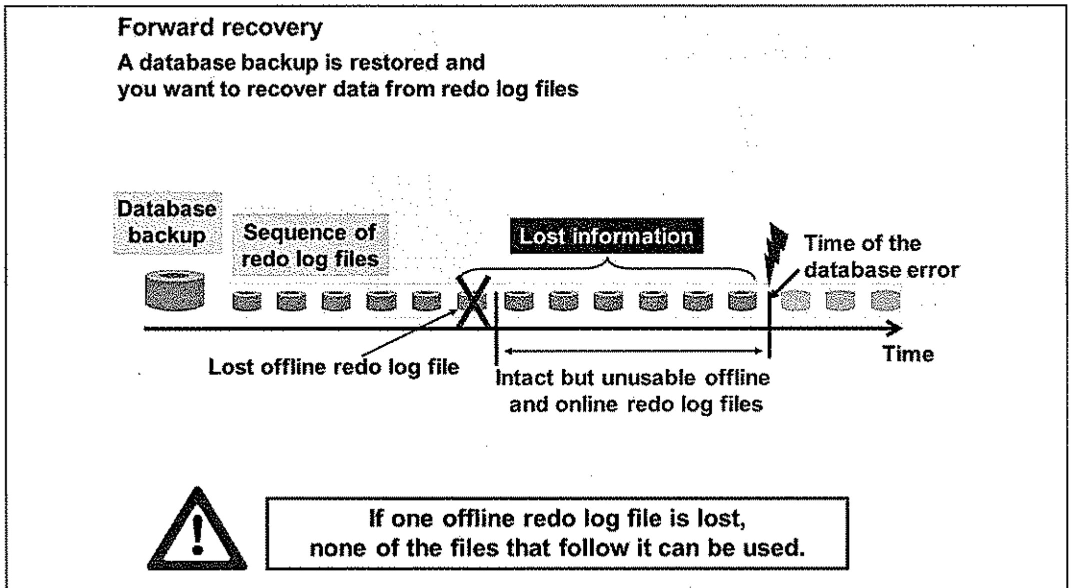


Figure 39: Importance of Redo Log Files

When you want to perform a complete recovery, to recover the data to the (committed) state that the data had at the time of crash, you need all the offline and online redo log files written from the point in time of the last database backup.

If a file is missing from the chain of offline redo log files, a restore of subsequent offline redo log files and corresponding recovery of the database are not possible. You can then only perform a point-in-time recovery, using all offline redo log files older than the lost one. This will result in a loss of data changes performed from the point at which you lost the offline redo log file. Therefore, keep at least two copies of all offline redo log files on disks or on tapes.

Disaster Recovery

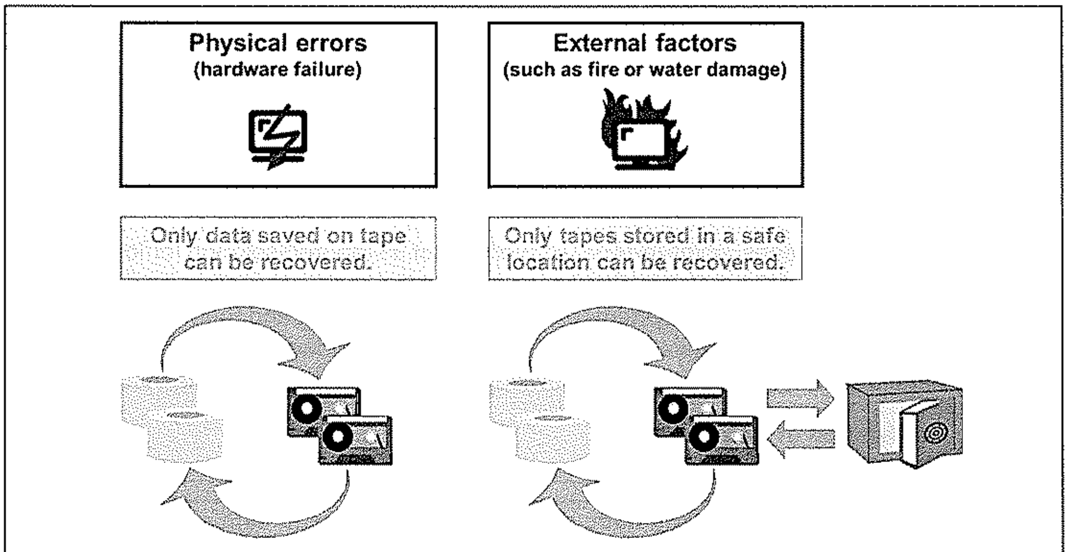


Figure 40: Disaster Recovery

If you have a hardware failure, you can lose a disk, a disk subsystem, or the complete hardware set. In this case, only data backed up on external media, such as tapes, can be restored. This includes the offline redo log files. Redo log information that is not stored on tapes can be lost.

If data loss occurs due to external factors, such as fire or water damage, all backup media that is not stored in a safe location can be lost.

Recovery from a Logical Error

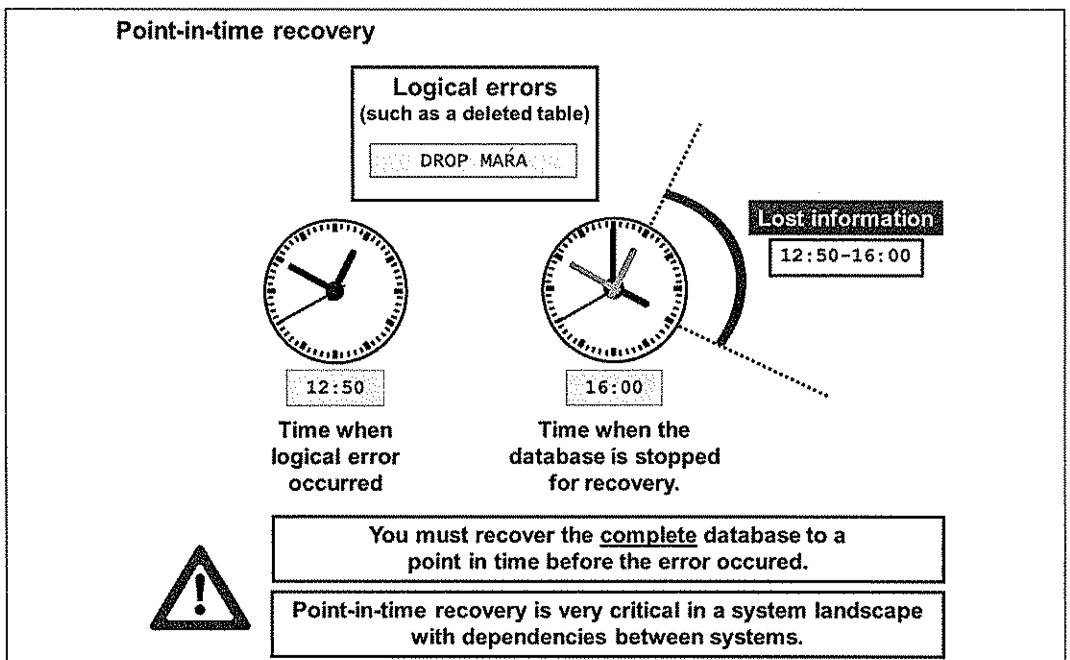


Figure 41: Recovery from a Logical Error

Point-in-time recovery is needed in other situations, as well, for example, when you need to reset your system to a state before an upgrade and a complete database backup corresponding to that point in time is not available.

When performing a point-in-time recovery, you must recover the complete database, because the data from different tables must be consistent. Consequently, all data changes made between the time chosen for point-in-time recovery, and the time the database is stopped for recovery, is lost. This is especially critical in a system landscape with dependent data stored in two or more systems. If performed in just one of the systems, point-in-time recovery results in data inconsistencies.

For these reasons, point-in-time recovery is not a standard solution for logical errors in production systems. Depending on the table, it is possible to restore and recover the database on a different system (on a different computer), and then import the missing table or the missing table rows from that system to your production system. This method prevents data loss, but it requires expert knowledge of the application module that uses the table.

Verification of Data

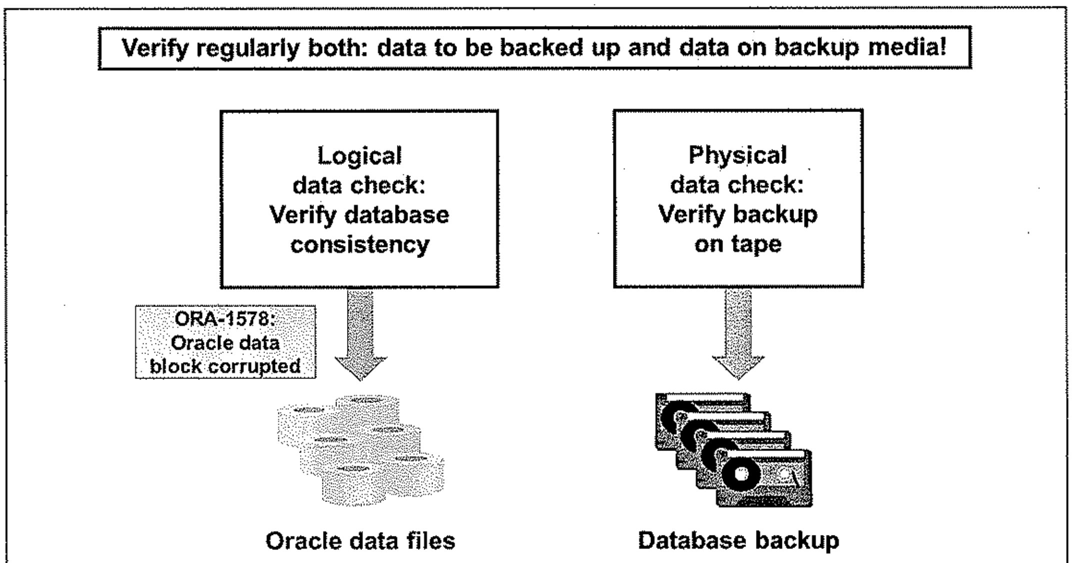


Figure 42: Verification of Data

Make sure your backup strategy includes verifying the data to be backed up, as well as the data on tapes. To verify the consistency of the Oracle database, perform a logical data check.

A logical data check discovers the following corrupt data blocks:

- Corrupt Oracle blocks (error ORA-1578) can appear in your database because of operating system (OS) or hardware errors.
- Without a logical data check, corrupt data blocks are detected only when Oracle processes access these data blocks while attempting to access a table within the database. Corrupt blocks that are accessed rarely may remain undetected in your system for a long time.
- Corrupt Oracle blocks are not recognized during a backup; therefore, a database backup can contain corrupt blocks. They make the backup unusable because they are restored in the database in exactly the same state.

Perform logical data checks at regular intervals, preferably once a week. For optimal performance, carry out this check during periods of low system activity, for example, on weekends.

To verify the tapes used for a database backup, perform a physical data check. During this check, read the tapes and examine the physical correctness of the data transferred.

At the end of an offline backup, you can check at the binary level whether the files read from tape are identical to those in the database. This requires the database to remain closed during the procedure.

After an online backup, during which data changes can occur in the database, you can only check that all files on the tapes are readable.

Backup Cycle

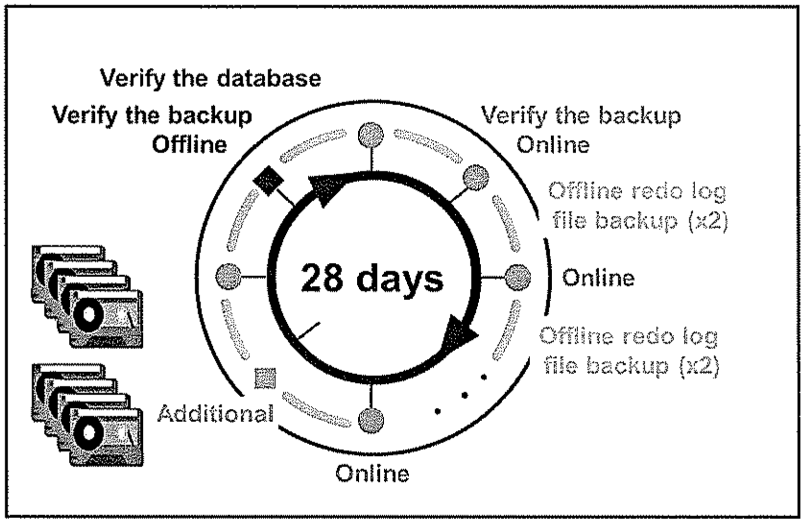


Figure 43: Backup Cycle

A backup cycle is a time period during which you keep the backups on your tapes. The length of the backup cycle is the retention period for your tapes. A tape is reused only if the backup on it is older than the retention period.

SAP recommends a backup cycle of four weeks. The backup cycle must be the same for database backups and offline redo log backups.

Recommended backup cycle steps:

- Perform a complete online backup each workday.
- Perform a complete offline backup at least once in the cycle.
- Back up the offline redo log files on each workday and after every online and offline backup. Ensure that you back up every offline redo log file twice, on separate tapes, before the file is deleted in the archive directory.
- To verify a backup, carry out a logical check of the database before or after the backup and check the backup for physical errors. You must perform backup verification at least once in the backup cycle. However, SAP recommends that you do it once a week.

- Remove the last verified full offline backup of each cycle from the tape pool, and keep this backup in long-term storage. The removed tapes must be replaced with new ones.

The relation between the backup cycle length and the frequency of complete database backups should be such that you always have several generations of complete backups. This protects you from data loss, even in a situation in which your last database backup cannot be found or is unusable.

Changes to the database file structure affect the subsequent database restore. These changes occur when a data file is added, when a data file is moved to a different location, or when a tablespace and its data files are reorganized. Perform additional backups after each database reorganization and system upgrade. Place these additional backups in long-term storage.

You can also perform additional backups after database structure modification, but doing so is not required. BRRECOVER performs such modifications automatically during recovery.

Backup Methods

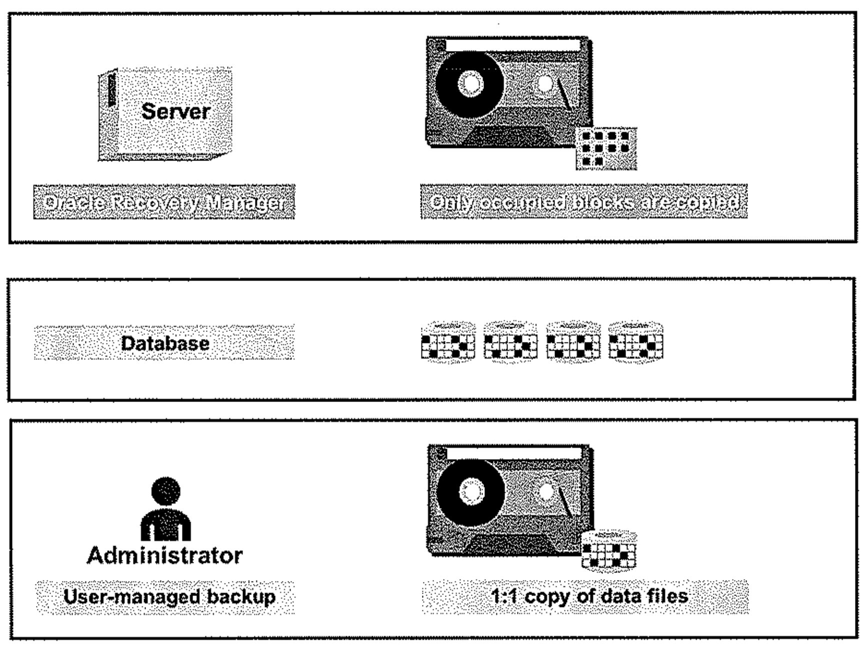


Figure 44: Backup Methods

The main methods of backing up an Oracle database are as follows:

- User-managed backup

  In this method, the administrator performs a manual backup on the OS. Backups with OS errors are created and are written back if a recovery is necessary.

- Server-managed backup

  This method uses the RMAN, which performs the backups using shadow processes on the Oracle server.

For these backup methods, different backup types are available on two levels for an Oracle database.

These backup types depend on the following conditions:

- Whether the database is open or closed or open during the backup procedure
- Whether the database is backed up completely, and if is not backed up completely, the part of the database that is backed up

Backup Types

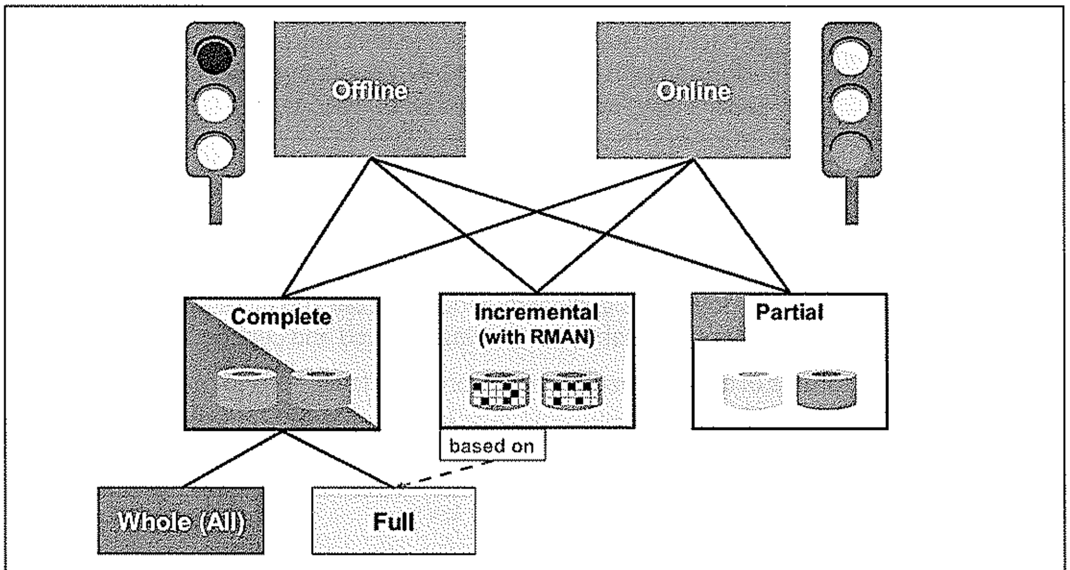


Figure 45: Backup Types

The different backup types are as follows:

- Offline backup

  The database is shut down before the backup. Database files are, therefore, copied to the backup media in a closed state. If the database has been shut down consistently (standard when using SAP tools), the data is backed up in a consistent state, and can be restored and opened even without redo log files.

- Online backup

  The database remains open during the backup, and activities can take place (such as data modification) at this time. Data blocks copied to the backup medium at the beginning of the backup procedure can correspond to an older system change number than those backed up at the end. To restore a consistent database from such a backup, you need at least the redo information written during the time of the backup procedure. With this redo information, all blocks in the restored database can be recovered to a state corresponding to the time of the end of the backup.

> Hint:
> Performance of the system may go down slightly during an online backup, so online backups should be scheduled at times of low activity.

- Complete backup

  All data in the database is backed up. When you perform a full backup, after backing up all the data in the database, the RMAN writes an additional piece of information (the catalog information) to the control file. This makes it possible to create incremental backups. A whole backup creates a backup of the entire data without the catalog information. In terms of the database data, there is no difference between a whole backup and a full backup.

> Hint:
> When a full backup is performed, RMAN is always used to create and append the backup catalog information to the control file. The backup itself can be performed with or without RMAN, depending on further backup settings (`tape_copy_command` and `backup_dev_type`).

- Incremental backup

  If you have created a full backup, you can later back up just those data blocks that have changed since the time of the full backup. This will reduce the amount of data to be backed up. However, it does not significantly reduce the backup time, because the block must be read to verify that the database block was changed and consequently must be backed up.

  The features of an incremental backup are as follows:

  - An incremental backup can only be based on a previous full backup.

  > Caution:
  > An incremental backup is not useful if the corresponding full backup is already overwritten. When performing incremental backups, it is highly recommended that you perform at least one full backup per week and four full backups per backup cycle.

  - Like a full backup, an incremental backup is controlled by RMAN, which uses the control file for this purpose.
  - SAP tools only supports a cumulative incremental backup. This means that each incremental backup saves all blocks that were modified since the last full backup, not since the last incremental backup.
  - An incremental backup performed with SAP tools is always a backup of the whole database. You cannot choose individual data files to be backed up with an incremental backup.
  - When performing an incremental backup, Oracle reads all blocks of all data files to check which ones have been changed. Therefore, an incremental backup can reduce the backup time only if a long backup runtime was caused by low throughput on the tape stations.

- Partial backup

  If a complete backup takes too long, you can choose to back up the database in smaller parts. The sum of individual partial backups must cover the entire database, for example, during one week.

> Caution:
> Although both Oracle and SAP tools support the recovery of data files from different backup runs, this type of recovery requires all offline and online redo log files generated since the backup of the oldest data files; therefore, the recovery process can be very time consuming.

Backup Strategy – Example 1

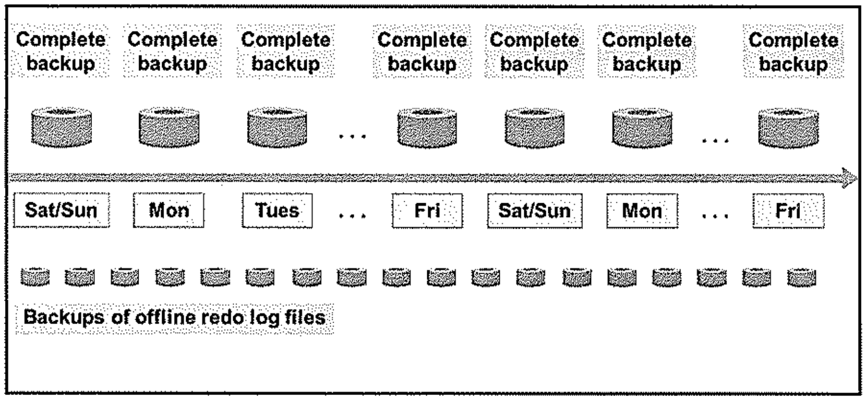


Figure 46: Backup Strategy – Example 1

Depending on factors such as database size, tape size, availability, and other factors, you can define a suitable backup strategy.

You can use a simple backup strategy consisting of regular complete database backups and offline redo log file backups. To recover from a media error (disk), you restore missing database files from a complete backup (preferably from the last one), restore offline redo log files written during and after this backup (those that have already been deleted from the archive directory), and completely recover the database.

Backup Strategy – Example 2

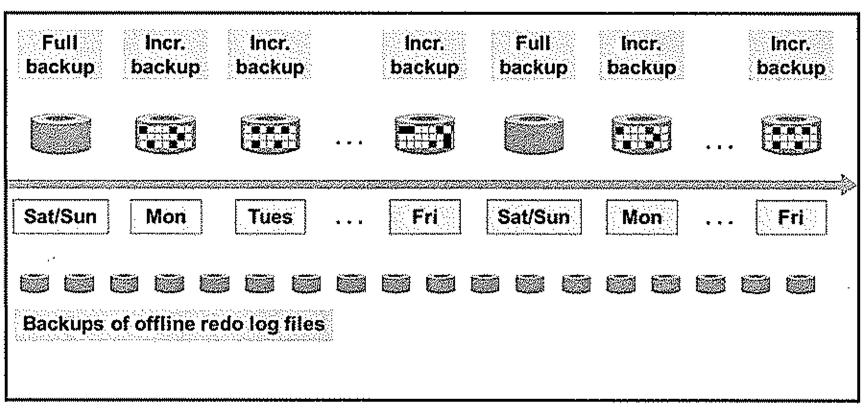


Figure 47: Backup Strategy – Example 2

In a system where incremental backups can reduce backup time, you can perform complete backups less often and replace them with incremental backups. The complete backups must be full backups.

To recover from a disk error, perform the following steps:

1. Restore missing database files from the last full backup.
2. Update them with a restore from the last incremental backup.
3. Recover with the help of redo information written during and after the last incremental backup.

Backup Strategy – Example 3

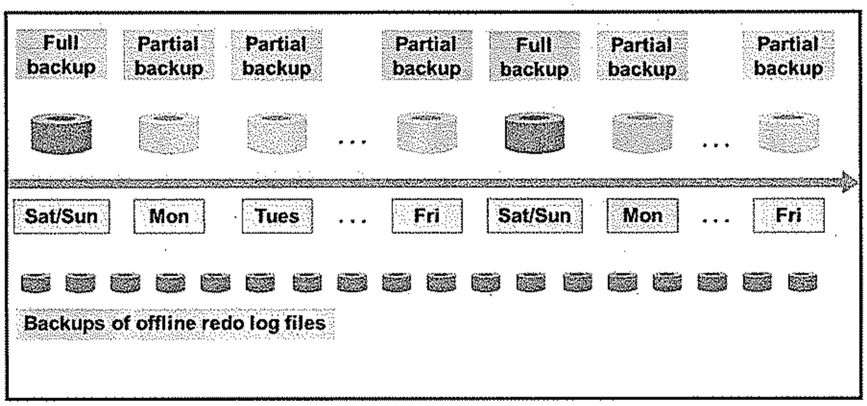


Figure 48: Backup Strategy – Example 3

If you replace a full backup with a partial backup during a week, the runtime of the individual backups is reduced. The recovery may take longer in the event of errors in the storage medium. For example, if a crash occurs on Thursday and the last backup containing lost files is from Monday, you must redo all data modifications performed in these files since Monday.

Unit 2
Exercise 5

Create a Backup Strategy

Business Example

You need to plan a good backup strategy.

Evaluate the given backup strategy.

1. Technical specifications: The planned size of the database is roughly 100 GB. A maximum of 50 online redo log files of 20 MB are expected to be written daily. Three tape devices are available, and each one can write or read up to 6 GB per hour. The tapes have a capacity of 40 GB. It takes, on average, three minutes to apply an offline redo log file during the recovery.

   Strategy: An online backup is performed every night. Three tapes are reserved for each night. The database administrator performs a backup of the offline redo log files daily and deletes the offline redo log files from the disk afterward.

   Is this a good backup strategy?

   Can a full restore be performed in 8.5 hours?

   What is the significance for an instance recovery if the error that led to the restore and recovery operation occurred during a long background-processing job without a commit?

Unit 2
Solution 5

Create a Backup Strategy

Business Example

You need to plan a good backup strategy.

Evaluate the given backup strategy.

1. Technical specifications: The planned size of the database is roughly 100 GB. A maximum of 50 online redo log files of 20 MB are expected to be written daily. Three tape devices are available, and each one can write or read up to 6 GB per hour. The tapes have a capacity of 40 GB. It takes, on average, three minutes to apply an offline redo log file during the recovery.

   Strategy: An online backup is performed every night. Three tapes are reserved for each night. The database administrator performs a backup of the offline redo log files daily and deletes the offline redo log files from the disk afterward.

   Is this a good backup strategy?

   Can a full restore be performed in 8.5 hours?

   What is the significance for an instance recovery if the error that led to the restore and recovery operation occurred during a long background-processing job without a commit?

   a) When using this backup strategy, the issue is that only one copy of the archived redo log files is written to tape before deletion. SAP recommends that at least two copies be written to different backup media. The data is distributed automatically by BRBACKUP across the tape devices, so that the backup can be performed unattended, even if the files are not compressed.

      If the data volume is distributed over the three backup media, each tape will contain approximately 33 GB. At a read rate of 6 GB per hour, a restore operation would take approximately 5.5 hours. 1 GB of offline redo log files can be restored in 10 minutes. It takes approximately three minutes to update the redo information to one single redo log file on the database. Therefore, it would take approximately 150 minutes to carry out a recovery with all offline redo log files from one day.

      To restore and recover the database would take up to eight hours and ten minutes. However, this time does not include the time to analyze and repair the error that led to the restore. Additionally, the time of the instance recovery that is performed at system startup is not accounted for in this calculation. Because of these time periods are not accounted for in the calculation, it is unlikely that a full restore and recovery can be performed in 8.5 hours.

      An uncommitted transaction must be rolled back during instance recovery. Therefore, the database needs more time to complete the recovery.

LESSON SUMMARY

You should now be able to:

- Create a backup strategy

Unit 2
Lesson 2

Reviewing Backup Tools

LESSON OVERVIEW

This lesson explains the tools BRBACKUP and BRARCHIVE, which are used to back up the database. The lesson also explains tape management with BR*Tools because BR*Tools offers functions for tape management that are similar to other backup utilities.

Business Example

After defining your backup strategy and providing the necessary number of tapes, you want to perform the backups. To do this, you must modify the BR*Tools parameter for backup and restore and initialize your tapes. For this reason, you require the following knowledge:

- An understanding of the different SAP tools for backup, restore, and recovery
- An understanding of the concept of Oracle's Recovery Manager (RMAN)
- An understanding of how to customize SAP tools
- An understanding of tape management with BR*Tools
- An understanding of how to initialize and manage backup tapes with BR*Tools

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Work with backup tools

Backup Tools

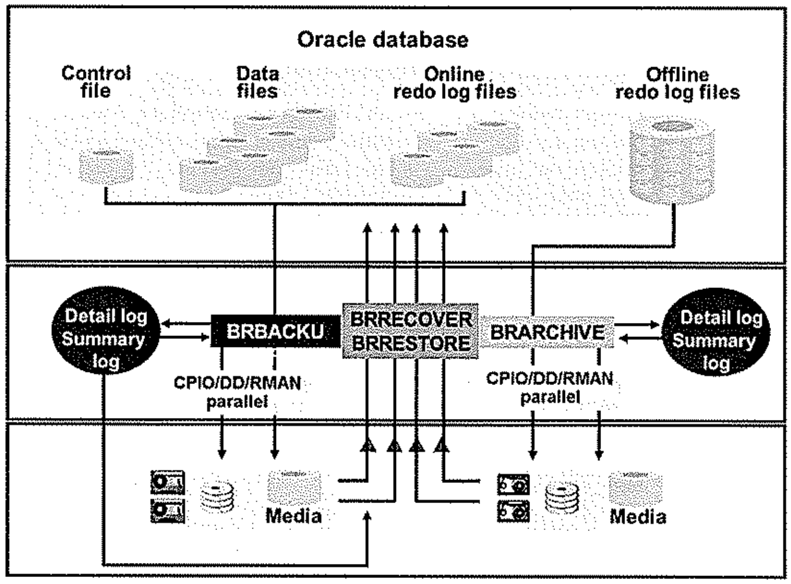


Figure 49: SAP Tools for Backup, Restore, and Recovery

The Oracle administration tools delivered by SAP include programs for backing up database files and other files of an SAP system and programs for restoring missing files and recovering data files to a consistent state.

The Oracle administration tools are as follows:

- BRBACKUP

  This program backs up Oracle data files, the control file, and the online redo log files, where necessary. BRBACKUP can also be used for backing up the Oracle software directories and the SAP system directories.

- BRARCHIVE

  This program backs up the offline redo log files of the database.

- BRRESTORE

  This program can restore all files belonging to the database system from the backups, both database files and offline redo log files.

- BRRECOVER

  This interactive program checks the database for missing files, calls BRRESTORE for restoration of missing data files or for restoration of offline redo log files needed for recovery, performs the recovery, and opens the database.

Both BRBACKUP and BRARCHIVE record the performed actions in log files. BRRESTORE analyzes and uses these log files to restore missing files.

BRBACKUP and BRARCHIVE support the following backups:

- Backups to tapes
- Backups to disks

- Backups with third-party tools

Customizing SAP Backup and Restore Tools

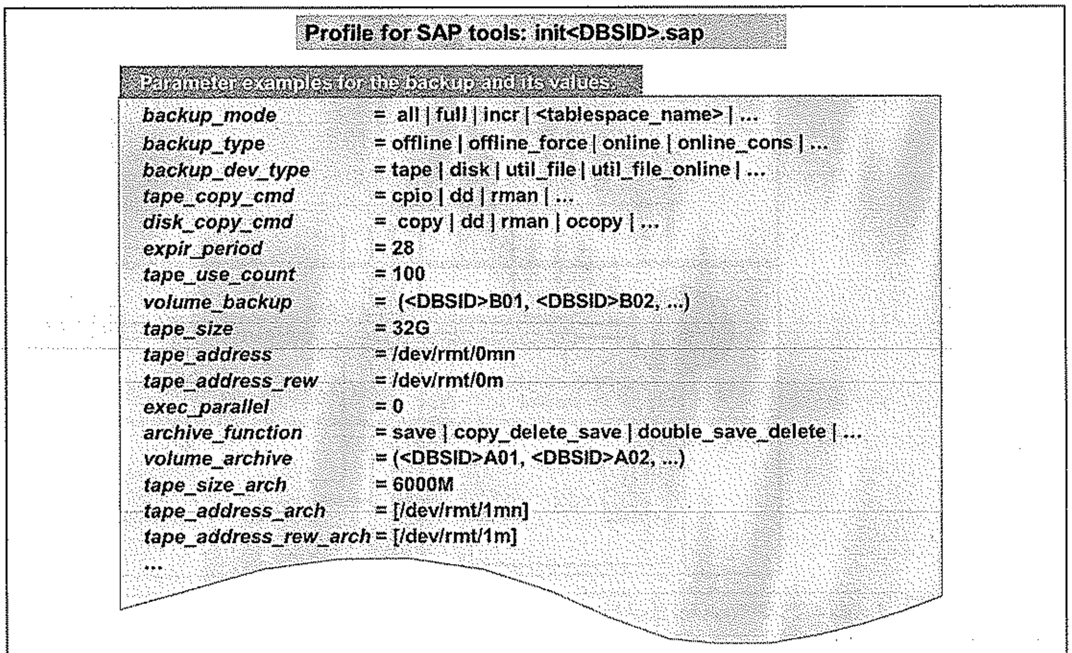


Figure 50: Profile of BR*Tools

The configuration for the initialization profile init<DBSID>.sap contains parameters that determine how BR*Tools (for example, BRBACKUP, BRARCHIVE, and BRRESTORE) perform various functions. init<DBSID>.sap is stored in the $ORACLE_HOME/dbs (UNIX) or %ORACLE_HOME%\database (Windows) directory.

To configure the behavior of SAP tools, you can edit the profile with a text editor. If you then start a tool without command options, the values in the initialization profile are used. If a parameter value is not specified in the profile, the SAP tool uses the default value for the parameter.

If you use BRBACKUP or BRARCHIVE with command options, these override the corresponding values in the initialization profile.

Parameters for Configuring BRBACKUP and BRARCHIVE

The important parameters for configuring BRBACKUP and BRARCHIVE are as follows:

- backup_mode

  This parameter determines the scope of the backup activity, that is, which part of the database or which directory will be backed up. The two types of complete database backups are determined through values all (whole backup) or full (full backup). Partial backup can be indicated, for example, using a tablespace name, file IDs, or a path to a directory. For incremental backups, use incr. The value ora_dir stands for Oracle software directory and the value sap_dir stands for the SAP system directory.

- backup_type

  This parameter helps you choose between online and offline backup.

- backup_dev_type

  This parameter specifies the backup medium you want to use, such as tape or disk, or it points out usage of an external backup program using the interface BACKINT (parameter value util_file or util_file_online).

- tape_copy_cmd

  This parameter contains the copy command used to copy files from a disk to a tape (cpio, dd, rman, and so on). This parameter does not affect raw devices, which are always copied with Data Description (DD), or directories, which are always copied with Copy In and Out (CPIO). The profiles init<DBSID>.ora and init<DBSID>.sap, and log files, such as summary log and detailed log, are always written with CPIO onto tape.

- disk_copy_cmd

  This parameter provides the copy command to be used to copy files to local disks. The copy value corresponds to the cp command on UNIX and copy on Windows.

> Hint:
> With BR*Tools 7.00, database and archive log files can be backed up to a local disk using the Oracle utility command OCOPY.

- expir_period and tape_use_count

  These parameters are used for tape management and specify the retention period and the recommended maximum number of times to which a volume can be written.

- volume_backup and volume_archive

  The parameter volume_backup provides names of volumes to be used for backups created with BRBACKUP. The parameter volume_archive provides names of volumes to be used for backups of offline redo log files created with BRARCHIVE. For each of these parameters, if you specify more than one volume, you must separate the names with commas and enclose the list in parentheses.

- tape_address*

  The parameter values tape_address and tape_address_rew specify the device addresses (device special files) of the tape drives that you want to use to backup the database (or the restore) with or without rewind. BRARCHIVE uses the same values if the optional parameters tape_address_arch and tape_address_rew_arch are not defined. You can enter more than one device address as a parameter value, which indicates that you are getting ready to perform a parallel backup to several devices, but the number of device addresses in a pair of rewind or no rewind parameters must always be the same. BRARCHIVE can use two devices at maximum. Note that values of all these parameters are operating system (OS) dependent. The given example is valid for Hewlett-Packard UniX (HP-UX).

Hint:
When you use the copy command, use DD instead of the default CPIO for copying data files to tapes because DD is faster and the required backup time can be reduced significantly. For best performance, specify a block size (for copying between disks and tapes) of at least 64 KB, using parameters `ddflags` and `dd_in_flags`. Contact your hardware manufacturer for recommendations for your tape device.

Note:
For a detailed description of all parameters for SAP tools, see the online help at *SAP Library → SAP Database Guide: Oracle*.

Integration of RMAN into SAP Tools

Oracle RMAN is the default Oracle backup and restore program. RMAN executes runs in a client process and connects to the database similar to SQL*Plus. By integrating RMAN into BRBACKUP and BRARCHIVE, SAP has added more flexibility to important backup strategies.

BRBACKUP supports RMAN for backing up database files in the following ways:

- RMAN is able to classify a complete database backup as a level 0 backup (full backup), which serves as a basis for level 1 backups (incremental backups).
- Data can be written to tapes (or other backup media) using RMAN instead of OS tools CPIO or DD.

For a complete backup, these two uses are independent of each other, meaning that each of the four combinations is possible. However, an incremental backup can be created only when RMAN is used as the program for writing data to backup media.

Caution:
If you use Oracle RMAN for writing data to the backup media, you must also use RMAN to restore and recover missing database files. RMAN is recognized and performed by BRRECOVER automatically. If BRRECOVER encounters an issue and cannot continue, the restore and recovery must be performed at the Oracle level and the user must be an expert in using RMAN.

Backups Without RMAN

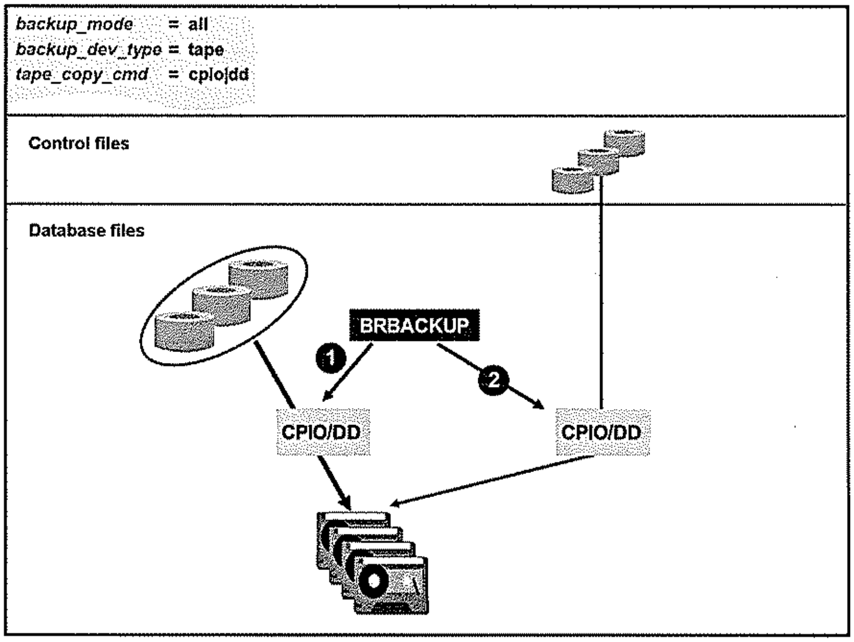


Figure 51: Backups Without RMAN

If backups are performed without Oracle RMAN, BRBACKUP and BRARCHIVE call CPIO or DD to save database files, and the control files, to the tape.

Full Backup with OS Tools and RMAN

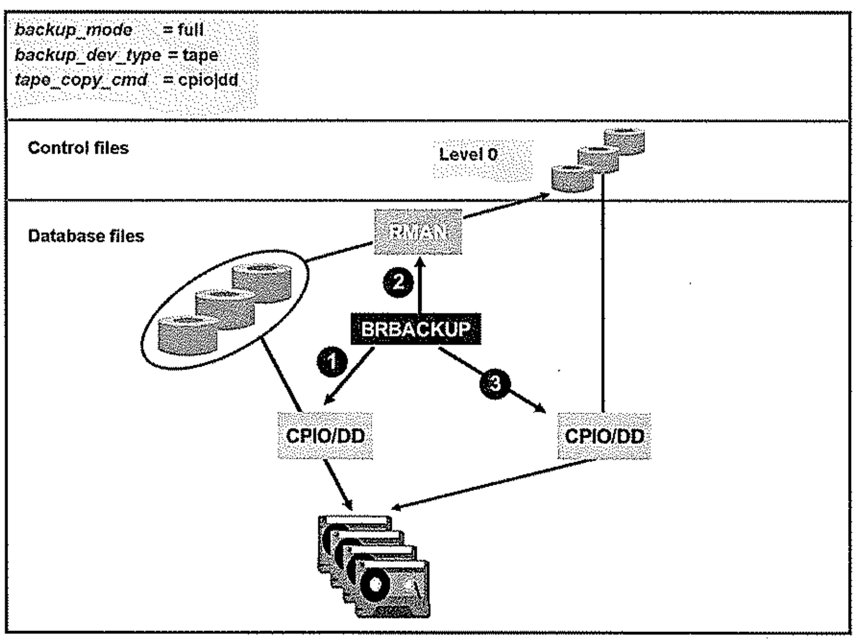


Figure 52: Full Backup with RMAN

To perform a full backup, set `backup_mode = full`. This causes RMAN (started by BRBACKUP) to write information about backed up data files to the control file and identify the backup as level 0 backup (full backup).

> Hint:
> When you perform a "native" backup with RMAN directly on Oracle level, RMAN can place information about backups in a separate database called the recovery catalog. The recovery catalog is crucial for security, because without the recovery catalog data, RMAN cannot recover the database automatically from previous backups. To avoid the need to secure the recovery catalog, the SAP implementation makes RMAN write the backup information only to the control file. To use RMAN directly is not part of SAP strategy for Oracle backups.

To back up files on backup media using BRBACKUP, set the parameter `tape_copy_cmd` to either CPIO or DD (when performing a backup to a disk, set the parameter `disk_copy_cmd` correspondingly). Data files are then saved to a tape with the command specified. At the end of the backup process, BRBACKUP starts RMAN to write the backup information to the control file. Finally, the control file is written to a tape with the same OS tool as before (specified in `tape_copy_cmd`).

Full Backup with RMAN and System Backup to Tape (SBT) Library

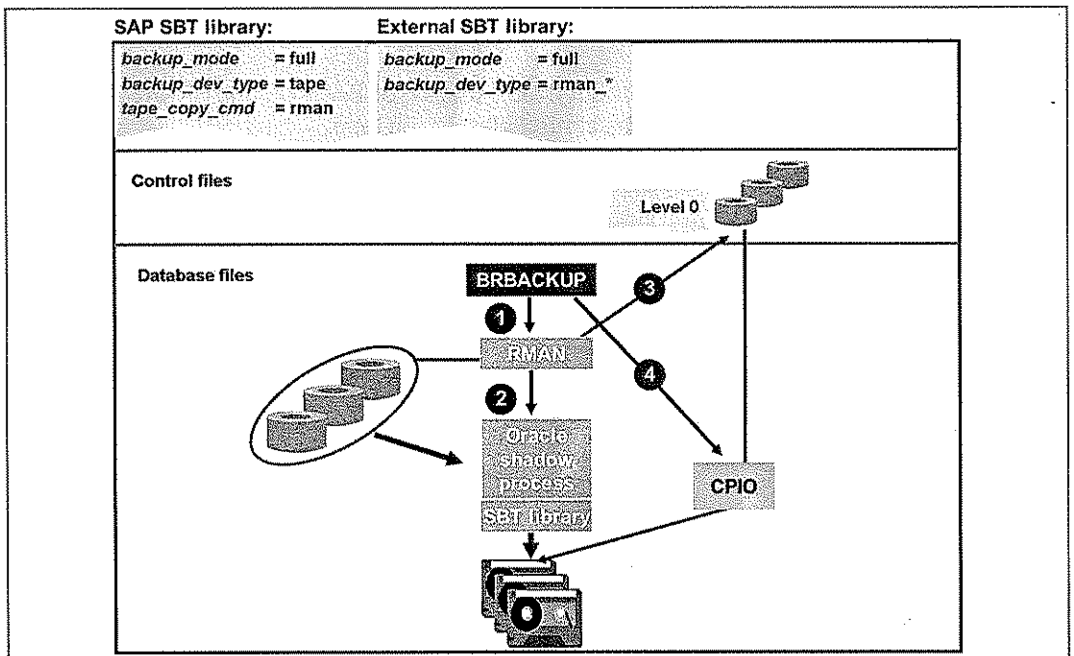


Figure 53: Full Backup with RMAN and the SBT Library

To leave the control of copying data to backup media to RMAN, set `tape_copy_cmd` (or `disk_copy_cmd`) to the `rman` value. In this case, RMAN is already started by BRBACKUP at the beginning of the backup process and takes care of backing up the data files. Through its Oracle shadow process, RMAN reads data blocks from the database, checks them for corruption, and filters out those blocks that have never been used (those still in the initial status). Used blocks are then written by the shadow process to the backup medium.

In the last phase of the backup process, if `backup_mode` was set to full, RMAN writes the backup information to the control file and CPIO copies the control file to the tape.

RMAN can perform backups directly to a disk but not directly to a tape. For backups to a tape, RMAN uses the SBT interface provided by Oracle, for which manufacturers of external backup utilities have to provide a library. The SBT library allows data to be backed up to a tape directly.

Before performing backups using RMAN, you must install the corresponding backup library in the following situations:

- When you do not use an external backup utility, you must install the SBT library provided by SAP. The SAP SBT library is automatically copied to the directory `/usr/sap/<SAPSID>/SYS/exe/run` during the installation of an SAP system. However, it must be made available in the directory `$ORACLE_HOME/bin`. To use the SAP SBT library, set `backup_dev_type` to `tape`, `tape_auto`, or `tape_box` and `tape_copy_cmd = rman` after installing the library.

- When you use an external backup utility, you must install the SBT library of the external backup tool. For information on how to get the SBT library for your external backup, contact the vendor of the external backup tool. Oracle provides a limited single-server version of Legato Networker, including the Legato SBT library, on the first Oracle installation CD. To use the SBT library of an external backup tool, set `backup_dev_type` to `rman_disk` or `rman_util`. The `tape_copy_cmd` parameter is ignored in this case.

> Hint:
> Setting `tape_copy_cmd = rman_disk` or `rman_stage` is possible as of SAP Web AS 6.20. This setting enables you to perform backups with BR*Tools using RMAN and an external SBT library without having to use the BACKINT interface of the external backup tool.

> Note:
> For more information about installation of the SBT library, see SAP Note 142635.

Advantages of Using RMAN to Copy Data to Backup Media

Using RMAN to copy data to backup media has the following advantages:

- All blocks are checked for block corruption. This ensures that each successful backup contains the database in a consistent state; so an extra verification of the database becomes unnecessary.

- Only used blocks are copied to the backup media. This can reduce the amount of data to be backed up. However, blocks that are empty but have been used before (blocks from dropped tables) are always backed up.

- In a standard online backup, tablespaces are set to backup mode to deal with possible inconsistencies within data blocks. Such an inconsistency can occur, for example, when CPIO or DD performs a copy of an 8 kB Oracle block in smaller OS units while the block is being overwritten by the database writer process. When a tablespace is in backup mode, whole dirty blocks are copied from the buffer pool to the current redo log instead of writing only modified records there, which can drastically increase the amount of redo log entries.

With Oracle RMAN, this is not necessary because the blocks are checked to see whether the data is consistent. RMAN compares the checksum before and after the copy of a block read from the disk because each block contains a checksum. If these two checksums are not equal, RMAN reads the block once more. Consequently, much less redo log information is written during an online backup with RMAN, compared to a standard online backup.

> Hint:
> As of BR*Tools 7.00, RMAN binary compression can be activated by setting the parameter `rman_compress = no|yes` in `init<DBSID>`.

A whole or partial backup with RMAN (`tape_copy_cmd = rman` or `backup_mode = <object_list>`, `backup_mode = all` or `disk_copy_cmd = rman`) is possible. All mentioned advantages apply in this case as well. Obviously, a whole backup is not a level 0 backup and cannot be used as a basis for incremental backups.

> Hint:
> As of SAP Web AS 6.10, BRARCHIVE also supports RMAN backups of offline redo log files with the SAP backup library. The advantage of this process is that the data in the offline redo log files is checked for internal consistency during the RMAN backup. This verification functionality for offline redo log files was missing in older releases.

Save Sets

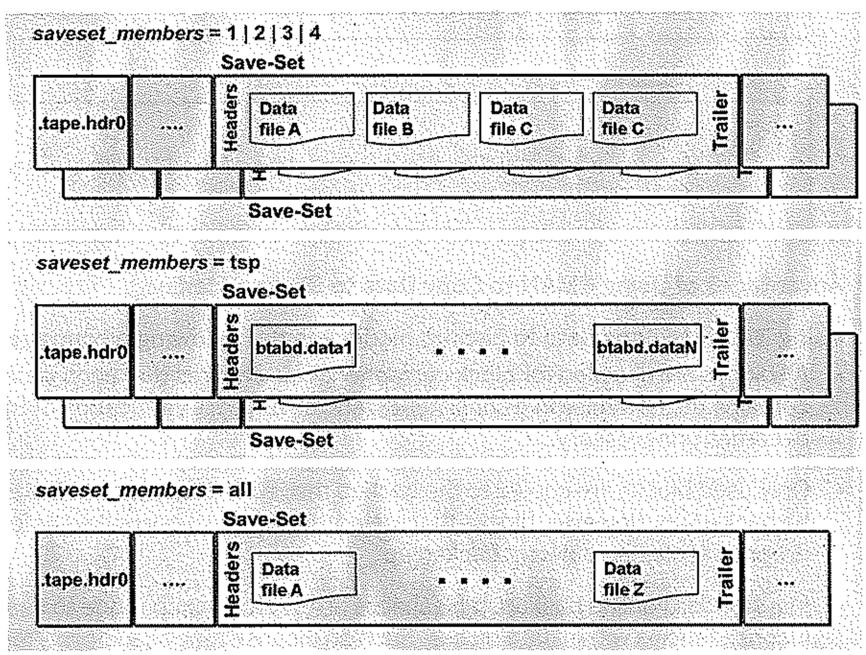


Figure 54: Save Sets

When you back up an Oracle database with RMAN, the SAP backup library helps optimize the utilization of fast tape drives by combining multiple data files in save sets. If several data files belong to a save set, RMAN reads the data in parallel from the files. This multiple file access, also known as file multiplexing, maximizes the flow of data to keep tape drives in streaming mode. Through a higher output to tape stations, the time required for a backup can be reduced.

A save set consists of a header, a trailer, and the blocks of at least one data file. Each save set is treated as an indivisible unit, which must always be stored on a single tape.

In `init<SAPSID>.sap`, the parameter `saveset_members` determines the maximum number of files in a save set.

The following table represents the possible value of the parameter and their meanings:

Value of the parameter	Meaning
`1, 2, 3, or 4`	The value indicates the number of files to be grouped together to form one save set (the default is 1).
`tsp`	One save set is formed for each tablespace that is to be backed up. The save set contains the data of all data files belonging to a tablespace (as long as they fit on one tape).
`all`	Only one save set containing all data files of the database is created if the tape used is large enough for it.

Caution:
Using large save sets can speed up the backup process, but it has the disadvantage that the restore and recovery time can increase. When only recovery of one (damaged) data file is needed, the complete save set containing this file must be read from the tape. You must, therefore, determine the minimum save set size for your system that guarantees a reasonably fast output to tape devices during backup.

With backups to disks performed with RMAN (`backup_dev_type = disk`, `disk_copy_cmd = rman`), no save sets are formed. Data files are directly copied to disks, similar to when CP or DD is used. Save sets are created only when an SBT backup library is used for incremental backups.

Preparation Run

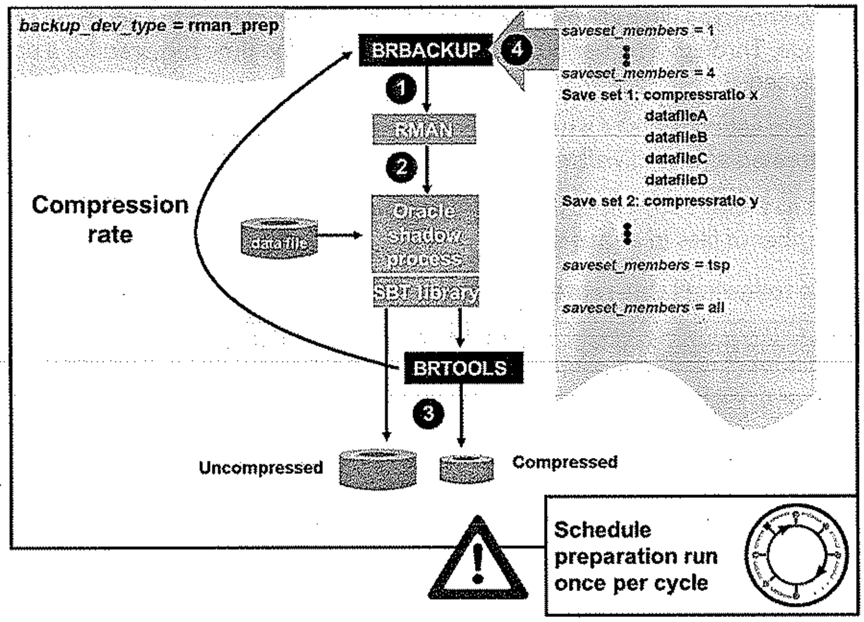


Figure 55: Preparation Run

You must perform a trial run to determine the optimal save set distribution of the data files that you want to back up in either of the following cases:

- If a backup with RMAN is supposed to form save sets with more than one member (because you set `saveset_members` to a value different than 1)
- If you use tape stations with hardware compression and you want RMAN to take a compression rate into account when creating save sets with appropriate sizes to match the tape size

In the preparation run, which can be started in the DBA Cockpit or using transaction `DB13`, action *Prepare for RMAN Backup*, BRBACKUP starts an RMAN backup of every data file to a save set of its own. No backup is created during this run. The SAP backup library estimates the compression rate of the save set by letting BRTOOLS compress the file and determine the decompressed and compressed file sizes. The expected compression rate of the save set with one member is then sent to BRBACKUP.

At this point, BRBACKUP determines how data files are allocated to save sets for every possible value of `saveset_members` and calculates the compression rate of each save set. The information on the composition of the save sets and the compression rates is stored in the database and is used during future backup runs.

The allocation of files to save sets cannot be controlled manually. You can only change it, if necessary, by carrying out a new preparation run. Between two preparation runs, save sets that correspond to a specific value of `saveset_members` remain unchanged and contain the same files.

If, during a backup, RMAN finds new data files that were not included in the last preparation run (for example, because a data file was added), each of these files is put in its own save set.

SAP recommends that you perform a preparation run once per backup cycle and after major database changes, for example, after adding a file to a tablespace, after reorganization, after mass data transfer, or after an SAP or database release upgrade.

Incremental Backup

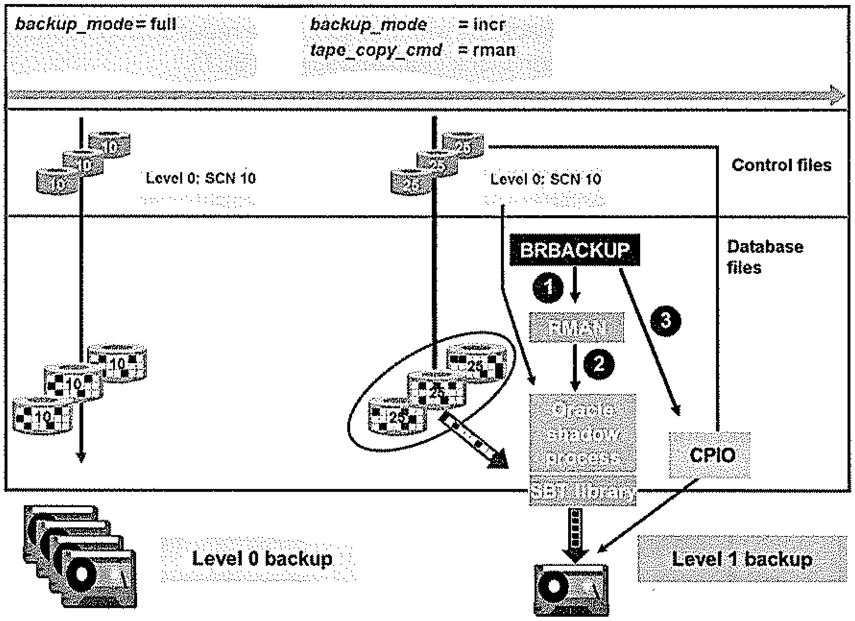


Figure 56: Incremental Backup

An incremental backup, specified with the parameter `backup_mode = incr`, is also known as a level 1 backup. It is always based on the last level 0 backup (full backup). RMAN reads information about the last level 0 backup from the control files.

An incremental backup is always a backup of the whole database, not of individual data files. OS tools cannot be used for writing data to backup media. So for an incremental backup, parameter settings of `tape_copy_cmd` or disk_copy_cmd are ignored and implicitly set to `rman`.

After the incremental backup is complete, a control file is saved to a tape by CPIO.

In an incremental backup, all blocks of all data files are always read. However, only those blocks that have changed since the last level 0 backup are backed up. An incremental backup can, therefore, reduce the backup time if the tape stations have fewer throughputs.

With SAP tools, only a cumulative level 1 backup is supported as an incremental backup. This means that an incremental backup includes all those blocks that have been already saved during a previous incremental backup (based on the same full backup).

Only one save set (with the extension `.INCR`) is created for an incremental backup. The parameter `saveset_members` is internally set to `all` for an incremental backup run. Because only one save set is created, the backup must fit on one tape. Follow-up tapes may be used.

If data files are added between the last level 0 backup and the level 1 backup, a level 0 backup is performed for these files before the start of the actual level 1 backup. All new data is backed up to one separate save set, which always gets the extension `.FULL`, even if it contains only a part of the database.

Fast Incremental RMAN Backups

Up to and including Oracle 9i, RMAN must read all blocks of a file during incremental backup, even unchanged files, to determine which blocks have been changed since the last full backup. An incremental backup (level 1), therefore, lasts a similar length of time as a full backup (level 0).

The time for incremental backups with RMAN can be reduced considerably as of Oracle 10g, with the new RMAN function block change tracking (BCT).

If BCT is activated, Oracle keeps a log in the block change tracking file (BCTF) of which data blocks were changed since the last level 0 RMAN backup. The information from this BCTF is then used by RMAN during the incremental backup to read and back up only the changed blocks. There is one BCTF for each database. The BCTF is written by the new Oracle background process change tracking writer (CTWR).

The duration for an incremental backup is approximately proportional to the size of the database if BCT is deactivated, but proportional to the number of changed blocks if BCT is activated. If only small parts of the blocks are changed between two level 0 backups, the time saved as a result of BCT is considerable.

BCT Functioning

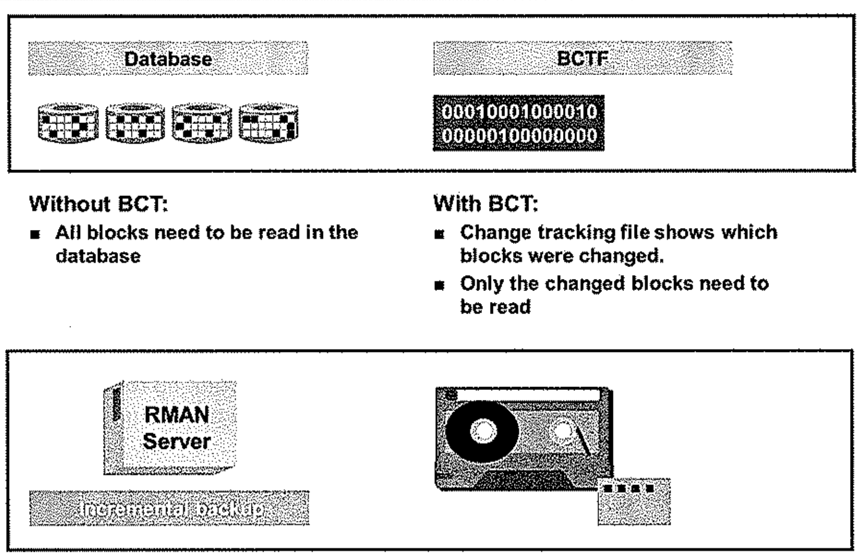


Figure 57: BCT Functioning

By default, BCT is deactivated. When BCT is activated for the first time, RMAN must read all blocks per file during the first level 0 backup, because the BCTF does not yet reflect the true, current block status the first time it is created. RMAN can then use the BCTF information in any subsequent incremental backups.

> Caution:
> In the SAP environment, BCT is supported as of BR*Tools 7.00 for Oracle databases as of Oracle 10.2.0.2. The BR*Tools configuration does not need to be changed for BCT. The BCTF is not backed up, created, or managed by BR*Tools.

The BCTF has the SAP standard name bctf<DBSID>.ora and is saved in the following directories:

- `$ORACLE_HOME/dbs` (Unix)
- `%ORACLE_HOME%\DATABASE` (Windows)

BCT can be activated and deactivated using the following SQLplus commands:

- Activate BCT

```
ALTER DATABASE ENABLE BLOCK CHANGE TRACKING USING FILE '<Filename>'
REUSE
```

- Deactivate BCT

```
ALTER DATABASE DISABLE BLOCK CHANGE TRACKING
```

> Note:
> For more information, see SAP Note 964619.

Recovery with Incremental Backup

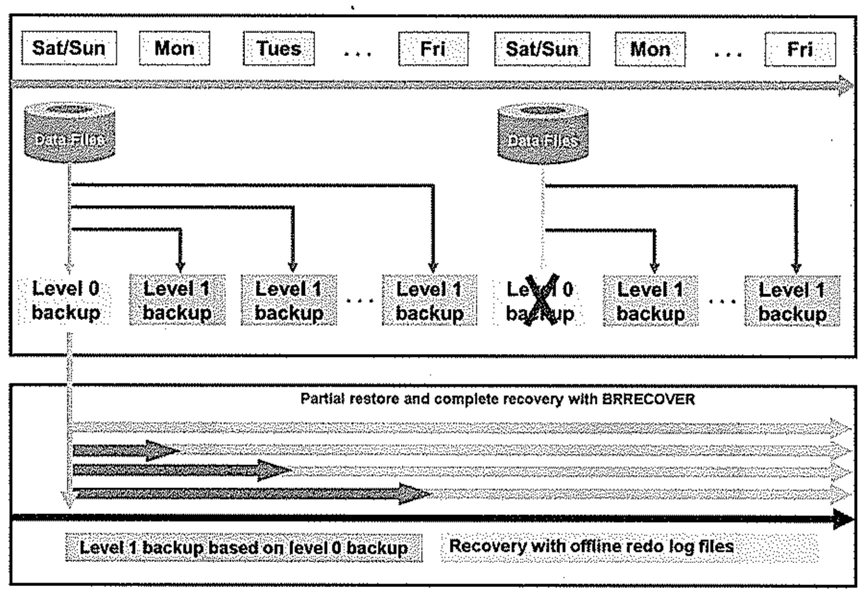


Figure 58: Recovery with Incremental Backup

When a restore or recovery is to be carried out (for example, due to a disk crash), a level 1 backup is not sufficient to repair the database. A level 0 backup of the lost files is always required.

The following steps are performed when a restore or recovery is carried out:

1. The lost files are recovered from a level 0 backup.
2. The changed blocks from a level 1 backup, which must be based on the applied level 0 backup, can be imported to the data file.

3. A recovery is performed from the time of the level 1 backup.

You only need to apply one incremental backup, preferably the latest one, because incremental backups created with SAP tools are cumulative.

If no level 1 backup is available for the level 0 backup, perform the recovery based on the last available level 0 backup. This usually takes longer than a recovery that uses the level 1 backup.

If you cannot use the last level 0 backup, you must use the previous level 0 backup for restoring the data file. In the second step, only a level 1 backup can then be used because the second step is based on this level 0 backup. Incremental backups based on the damaged full backup cannot be used.

> Caution:
> Perform at least two, preferably four, level 0 backups within one backup cycle. An incremental backup without its corresponding level 0 backup is not useful.

External Backup Tools

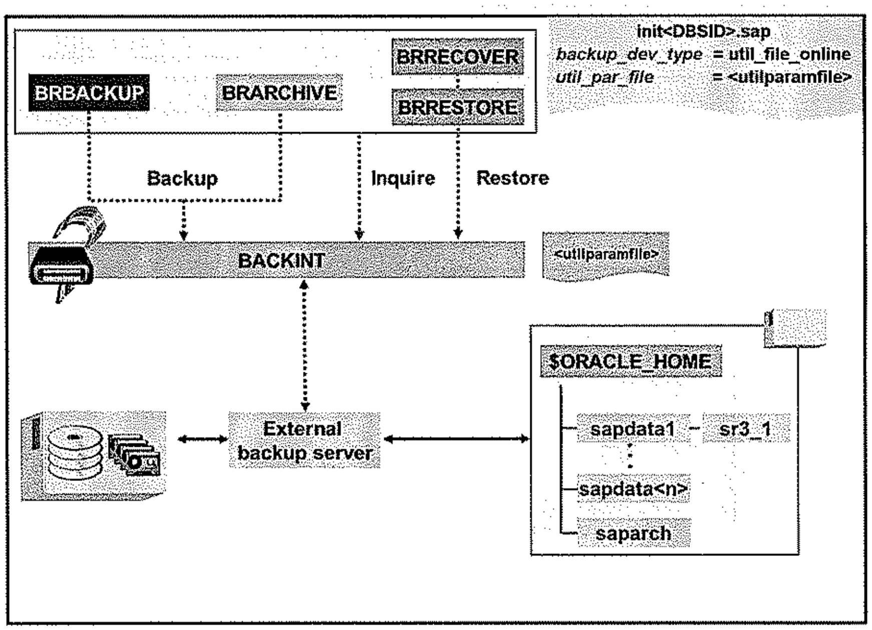


Figure 59: External Backup Tools

The SAP tools BRBACKUP, BRARCHIVE, BRRESTORE, and BRRECOVER provide an interface called BACKINT that can be used to access external backup programs. You can use this interface if the BACKINT is supported by the supplier of the external backup program.

External backup programs provide the following advantages:

- You can use new, manufacturer-specific backup media, such as tape robots and magneto-optical media. SAP tools, for example, do not support a direct backup to or restore from optical storage media. However, you can use such media with an external backup program and the BACKINT interface.

- You can set up a consistent backup procedure for file systems and databases.
- The client-server backup configuration allows for the use of one backup server (including mainframe).

The backups must still be started by SAP tools. This ensures that all actions are logged and that backups can be monitored using Computing Center Management System (CCMS). In addition, starting backups with SAP tools allows you to use the restore and recovery features of BRRECOVER.

The system performs the following backup and restore tasks:

- The system uses BRBACKUP, BRARCHIVE, BRRESTORE, and BRRECOVER to backup and restore media.
- The external backup program manages the backup media.
- BRBACKUP or BRARCHIVE uses BACKINT to pass a backup request to the external backup program. This request contains a list of the files for backup. BRRESTORE also uses BACKINT to trigger the external program to restore the requested files. BRRECOVER calls BACKINT while performing disaster recovery.
- The external backup program performs all the backup operations.
- BRBACKUP, BRARCHIVE, BRRESTORE, or BRRECOVER evaluates the confirmation messages of the external backup program.

To configure the BACKINT interface, set parameter `backup_dev_type` to `util_file` or `util_file_online` in the `init<DBSID>.sap` file (if using the second parameter, the tablespace to be backed up is set to the backup mode dynamically, upon request from BACKINT). The value of the parameter `util_par_file` must refer to the configuration file that contains parameters for the external backup utility. For the name and location of the parameter `util_par_file`, refer to the documentation of your external backup utility.

To use external backup tools in combination with RMAN backups, set `backup_dev_type` to `rman_util`, `rman_disk`, or `rman_stage`.

> Note:
> For more information, see SAP Note 142635.

> Hint:
> Setting `backup_dev_type` to any `util_file*` or `rman_*` values turns off tape management performed by SAP tools. This means that the `tape_copy_cmd` parameter is ignored.

> Note:
> You can use the BACKINT solution only with certified external backup tools. For more information about SAP partners that support the interface to external backup programs, see http://service.sap.com/partners.

Unit 2
Exercise 6

Use Backup Tools

Business Example

As a prerequisite of RMAN backups, determine the optimal save set distribution.

Start an RMAN preparation run to determine the optimal save set distribution of the data files. Also, determine the compression rate of backups performed without RMAN.

1. Determine the optimal save set distribution of backups performed with RMAN.
2. Why is the option *BRBACKUP run type (type)* set to *offline*, in the *BRBACKUP options for RMAN preparation* menu?
3. Start the preparation run in online mode.
4. Check the results of the RMAN preparation run in the log file.
5. Determine the compression rate of backups performed without RMAN.

Unit 2
Solution 6

Use Backup Tools

Business Example

As a prerequisite of RMAN backups, determine the optimal save set distribution.

Start an RMAN preparation run to determine the optimal save set distribution of the data files. Also, determine the compression rate of backups performed without RMAN.

1. Determine the optimal save set distribution of backups performed with RMAN.
   a) Start BRGUI or BRTOOLS and choose *Backup and database copy → Additional functions → Preparation of RMAN backups*.

2. Why is the option *BRBACKUP run type (type)* set to *offline*, in the *BRBACKUP options for RMAN preparation* menu?
   a) The menu options in the BR*Tools are filled according to the settings in the parameter file `init<DBSID>.sap`.

3. Start the preparation run in online mode.
   a) Enter **`online`** in the parameter *BRBACKUP run type (type)*.

   The following list of options appears:

```
BR0657I Input menu 27 - please enter/check input values
-----------------------------------------------------------------
--------------
BRBACKUP options for RMAN preparation

 1 - BRBACKUP profile (profile) ...... [initT99.sap]
 2 - Database user/password (user) ... [/]
 3 - BRBACKUP run type (type) ........ [online]
 4 ~ Files for preparation (mode) .... [all]
 5 - Confirmation mode (confirm) ..... [yes]
 6 - Query mode (query) ............. [no]
 7 - Parallel execution (execute) .... [0]
 8 - Additional output (output) ...... [no]
 9 - Message language (language) ..... [E]
10 - BRBACKUP command line (command) . [-p initT99.sap -d
rman_prep -t online -m
 all -e 0 -l E]

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-----------------------------------------------------------------
--------------
BR0662I Enter your choice:
```

   b) Choose *Continue*.

4. Check the results of the RMAN preparation run in the log file.
   a) The log file can be displayed using BRGUI or BRTOOLS. In the main menu, choose *Additional functions → Show profiles and logs → BRBACKUP logs*.

b) In the menu *Display of BRBACKUP logs*, choose the log file with the extension **.rmp*.

c) Choose *Continue*.

The optimal save set distribution of the data files for the each possible setting of the parameter `saveset_members` is displayed at the end of the log file. The following information appears:

```
BR0001I **************************************************

BR0527I Save sets with 1 file:

Saveset          Size          Rate        Compressed    Name

    1       174522368       2.3059:1        75686184     ...
\SYSTEM.DATA1
    2       111280128       3.7769:1        29462982     ...
\SYSAUX.DATA1
    3        21037056       2.5050:1         8398152     ...
\UNDO.DATA1
    4         1114112     207.4310:1            5371     ...
\T99.DATA1
    5         3145728       8.1145:1          387669     ...
\T99USR.DATA1

BR0527I Save sets with 2 files:

Saveset          Size          Rate        Compressed    Name

    1       285802496       2.7181:1       105149166     ...
\SYSAUX.DATA1
                                                         ...
\SYSTEM.DATA1
    2        22151168       2.6359:1         8403523     ...
\T99.DATA1
                                                         ...
\UNDO.DATA1
    3         3145728       8.1145:1          387669     ...
\T99USR.DATA1

BR0527I Save sets with 3 files:

Saveset          Size          Rate        Compressed    Name

    1       286916608       2.7285:1       105154537     ...
\T99.DATA1
                                                         ...
\SYSAUX.DATA1
                                                         ...
\SYSTEM.DATA1
    2        24182784       2.7525:1         8785821     ...
\T99USR.DATA1
                                                         ...
\UNDO.DATA1

BR0527I Save sets with 4 files:

Saveset          Size          Rate        Compressed    Name

    1       307953664       2.7120:1       113552689     ...
\T99.DATA1
                                                         ...
\UNDO.DATA1
```

```
                                                          ...
\SYSAUX.DATA1
                                                          ...
\SYSTEM.DATA1
      2        3145728        8.1145:1        387669      ...
\T99USR.DATA1

BR0527I Save sets with 'tsp' files:

Saveset          Size          Rate       Compressed      Name

      1        1114112      207.4310:1         5371       ...
\T99.DATA1
      2        3145728        8.1145:1        387669      ...
\T99USR.DATA1
      3       21037056        2.5050:1       8398152      ...
\UNDO.DATA1
      4      111280128        3.7769:1      29462982      ...
\SYSAUX.DATA1
      5      174522368        2.3059:1      75686184      ...
\SYSTEM.DATA1

BR0527I Save sets with 'all' files:

Saveset          Size          Rate       Compressed      Name

      1      311099392        2.7304:1     113940358      ...
\T99.DATA1
                                                          ...
\T99USR.DATA1
                                                          ...
\UNDO.DATA1
                                                          ...
\SYSAUX.DATA1
                                                          ...
\SYSTEM.DATA1

BR0280I BRBACKUP time stamp: 2011-06-01 09.47.05
BR0533I Uncataloging save sets created by RMAN...

BR0522I 5 of 5 files / save sets processed by RMAN

BR0280I BRBACKUP time stamp: 2011-06-01 09.47.09
BR0534I Save sets created by RMAN uncataloged successfully

BR0056I End of database backup: befzxxgk.rmp 2011-06-01 09.47.09
BR0280I BRBACKUP time stamp: 2011-06-01 09.47.10
BR0052I BRBACKUP completed successfully
```

d) Choose *Skip* to leave the *Display of file content* menu.

5. Determine the compression rate of backups performed without RMAN.

   a) Start BRGUI or BRTOOLS and choose *Backup and database copy → Additional functions → Update of compression rates*.

   The following list of options appears:

```
BR0657I Input menu 26 - please enter/check input values
-----------------------------------------------------------------
--------------
BRBACKUP options for determination of compression rates

 1 - BRBACKUP profile (profile) ...... [initT99.sap]
```



```
 2 - Database user/password (user) ... [/]
 3 - BRBACKUP run type (type) ........ [online]
 4 ~ Files for compression (mode) .... [all]
 5 - Confirmation mode (confirm) ..... [yes]
 6 - Query mode (query) ............. [no]
 7 - Parallel execution (execute) .... [0]
 8 - Additional output (output) ...... [no]
 9 - Message language (language) ..... [E]
10 - BRBACKUP command line (command) . [-p initT99.sap -k only -
t online -m all
-e 0 -l E]

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-----------------------------------------------------------------
--------------
BR0662I Enter your choice:
```

The compression rate is displayed at the end of the output as follows:

```
BR0280I BRBACKUP time stamp: 2011-06-01 09.21.04
BR0063I 7 of 7 files processed - 680.969 of 680.969 MB done
BR0204I Percentage done: 100.00%, estimated end time: 9:21
BR0001I **************************************************

BR0115I Compression rate for all files 6.0804:1

BR0056I End of database backup: befzxuzq.cmb 2011-06-01 09.21.05
BR0280I BRBACKUP time stamp: 2011-06-01 09.21.06
BR0052I BRBACKUP completed successfully

#################################################################
##############
```

Tape Management with BR*Tools

To facilitate the management of tapes used to backup your Oracle database, BRBACKUP and BRARCHIVE offer a tape management system.

The tape management system has the following functions:

- Helps you find and correctly use the tapes necessary to perform a backup
- Helps you find the appropriate tapes when you need to restore and recover your database
- Provides for tape protection so that tapes are not accidentally overwritten

Tape Pools

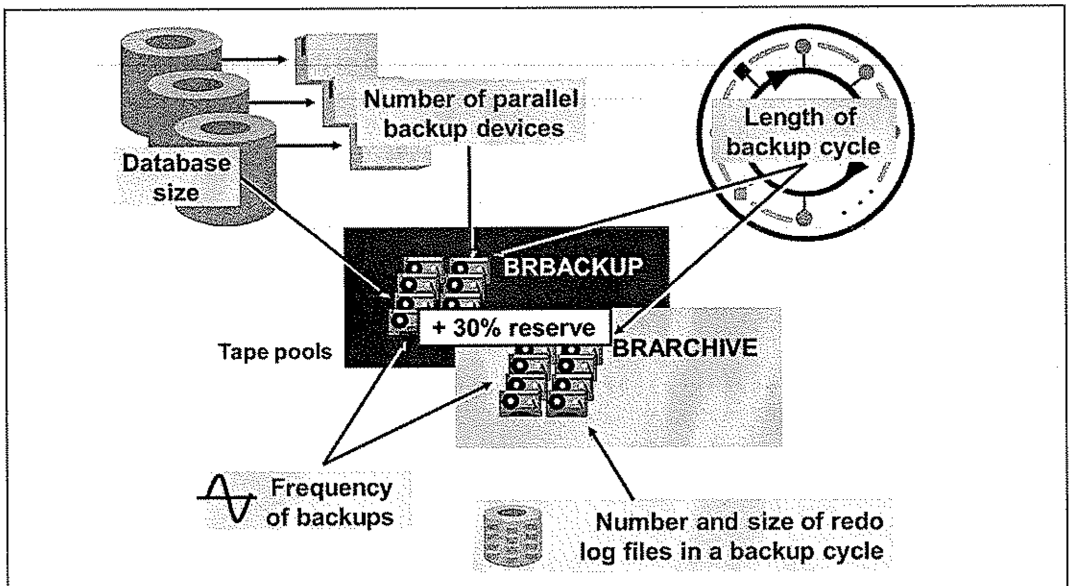


Figure 60: Tape Pools

For Oracle administration in an SAP system, the following tools are used for backups:

- BRBACKUP

  This tool is used for database backups.

- BRARCHIVE

  This tool is used for offline redo log file backups.

A separate pool of tapes is required for database backups, and another one for offline redo log file backups. You must ensure that enough tapes are provided in each tape pool to span the entire backup cycle. Backup tapes from each pool can be reused at the end of the backup cycle.

> Hint:
> If backups performed with BRBACKUP and BRARCHIVE are written directly to a tape, at least one tape is needed for each run of any of these tools. You cannot save the backup of two backup runs to the same tape.

The number of tapes you need for database backups is the product of the following factors:

- Number of tapes needed for one complete database backup

  This number depends on the size of your database and on the capacity of the tapes you use.

- Number of backups per cycle

  This number depends on the length of your backup cycle and the frequency of database backup operations.

The number of tapes you need for backing up offline redo logs depends on the following factors:

- The average number and the size of the redo log files created in a backup cycle (which, in turn, depend on the length of your backup cycle and on the activities in your database)

- The storage capacity of the tapes you use

The number of tapes for backing up offline redo log files can also be influenced by the frequency of these backup operations, in relation to database activities. If the redo information created between two offline redo log backups does not fill one tape, you need more tapes for the cycle.

> Hint:
> The default actions of the `DB13` templates within the DBA Cockpit *Whole database offline + redo log backup* and *Whole database online + redo log backup* back up the data files and redo log files offline in one run. This strategy saves tapes because no extra tape pool for BRARCHIVE is necessary (assuming that data files and offline redo logs fit on one tape).

In addition to the number of tapes you need, based on your backup strategy, you must have a reserve of 30% more tapes than required in each tape pool. This reserve is useful in the case of database growth, exceptionally high redo log volume caused by additional activities in the database, or if additional backups need to be performed.

Initialization of Tapes

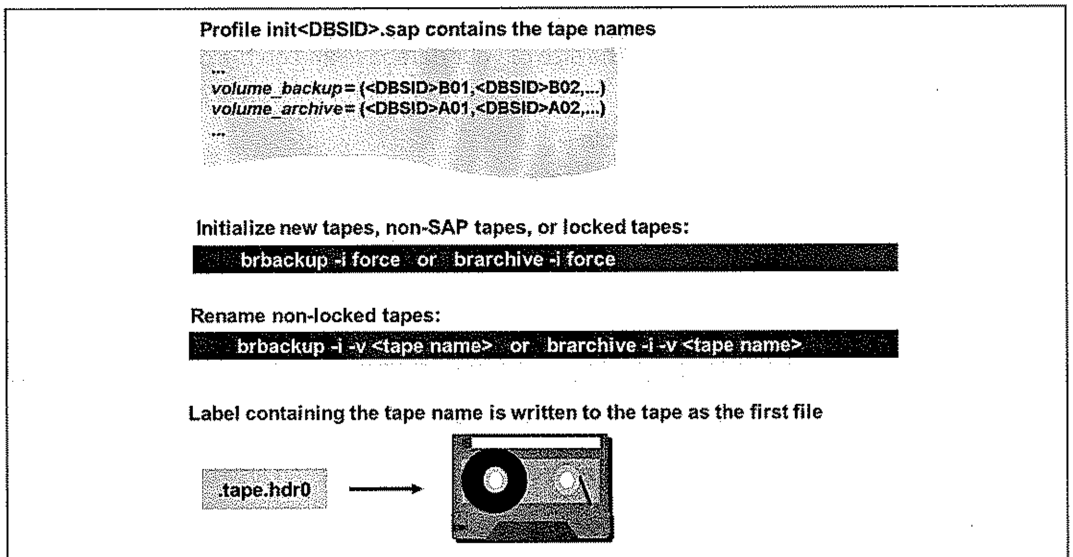


Figure 61: Initializing Tapes

A prerequisite for using the tape management system is two pools of initialized tapes. You must initialize one pool of tapes for BRBACKUP and another one for BRARCHIVE. Tapes that are initialized by BRBACKUP must not be used by BRARCHIVE, and tapes that are initialized by BRARCHIVE must not be used by BRBACKUP. Tapes that are not initialized at all are rejected.

During tape initialization, an SAP-specific label is written on the tape as the first file (`.tape.hdr0`), containing the tape name (volume name). You can specify the tape name explicitly, or BRBACKUP or BRARCHIVE will automatically select the tape names from the pool of names defined in the configuration file `init<DBSID>.sap` by parameters `volume_backup` and `volume_archive`.

The following naming conventions are recommended for your tapes:

- `<DBSID>B01, <DBSID>B02, ..., <DBSID>Bxx` for BRBACKUP
- `<DBSID>A01, <DBSID>A02, ... , <DBSID>Axx` for BRARCHIVE

To initialize tapes, start BRTOOLS or BRGUI and choose *Backup and database copy → Additional functions → Initialization of BRBACKUP tape volumes or Initialization of BRARCHIVE tape volumes*. Alternatively, you can start BRBACKUP or BRARCHIVE with the option `-i` | `-initialize` and specify all necessary options on the command line.

When you start the initialization of tapes using BRTOOLS, the following menu appears:

```
BR0657I
Input menu 32 - please check/enter input values
-----------------------------------------------------------------------
--------
Options for initialization of BRBACKUP tape volumes

 1 - BRBACKUP profile (profile) ....... [initT99.sap]
 2 - Initialization type (initialize) . [rename]
 3 - Number of volumes (number) ....... []
 4 - Confirmation mode (confirm) ...... [yes]
 5 - Message language (language) ...... [E]
```



```
 6 ~ Tape volume names (volume) ....... [T99B01]
 7 - BRBACKUP command line (command) .. [-p initT99.sap -i -l E -v
T99B01]

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-----------------------------------------------------------------------
--------
BR0662I Enter your choice:
```

The options for initialization of tapes are as follows:

- `initialize`

  The default initialization type is `rename`. Only those tapes can be renamed that were previously initialized and whose retention period has expired. In all other cases, initialization must be performed with initialization type `force`. To avoid overwriting a used tape with the `force` type, first check the label using initialization type `show`.

- `number`

  If empty, all not-yet-initialized tapes are initialized. Set `number` to initialize only a certain number of tapes.

- `volume`

  If empty, the tapes specified in the `init<DBSID>.sap` parameter `volume_backup` and `volume_archive`, respectively, are initialized. To initialize specific tapes, enter their volume names here, separated by commas.

Tape Label Contents

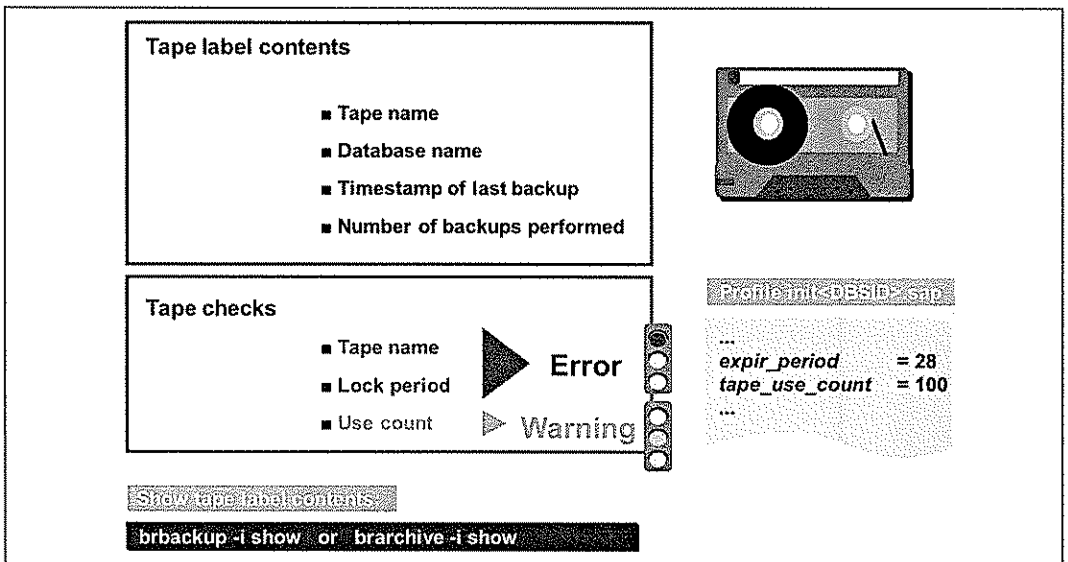


Figure 62: Tape Label Contents

After an initialized tape has been used by BRBACKUP or BRARCHIVE, the tape label contains the following information:

- The name of the tape
- The name of the database for which the backup was performed
- The timestamp of the last backup recorded on the tape

- The number of backups performed with this tape

By default, BRBACKUP and BRARCHIVE read the tape label before they start writing to the tape to check the following:

- The tape name
- Whether the tape is locked, that is, the configured expiration period has not ended yet. This period refers to the number of days specified in the parameter `expir_period` in the file `init<DBSID>.sap` that must have passed before the volume can be used again
- The number of times the tape has already been used

If the tape name is wrong or if the tape is locked, an error is reported and the tape is not used. If the tape is used more often than the value set in the parameter `tape_use_count` in the file `init<DBSID>.sap`, a warning is generated but the tape is used.

The expiration period always expires at midnight of the last day of the lock. If you set an expiration period of zero days, the volume is not locked at all and it can be overwritten on the same day.

Tape Checks

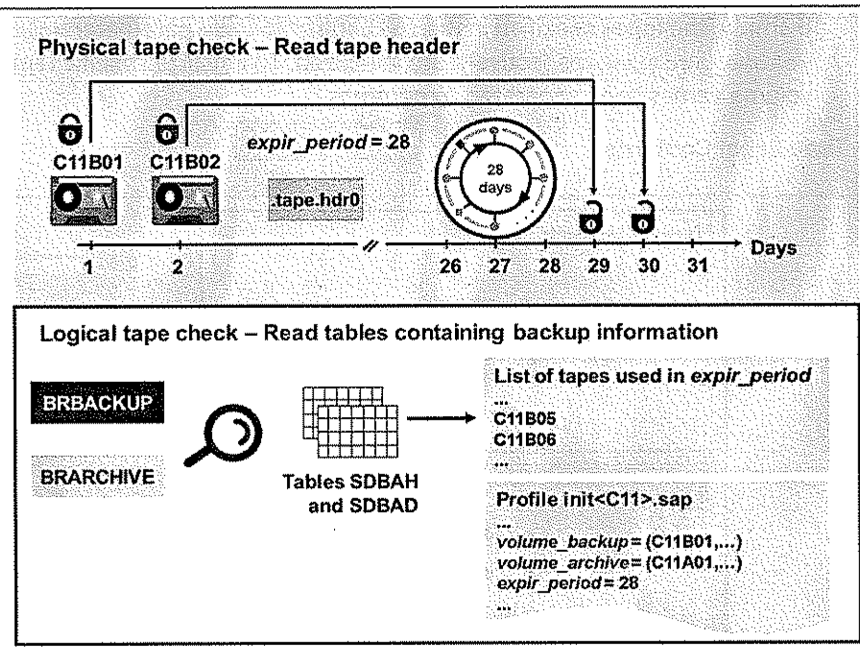


Figure 63: Tape Checks

At the beginning of a backup, BRBACKUP and BRACHIVE write the timestamp to the header file on the tape. Additionally, when a database file has been backed up successfully, information about the database (including timestamp) is written to special backup log files, summary log and detailed log, and also to the database tables *SDBAH* and *SDBAD*. That is why BRBACKUP and BRARCHIVE can use two different methods for checking that a tape is not locked.

The different methods are as follows:

- The physical lock check is derived from the tape label. The timestamp of the last backup found in the tape label and the parameter `expir_ period` found in `init<DBSID>.sap` determine whether the tape can be reused. If the number of days since the tape was last used is less than the value of the parameter `expir_period`, the tape is physically locked.

- The logical check is derived from the timestamp written to the tables SDBAH and SDBAD. To find which tapes can be used for the next backup and distinguish them from those that are still locked, BRBACKUP connects to the database and searches tables SDBAH and SDBAD for the tapes that were used in the lock period. These tapes cannot be used for the next backup, they are locked logically. The tape that follows the last used tape in the parameter `volume_backup`, and is not contained in the list of tapes used in the lock period, is selected for the next backup. After the last volume on the list is used, the first volume on the list is requested again.

The logical lock check for the offline redo log file backups is performed by BRARCHIVE using information from the summary log. Therefore, offline redo log files can be backed up even when the database is not available.

Under certain circumstances, discrepancies may occur between the physical and logical locks.

At the beginning of a backup, the volume label is written to the tape. If the backup is terminated before the first database file can be written to the tape, the volume is locked physically but not logically. At the next backup run, the volume is selected from the list (`volume backup or volume archive`) but rejected when the physical volume label check takes place. Therefore, you must reinitialize the volume with the same name to cancel the physical lock. Because the expiration period for this tape is not over yet, you must use initialization type `force` to cancel the physical lock.

If you reinitialize a volume before the expiration period ends (using initialization type `force`), this volume is no longer locked physically. However, it will not be selected automatically for the next backup as long as it remains locked logically. If you want to use this volume before the logical lock has expired, you can switch off the automatic tape selection temporarily by inserting the tape and performing the next backup for a volume named `scratch`.

Tape Selection

SAP tools provide the following procedures for selecting a tape for a backup:

- Automatic tape selection by BRBACKUP or BRACHIVE
- Manual tape selection by the operator
- Tape selection by an external tool

Automatic Tape Selection

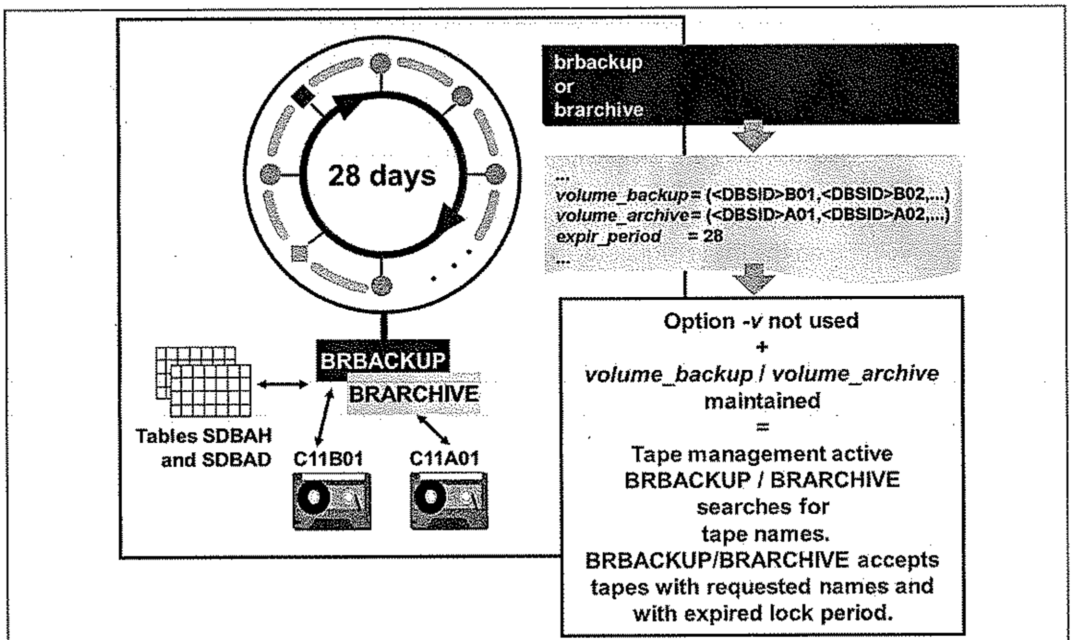


Figure 64: Automatic Tape Selection

If you want BRBACKUP or BRACHIVE to select the tapes to be used for the next backup run automatically, consider the following points:

- You must define the parameters `volume_backup` and `volume_archive` in the profile `init<DBSID>.sap`.
- You must not specify a volume when calling or scheduling the backup program.

BRBACKUP or BRACHIVE performs the logical lock check and requests the next unlocked tape in the order defined by the parameter `volume_backup` or `volume_archive`. The mounted tape is checked physically as well. If the operator does not mount the requested tape, the program aborts with an error.

To check which tape is automatically selected by a specific backup, select a backup in the DBA Cockpit or transaction `DB13` and display the action details by double-clicking the selection. At the OS level, you can check this using `brbackup|brarchive -q| -query [check]` to check which tape will be automatically selected on the next backup. Use the parameter `check` to perform a physical tape check.

Manual Selection of Tape with Any Name

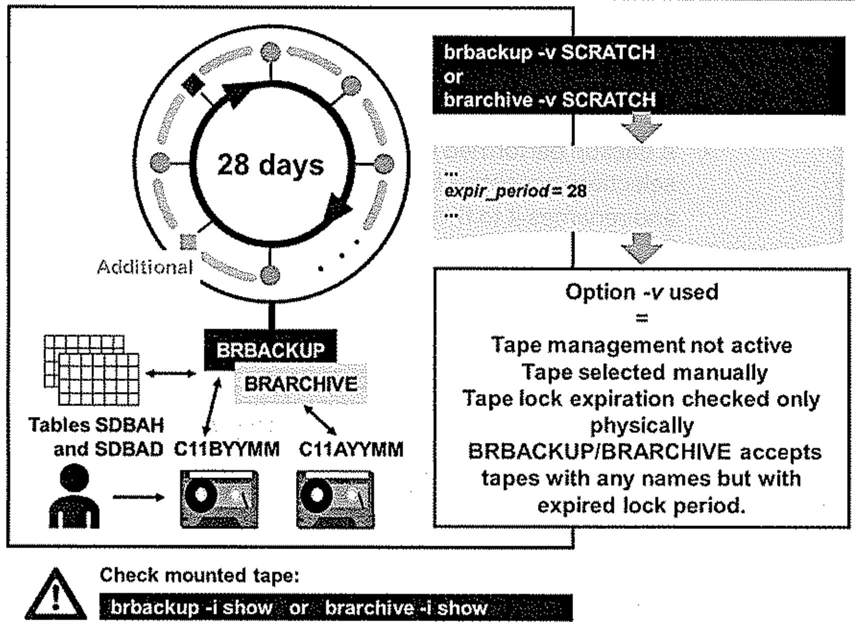


Figure 65: Manual Selection of Tape with Any Name

To allow for a backup on a volume with any name, specify the volume `scratch` when you start a backup from BRTOOLS or BRGUI, menu. Choose *Backup and database copy → Database backup and Backup and database copy → Archivelog backup* respectively, or use the option `-v|-volume scratch` for BRBACKUP or BRARCHIVE.

Performing a backup by specifying the symbolic volume name `scratch` switches off automatic tape management. BRBACKUP or BRARCHIVE performs just the physical lock check and accepts any initialized tape with an expired lock period. The operator must check that the proper tape is mounted and used for this backup. If the wrong tape is mounted by accident, it will be used (as long as it is not locked) and the old data on it will be overwritten.

Use this method, for example, for creating an additional month-end backup if you do not want this backup to be performed on your tape pool tapes.

You can also initialize some tapes with the symbolic volume name `scratch`. These scratch tapes can be used, for example, to replace a defective tape in your tape pool. When automatic tape selection is used (meaning no tape volume was specified with the backup), a scratch tape is automatically renamed to the volume name of the tape that was requested.

> Hint:
>
> **Do not confuse the following scratch options:**
>
> - A backup performed specifying the volume `scratch` accepts any initialized tape, which is not physically locked.
> - A backup performed without specifying any volume name (automatic tape selection) accepts a tape that was initialized with the volume name `scratch` and renames it to the requested volume name.

External or Manual Tape Selection

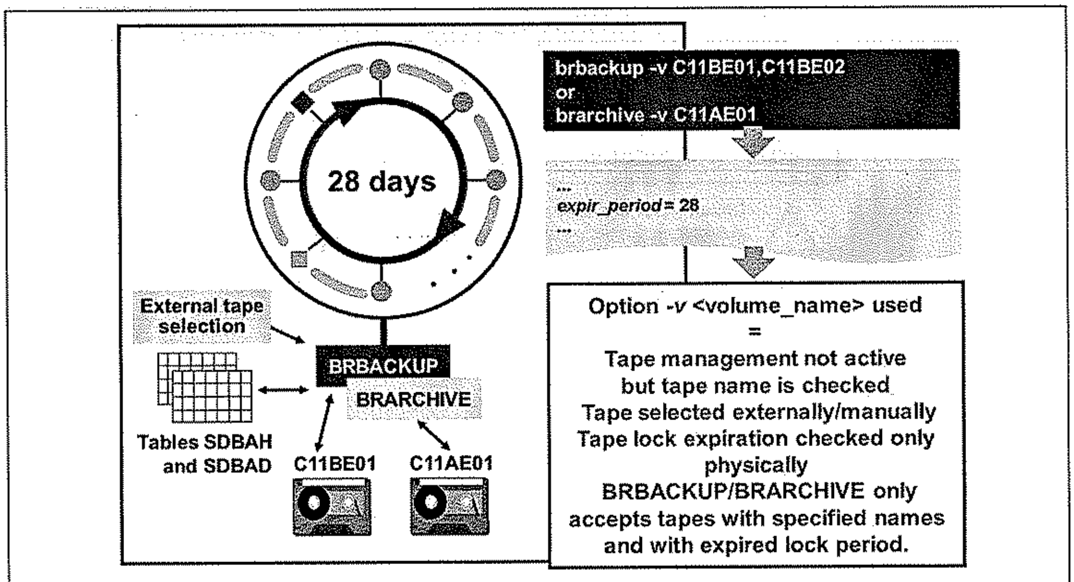


Figure 66: External or Manual Tape Selection with Specified Name

To specify the tapes to be used by BRBACKUP or BRARCHIVE, specify the volumes to be used with the option `volume` when starting a backup from BRTOOLS or BRGUI, choose menu *Backup and database copy → Database backup and Backup and database copy → Archivelog backup* respectively. You can also use the option `-v` | `-volume <volume>` for BRBACKUP or BRARCHIVE. Specifying a volume always deactivates the automatic tape selection, but the physical tape check is performed and the locked tapes are rejected.

Tape Layout

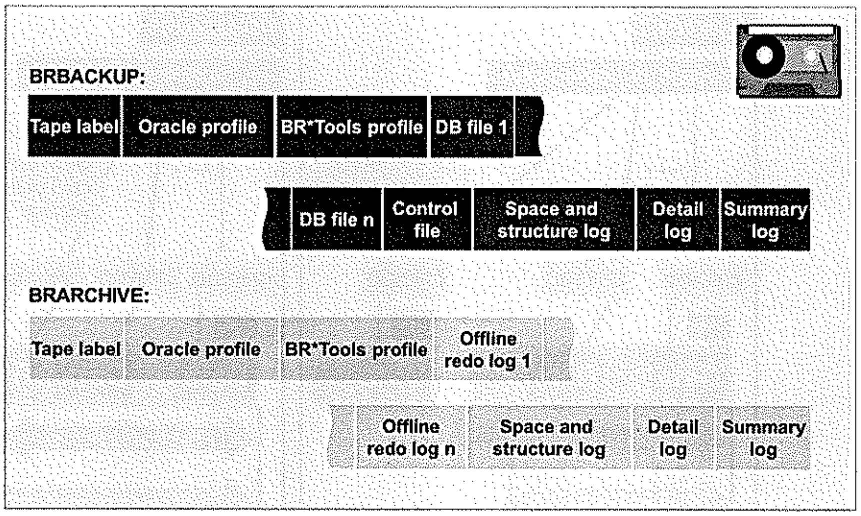


Figure 67: Tape Layout

The following table describes the files that are written to tapes by BRBACKUP and BRACHIVE:

File name	Description
`.tape.hdr0`	Tape label
`init<DBSID>.ora`	Database configuration file (Oracle profile)
`init<DBSID>.sap`	BR*Tools configuration file
`space<DBSID>.log`	Information about the creation, extension, or reorganization of tablespaces or tables (on the disk located in the `sapreorg` directory)
`struc<DBSID>.log`	History of database structure changes (located in the `sapreorg` directory)
`<action_ID>.<function-ID>`	A detail log of BRBACKUP/BRARCHIVE: the complete output of the BRBACKUP or BRARCHIVE run (located in the `sapbackup` or `saparch` directory)
`back<DBSID>.log`	A summary log of BRBACKUP – a list of all backups started with BRBACKUP (located in the `sapbackup` directory)
`arch<DBSID>.log`	A summary log of BRARCHIVE – a list of all offline redo log files backed up by BRARCHIVE (located in the `saparch` directory)

Configuration of Correct Tape Size

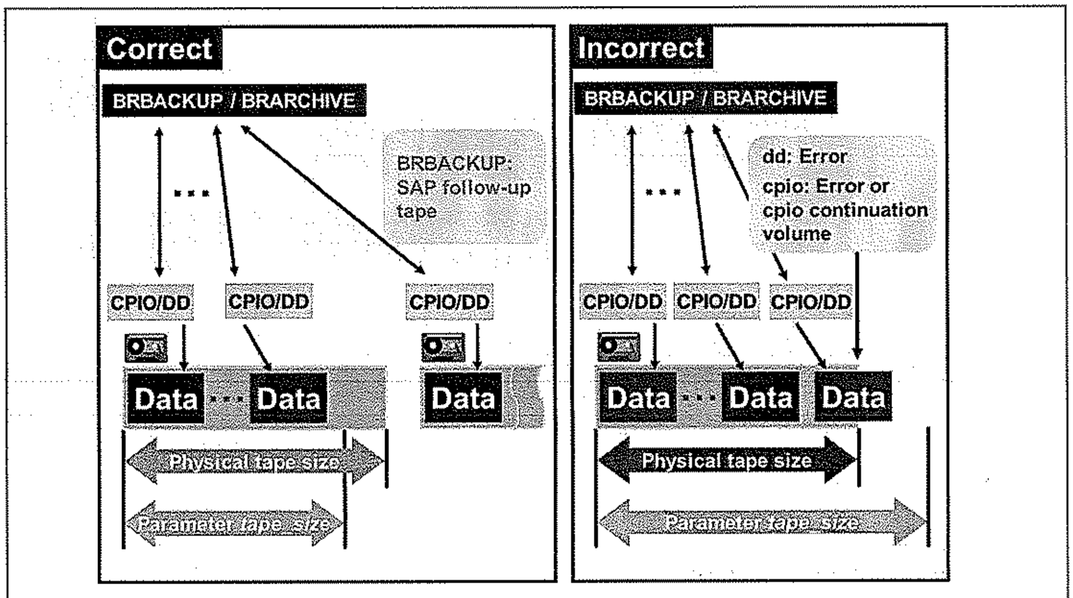


Figure 68: Initialization Parameter tape_size

The programs BRBACKUP and BRARCHIVE get the information about the memory capacity of the tapes to be used from the parameter `tape_size` or `tape_size_arch`. The parameter `tape_size` logically defines the memory capacity in gigabytes (GB), megabytes (MB), or kilobytes (KB) for the tapes that will be used for backups with BRBACKUP. The parameter value is also valid for tapes used by BRARCHIVE if the corresponding BRARCHIVE parameter `tape_size_arch` is not set. At the beginning of a backup, BRBACKUP / BRARCHIVE determines the data volume to be backed up and plans the distribution of this data over the initialized SAP tapes using the specified parameter value. Files are never split, they are backed up to a tape in one piece. The same statement is true for files on raw devices.

> Caution:
> The largest file and the largest raw device volume for a backup may not be larger than the value specified in `tape_size` (after compression, when applicable).

The value of the parameter `tape_size/tape_size_arch` should be slightly smaller than the physical tape capacity. A few more megabytes of space are needed (compared to the total size of data to be backed up) for backing up init-files and backup log files, and for writing CPIO file headers. This additional space is not taken into consideration when the total space needed is calculated. To be on the safe side, allow for a 10% safety margin when setting the parameter `tape_size/tape_size_arch`.

When all files are copied to a tape according to the plan, regardless of how much space remains free on the tape, BRBACKUP asks for an SAP follow-up tape. After the tape has been made available, BRBACKUP continues backing up data.

In contrast to BRBACKUP, BRARCHIVE does not have its own management of follow-up tapes. During an offline redo log file backup, the maximum number of offline redo log files that can fit on one tape (as defined by `tape_size` or `tape_size_arch`) is backed up. An SAP follow-up tape is not used. When the tape is full, you must start a new BRARCHIVE run to write to the next volume.

If the value for `tape_size` is too large, too many files may be planned for a copy to one tape. The copy program (CPIO or DD) then reaches the physical end of the tape while copying a file that does not fit onto the tape.

Depending on the copy program and the type of backup, the consequences of copying to tape are as follows:

- The copy program DD always generates an error message when it reaches the end of the tape. The error message depends on the OS. In Windows, the message *Physical End of tape has been reached* appears, and in UNIX the message *I/O-Error* appears. The backup process terminates with an error.
- During a serial database (or offline redo log file) backup, CPIO requests a CPIO (not an SAP tool) continuation volume and the backup process continues.

> Caution:
> Although the database backup terminates successfully, problems may arise during a restore from this database backup because SAP tools do not request the CPIO continuation volume directly.

- During a parallel database (or offline redo log backup), CPIO stops with an error message and the entire backup process terminates with an error.

Avoid reaching the physical end of a tape.

Hardware Compression and Tape Size

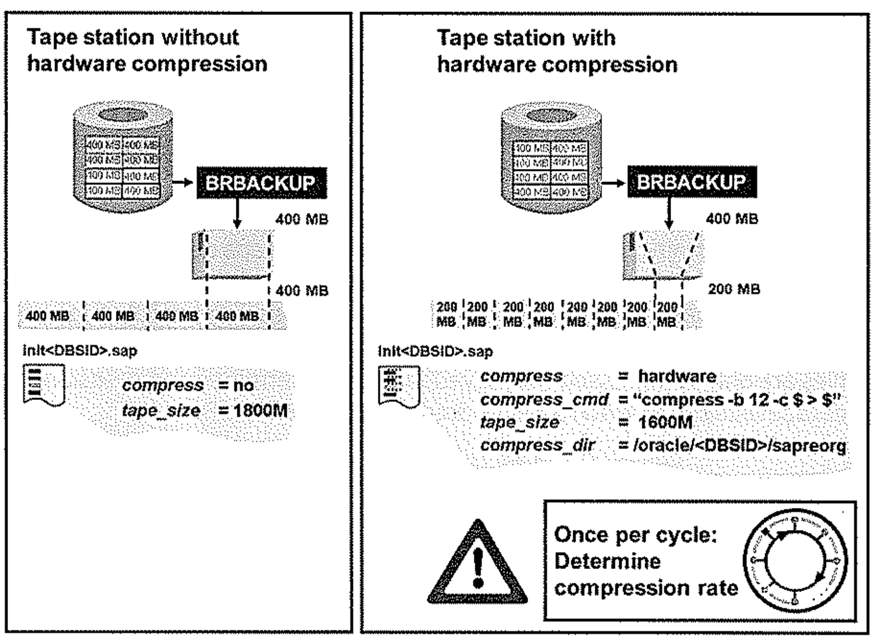


Figure 69: Hardware Compression

With the help of the initialization parameter `compress`, you can determine whether files will be compressed during the backup, or whether BRBACKUP or BRARCHIVE considers hardware compression when planning distribution of files over tapes.

The values of the compress parameter are as follows:

- `compress=no`

  Software compression is not used. You may use hardware compression, depending on the backup device, but do not use BRBACKUP or BRARCHIVE.

- `compress=yes`

  Software compression is used. In this case, you must also set the compress command in the parameter `compress_cmd`. The parameter value depends on the OS; examples are within the comments of `init<DBSID>.sap`. Furthermore, specify the directory in which compression is to be performed in the parameter `compress_dir`.

- `compress=hardware`

  This parameter can be set when tape units that support hardware compression are used. Setting this parameter to `hardware` does not activate hardware compression. This parameter is information for BRBACKUP to use the current compression rates (as for software compression) when calculating how much data will fit on one tape. You must also configure your backup device accordingly.

If you use software or hardware compression for files, the parameter `tape_size` specifies the total size of the files that will fit on one tape after compression. The corresponding space needed to store compressed files on tapes is calculated by BRBACKUP with the help of the current compression rates. BRBACKUP can properly determine the quantity of data to be saved on one tape after the compression only when it uses the correct compression rates of the database files. Ensure that the specified tape size is not exceeded and the database files are correctly distributed across the tapes.

When backup devices with hardware compression are used, BRBACKUP can only estimate the quantity of data that can be written to a volume. This is because these tools cannot directly determine the compression rates for hardware compression (tape stations do not report a compression rate). BRBACKUP uses the software compression rates as an estimate for hardware compression rates. BRARCHIVE always assumes a compression rate of 1:1 (no compression) for offline redo log files.

Before performing the first database backup using tape devices with hardware compression, you must start a compression run to determine the compression rates. In the DBA Cockpit or transaction `DB13`, start the action *compress database*, or execute `brbackup -k only`. This call does not actually start a backup; it only determines the compression rates. The database files are only compressed (not saved) and the determined compression rates are stored in a database table (*SDBAD*) and in a detail log of BRBACKUP.

> Caution:
> Repeat this activity of updating compression rates at least once per backup cycle or once a month and, additionally, after reorganization or after loading of a large amount of data.

> Hint:
> To determine the compression rates as close to the actual hardware compression rates as possible, set the parameter `compress_cmd` to `compress -b 12 -c $ > $` on UNIX platforms and to `mkszip -l 0 -c $ > $` on Windows (see SAP Note 19909).

The actual space needed for storing compressed files on tapes may differ from the value calculated by BRBACKUP with the help of compression rate estimates. Therefore, you must set the value of the parameter `tape_size` to an even smaller value than without compression (as an additional safety margin) to prevent the problem of reaching the physical end of the tape during a backup.

> Hint:
> If you want to use hardware compression, minimize the risk of reaching the physical end of the tape, and circumvent the need for updating the compression rates, set the parameter `compress` to `no` and `tape_size` to twice the physical size of your tapes. This works in most cases because the compression rates are usually in the ratio of least 2:1.

If the parameter `exec_parallel` is set to 0 during compression rate determination, one process per logical volume is triggered to determine the compression rate. If you set the parameter `exec_parallel` to a positive value smaller than the number of logical volumes, the number of processes required to determine the compression rate is limited to the number indicated by the parameter value. This reduces the CPU load on the database server.

LESSON SUMMARY

You should now be able to:

- Work with backup tools

Unit 2
Lesson 3

Performing Backups

LESSON OVERVIEW

This lesson explains how to perform backups using BR*Tools.

Business Example

While your daily backups run fine, you have learned about different backup types and scenarios and you want to test other scenarios. For this reason, you require the following knowledge:

- An understanding of how to perform online, offline, and partial backups and Oracle Recovery Manager (RMAN) backups, including incremental backups
- An understanding of how to create backups of archived redo log files

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Perform backups to the system

Backups

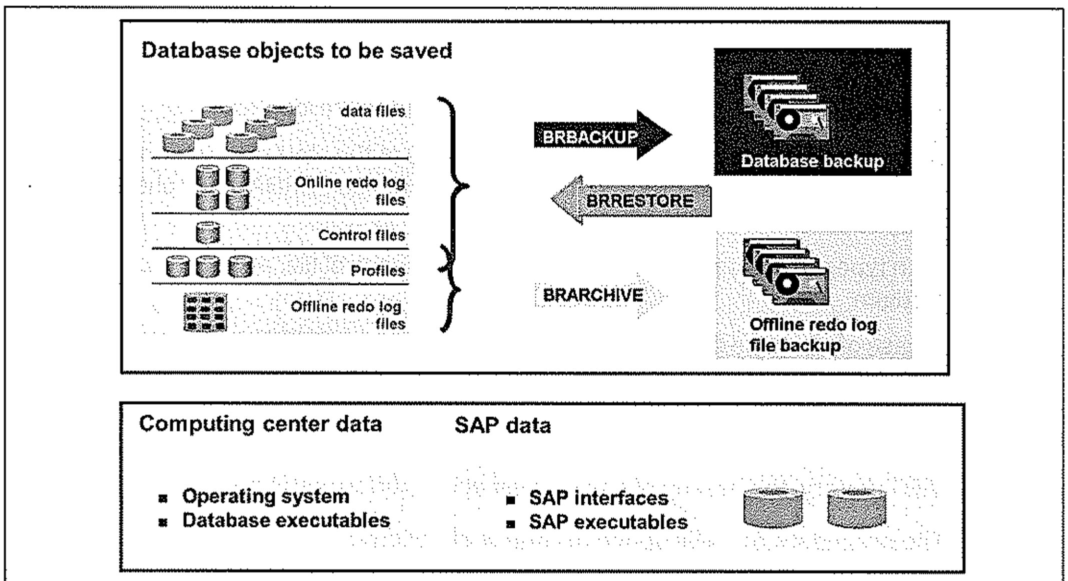


Figure 70: Objects for Backup

The primary task of BR*Tools, with respect to data security, is to back up all business data. However, your backup strategy must include backing up all objects, including the operating system (OS) files associated with the SAP system (SAP executables, interfaces, and archiving objects), and files in the Oracle software directories. These objects are usually backed up at OS level. Consider creating such a backup at least once per backup cycle.

In addition to the selected files, BRBACKUP always backs up the control file, the profile, and specific log files. A complete offline backup also backs up online redo log files. BRARCHIVE primarily saves offline redo log files in the backup medium, along with the profiles and log files.

Phases of a Database Backup

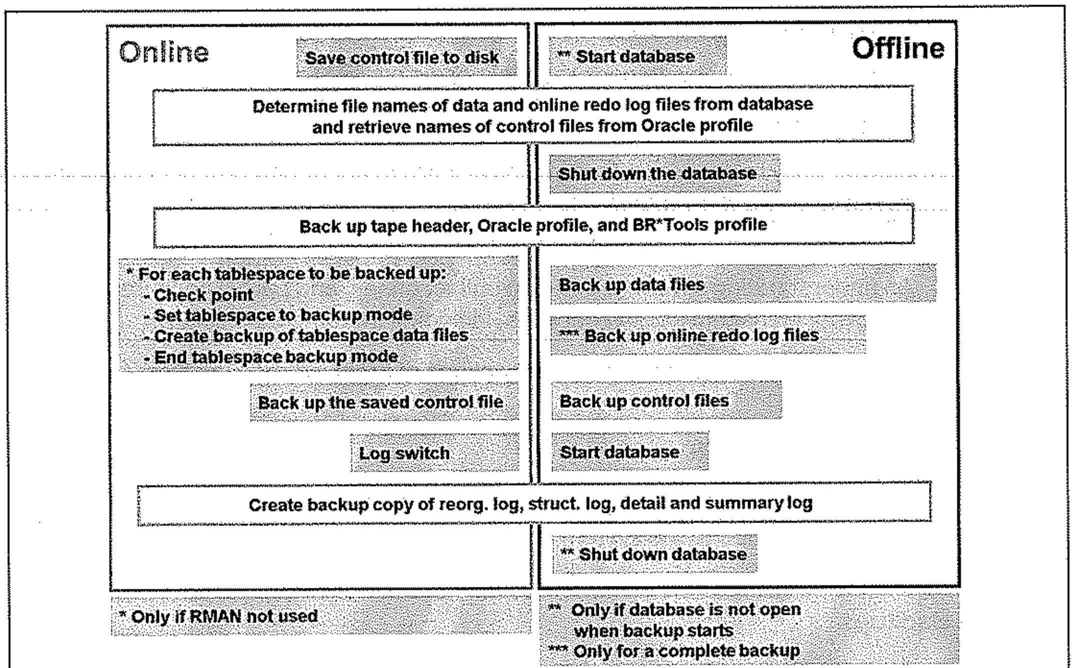


Figure 71: Phases of a Database Backup

Some steps of offline and online backup procedures are identical.

Both types of backup write a tape header (label) at the start of a backup to tape and read the header at the end of the backup. By reading the header, BRBACKUP checks whether the tape was correctly written.

Both types of backup also write the profiles, the control file, and some log files to the backup medium.

In case of an online backup, you cannot back up the control file during normal database operation. Therefore, at the start of the backup, a consistent copy of this control file is made to disk. This copy is backed up to tape after all files have been backed up.

In a standard online backup, without using the Oracle RMAN and a System Backup to Tape (SBT) library, the backup of each tablespace is performed in backup mode.

The procedure for tablespace backup is as follows:

1. Enter the oracle statement: `ALTER TABLESPACE BEGIN BACKUP` to enter the tablespace in the backup mode.

   This has the following consequences:

- A checkpoint on tablespace level is triggered before the backup of this tablespace starts and the database writer copies all corresponding dirty blocks of this tablespace to the data files.
- The system change number (SCN) corresponding to the tablespace checkpoint remains "frozen" in the headers of all tablespace files until the end of backup mode.
- Every data change performed in the tablespace is logged in the redo log on the data block level, instead of data record level. The whole 8 kB block containing this record is written into the redo log, in addition to the redo information for a modified record. The reason for logging the entire 8 kB block is to prevent Copy In and Out (CPIO) or Data Description (DD) from copying an Oracle block to the backup medium in units smaller than 8 kB. For this reason, the block is changed by the database writer, which can lead to inconsistencies within such a block. When the backup is later used for restoring the tablespace, all blocks written to the redo log during the backup mode overwrite the "suspicious" versions saved directly in the backup of the data files. The "frozen" SCN informs Oracle where the part of the redo log that contains the blocks of the corresponding tablespace starts.

2. When the last data block of the tablespace has been backed up, the tablespace is reset and the backup mode ends (`ALTER TABLESPACE END BACKUP`).

   Online backup with Oracle RMAN does not set tablespaces into backup mode because RMAN ensures that each data block is copied to the backup medium in a consistent state. At the end of an online backup, BRBACKUP performs a redo log switch.

3. Carry out a backup of offline redo log files and ensure that all redo information written during your online backup is there. This is important for data security, because the activities (data changes) are allowed in the database during an online backup, the backup does not contain a consistent snapshot of the database data. When the data is copied back to disk during database restoration, the data can only be made consistent if the corresponding redo information is available.

> Hint:
> Do not shut down the database in an extra step as a preparation for an offline backup. Because BRBACKUP must read tables *SDBAH* and *SDBAD* (containing a log of backups) and some Oracle dictionary views at the beginning of each backup procedure, BRBACKUP opens the database when it is closed. BRBACKUP then automatically shuts down the database before backing up data files, then opens the database again at the end to write backup protocol into *SDBAH* and *SDBAD*. BRBACKUP leaves the database as it was found. If the database was not open at the beginning of the process, BRBACKUP shuts the database down.

The online redo log files are only included in the backup during a complete offline backup.

Integration of SAP Backup Tools

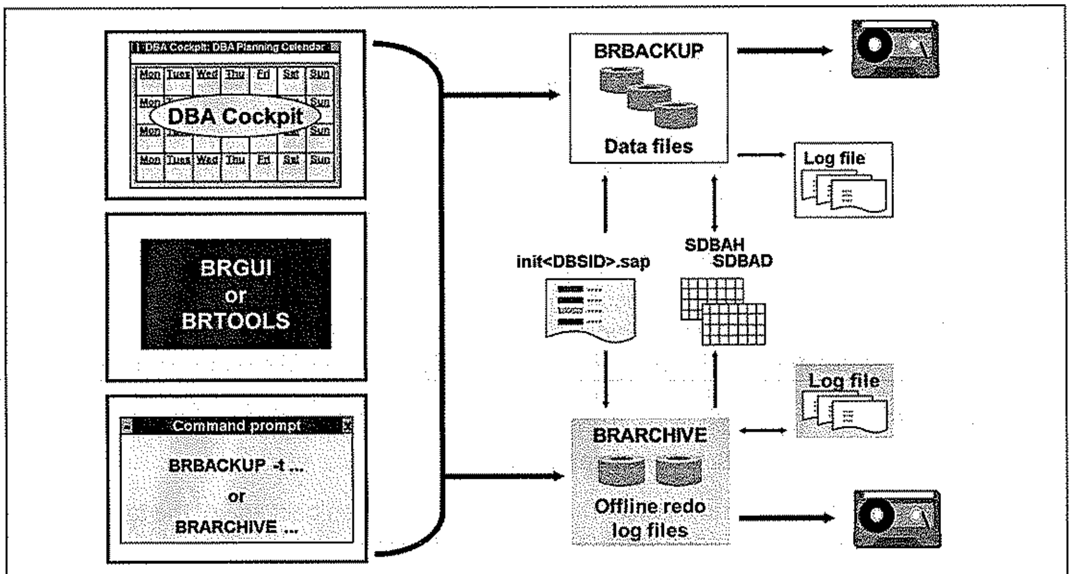


Figure 72: Integration of SAP Backup Tools

There are several interfaces through which you can start or schedule a database backup or an offline redo log backup. Regardless of which one you use, you always start BRBACKUP or BRARCHIVE, and these two tools always log backup actions in database tables *SDBAH* and *SDBAD*, as well as in their own summary log and a detail log per action. For internal tape management, BRBACKUP determines the required tapes from tables *SDBAH* and *SDBAD*, while BRARCHIVE does this based on its summary log.

For processing, BRBACKUP and BRARCHIVE read values of configuration parameters from the BR*Tools profile `init<DBSID>.sap`. Values found there override default values set in the code. However, parameter values specified at command line in a job definition or interactively in a menu have the highest priority.

> Hint:
> Selecting parameter values for a BRBACKUP or BRARCHIVE run does not change the values in the profile. To change a parameter value in the profile, you must use an OS editor. There is no SAP transaction for the maintenance of this profile.

Creating Database Backups

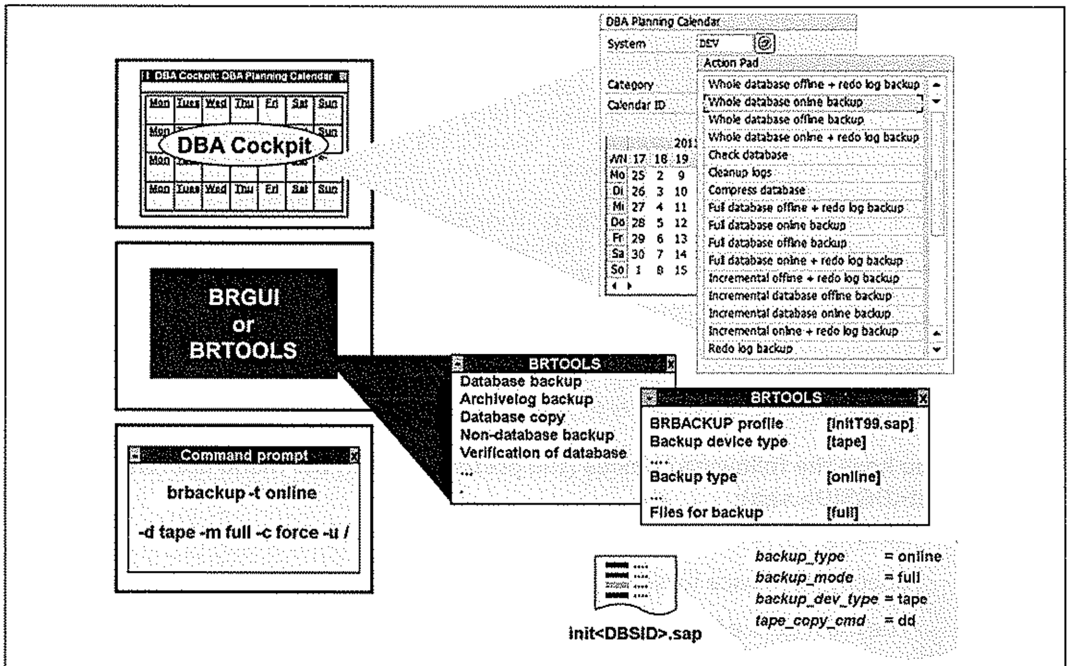


Figure 73: Scheduling and Performing Database Backups

At SAP level in the DBA Planning Calendar within the DBA Cockpit (transaction `DBACOCKPIT`) or transaction `DB13`, you can schedule various types of BRBACKUP and BRARCHIVE jobs. This is the recommended method for scheduling of all periodic databases and offline redo log backups in a backup strategy. The jobs are created as common SAP background jobs (which call BRBACKUP or BRARCHIVE on OS level) so that the scheduling can be monitored in `SM37`. Selecting a certain type of backup in `DB13` corresponds to particular choice of configuration parameter values.

In a special case in which you are not allowed to, or do not want to, access the SAP system through the SAP GUI, you can schedule BRBACKUP and BRARCHIVE jobs on the OS level, using their command line options and OS scheduling (UNIX: CRON, Windows: AT or graphical task scheduler).

For all further database backups (one-time actions and exceptional cases), you can use BRGUI or BRTOOLS and start a backup run interactively. Choose *Backup and database copy* from the main menu, then select the backup function you want to perform. Each selection allows you to specify relevant program options for the BRBACKUP or BRARCHIVE action as needed.

Performing Backups Using BRTOOLS or BRGUI

In most cases, backups are scheduled from the DBA Cockpit or transaction `DB13`.

To perform a backup of the database with BRTOOLS or BRGUI, choose *Backup and database copy → Database backup*.

This displays the following menu:


```
BR0657I Input menu 15 - please check/enter input values
-------------------------------------------------------------------------
--------
BRBACKUP main options for backup and database copy

 1 - BRBACKUP profile (profile) ....... [initT99.sap]
 2 - Backup device type (device) ...... [disk]
 3 # Tape volumes for backup (volume) . []
 4 # BACKINT/Mount profile (parfile) .. []
 5 - Database user/password (user) .... [/]
 6 - Backup type (type) ............... [offline]
 7 # Disk backup for backup (backup) .. [no]
 8 # Delete disk backup (delete) ...... [no]
 9 ~ Files for backup (mode) .......... [all]

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-------------------------------------------------------------------------
--------
BR0662I Enter your choice:
```

You have the following choices (the preset values in the menu are read from init<DBSID>.sap):

- `Backup device type (device)`

  Specify the device to which the backup needs to be performed.

  The main device types are as follows:

  - `tape`

    This device type performs a backup to tape. Device types `tape_auto` and `tape_box` support tape autoloaders and jukeboxes.

  - `disk`

    This device type performs backups to a local directory specified by the `backup_root_dir` parameter in `init<DBSID>.sap`.

  - `stage`

    This device type performs backup to a remote directory specified by the `stage_root_dir` parameter in `init<DBSID>.sap`.

  - `util_file` and `util_file_online`

    These device types perform backups using external backup tools with BACKINT.

  - `rman_disk`

    This device type performs backups using RMAN and an external SBT library. Profiles, control files, and log files are saved to disk. Alternatively, use `rman_util` for RMAN with external SBT library. The system saves the profiles, control files, and log files through BACKINT.

- `Tape volumes for backup (volume)`

  If the backup device type is `tape`, `tape_auto`, or `tape_box`, you can optionally specify the volume.

- `Backup type (type)`

The main backup types are online and offline. When performing an offline backup, you must shut down the database. To force the database to shut down, even when an SAP system is connected, use `offline_force`.

- `Back up disk backup (backup)`

  BRBACKUP fully supports a two-phase backup strategy in which a backup is performed to device `disk`, and then the disk backup is saved on tape. Select yes to backup a previous disk backup to tape and decide with option `Delete disk backup(delete)` to delete the disk backup after a successful backup to tape.

- `Files for backup (mode)`

  (`all`|`full`|`incr`) backup mode is specified. For a partial database backup, specify the tablespace(s) to be backed up.

Special Backups and Options

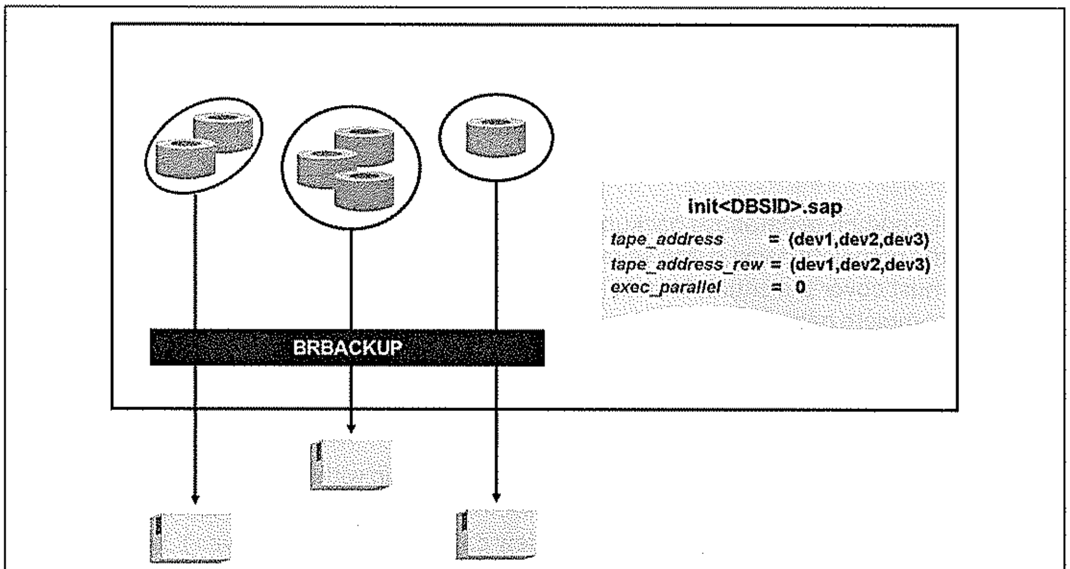


Figure 74: Parallel Database Backup

To reduce the time required to back up and restore the data files and offline redo log files, SAP backup tools support the parallel use of several tape stations.

BRBACKUP uses all tape stations defined in parameters `tape_address` and `tape_address_rew` in the profile `init<DBSID>.sap`. Both parameters must contain the same list of tape station addresses (no-rewind and rewind driver), and all used tapes must be the same size.

The database files selected for a backup are distributed across the tapes mounted in the tape stations. To keep backup times to a minimum, make sure the tape capacity is significantly larger than the total volume of data to be backed up to a tape.

Partial Database Backups

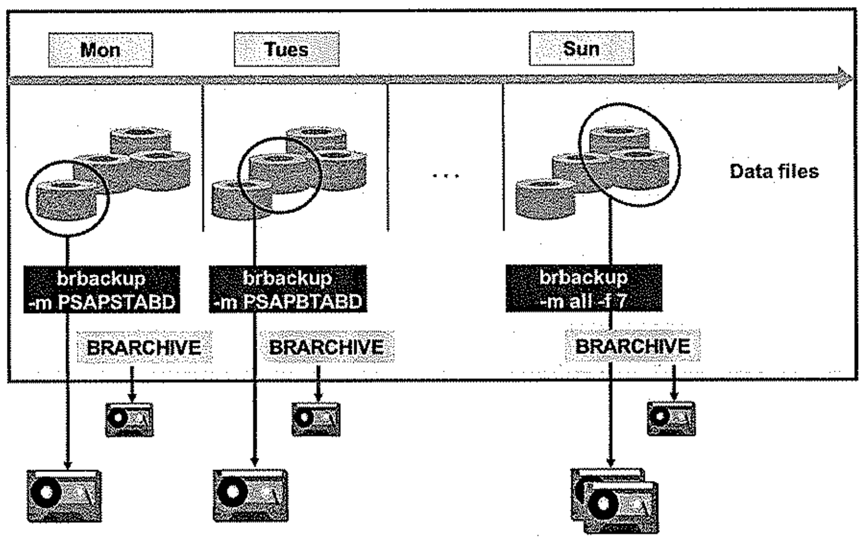


Figure 75: Partial Database Backups

If a complete database backup takes too long in your production environment, you can split the complete backup into several partial backups. However, the sum of the partial backups must cover the entire database in the selected time interval in which you would create one complete backup.

Both Oracle and SAP tools support the recovery of data files from different backup runs. For this type of recovery, the tools require all offline and online redo log files that have been generated since the oldest backup of data files.

To ensure that the complete database is backed up within the selected time interval, use BRBACKUP option `-f|-fill <days>`. The corresponding backup run completes the partial backups performed for the previous few days (specified as `<days>`). You must use BRBACKUP directly or from BRTOOLS or BRGUI, because transaction `DB13` does not support this option.

> Hint:
> You can also use this procedure to complete aborted backup runs. In this case, specify the log file associated with the aborted backup run, `-f|-fill <log_file>|last`.

Verification of Backups

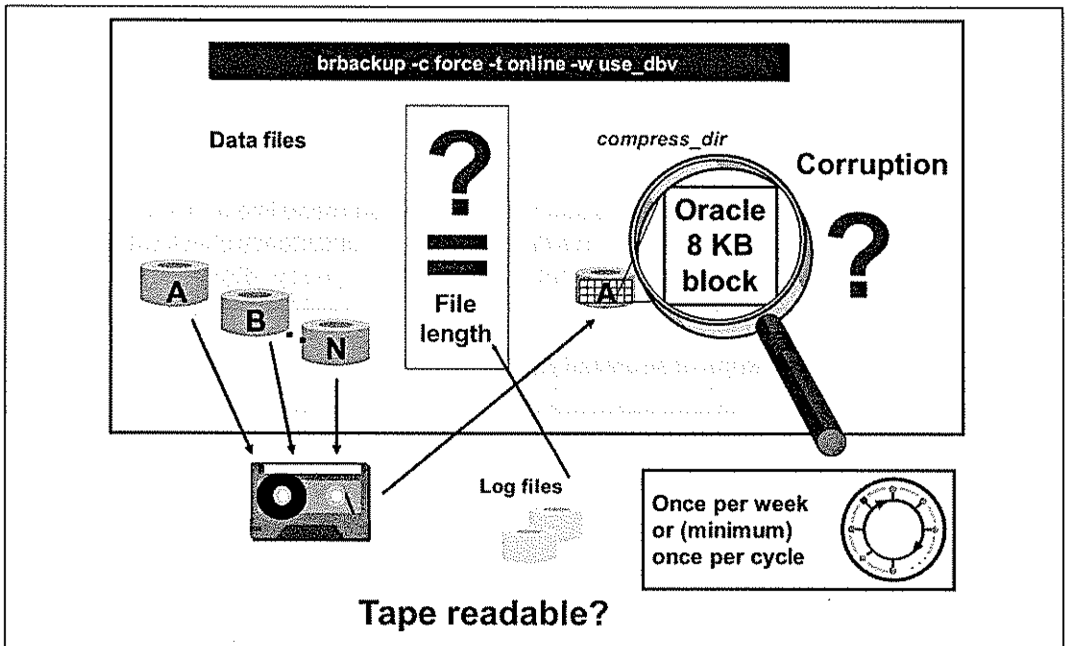


Figure 76: Verification of Backups

Even if a backup is reported as successfully completed, the backup may not be error free.

You can use the following types of verification to check whether the backup is complete and error free:

- Tape verification

  To verify whether a backup is readable, the files are restored file by file and compared with the originals. Depending on the type of backup, the system uses different checks, from conducting a binary comparison to simply comparing the size when online backups are performed.

  Run this check once per week or, at minimum, once per backup cycle.

- Block consistency

  This type of verification checks the database block by block, using the Oracle tool DBVERIFY.

  Run this check at least once per backup cycle. Whenever you have a bad Oracle block in a data segment, you must be able to restore the corresponding data file from a backup that does not include the bad block.

To perform a backup verification, proceed as follows:

1. Start BRTOOLS or BRGUI and choose *Backup and database copy → Verification of database backup* and *Backup and database copy → Verification of archivelog backup* respectively.

2. Choose the backup to be verified from the list and choose the type of verification in the option *Use DBVERIFY (use_dbv)*.

The types of verification and the action it performs are as follows:

- `No`

  It means that only tape verification is performed.

- `Yes`

  It means that tape verification is performed and database block consistency is checked.

> Hint:
> When verification is started from the command line or from BRTOOLS or BRGUI, you can also perform a database block consistency check without tape verification using the option `-w | -verify only_dbv`.

Creation of Backups of Archived Redo Log Files

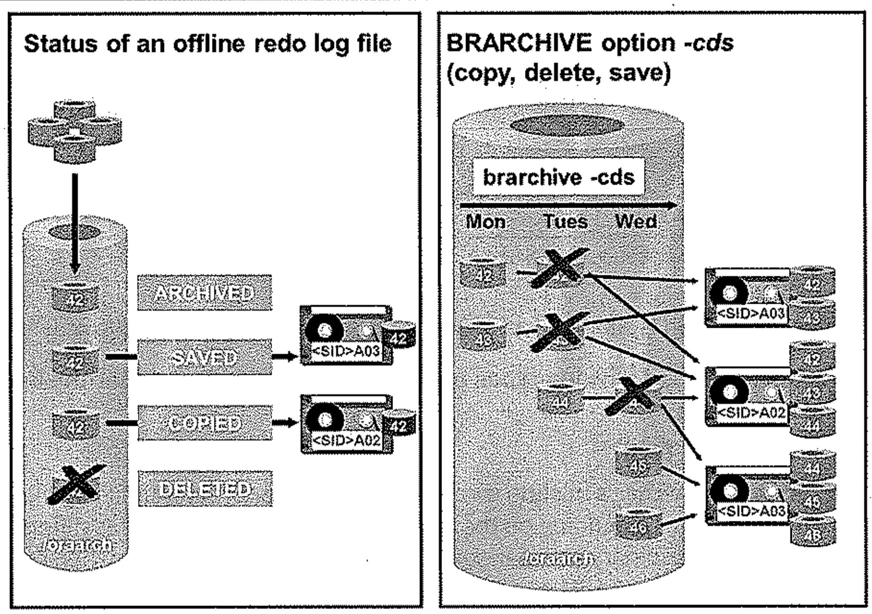


Figure 77: Backup and Status of Offline Redo Log Files

After a log switch, the Oracle process ARC0 copies the online redo log file that was the current redo log file before the log switch to directory `oraarch` as an offline redo log file. BRARCHIVE copies offline redo log files from this directory to a backup medium.

An offline redo log file can have various statuses for BRARCHIVE. These statuses are always updated in the summary log `arch<DBSID>.log` after a BRARCHIVE run.

During a backup to tape, an offline redo log file has the status ARCHIVE. At the first save, the file status is SAVED; the second time, it is COPIED; and after deletion, it has the status DELETED.

During a backup to disk, an offline redo log file has the status DISK. A second copy is not supported. The only statuses here are DISKSAV (first save to disk) and DISKDEL (deletion after a save to disk).

BRARCHIVE has several call options (functions) that determine how the offline redo log files are processed. SAP recommends using the option `-cds` (copy_delete_save), which is also the default option when starting BRARCHIVE from the DBA Cockpit or from transaction `DB13`.

First, all offline redo log files with status SAVED are saved to tape for a second time and subsequently deleted from disk. Then, all offline redo log files with status ARCHIVE are backed up to tape for the first time and their status is changed to SAVED.

After the backup, all offline redo log files exist at two locations, either in directory `oraarch` and on tape or on two different tapes. Thus, you can achieve a high safety rate without drastically increasing the tape requirement.

Performing Backups of the Archived Redo Logs Using BR*Tools

To perform backups of the archived redo logs, start BRTOOLS or BRGUI and choose *Backup and database copy → Archivelog backup*. The menu shows the options and parameters to select; most of them are similar to BRBACKUP.

The following menu appears:

```
Performing Backups of the Archived Redo Logs Using BR*Tools
BR0657I Input menu 17 - please enter/check input values
-----------------------------------------------------------------------
--------
BRARCHIVE main options for archivelog backup and verification
1 - BRARCHIVE profile (profile) ...... [initT99.sap]
2 - BRARCHIVE function (function) .... [save]
3 - Backup device type (device) ...... [disk]
4 # Tape volumes for backup (volume) . []
5 # BACKINT/Mount profile (parfile) .. []
6 - Database user/password (user) .... [/]
7 ~ Maximum number of files (number) . []
8 # Back up disk backup (archive) .... [no]
Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-----------------------------------------------------------------------
--------
BR0662I Enter your choice:
```

Select the function of BRARCHIVE from the *BRARCHIVE function (function)* menu. Almost any combination of `save`, `copy`, and `delete` is possible.

For example, using different functions, you can perform the following:

- Implement a two-phase strategy

  The first run saves new archived redo logs (`save`) and the second run creates a second backup and deletes archives that have been successfully backed up twice (`second_copy_delete`).

- Create a parallel backup on two different tape stations (`double_save`) and later delete (`delete_copied`)

> Caution:
> Regardless of the strategy that you use, for security reasons, at least two copies of archived redo logs should exist at any time.

Verification of Backups of Offline Redo Log Files

Database archive log files and their backups can be verified with BR*Tools 7.00 as of patch 22 with the Oracle RMAN. Also, see SAP Note 1016173. The `RMAN VALIDATE` command is called internally. This is especially important for archive log files because until now, there has been no way of verifying internal consistency. RMAN verification covers database files with the

DBVERIFY function, which means that `RMAN VALIDATE` does not offer any essential advantages compared to DBVERIFY.

RMAN verifications can be activated using the command options BRARCHIVE, `-w | -verify use_rmv | first_rmv | only_rmv`.

The option `first_rmv` verifies the original files with RMAN before the archive log files are backed up.

Verifications of archive log files with RMAN are supported as of Basis 7.00 Support Package 12 in the DBA Cockpit.

> Hint:
> This new functionality can also be used for Oracle 9i after BR*Tools 7.00 has been correctly installed in this environment (see SAP Note 849483).

BRBACKUP and BRARCHIVE – One-Run Strategy

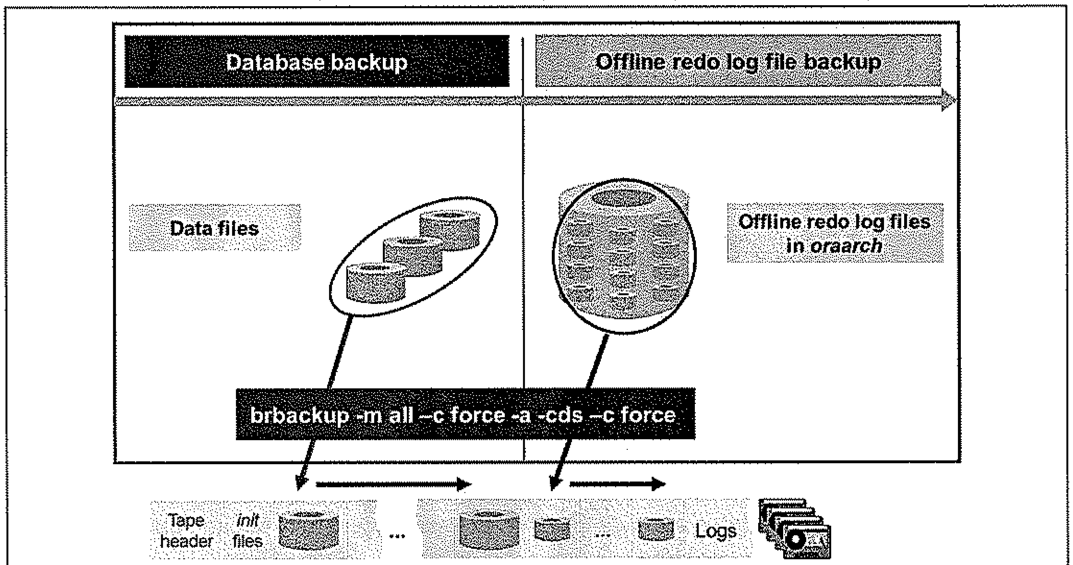


Figure 78: BRBACKUP and BRARCHIVE – One-Run Strategy

The advantage of the one-run strategy is that you can create a complete database backup and an offline redo log backup in one backup procedure. BRBACKUP and BRARCHIVE are called together rather than individually. Only one tape pool (in this case, the one defined in parameter `volume_backup`) is used. The offline redo log files are backed up to the tapes where the database files are backed up. This saves tapes and reduces management costs.

To define the one-run strategy for BRBACKUP, use the option `-a|-archive`. After this, the options for BRARCHIVE follow and in the DBA Cockpit or transaction `DB13`, you can define a corresponding job.

With this procedure, BRBACKUP backs up all database files and then, it starts BRARCHIVE passing it to the options entered after `-a|-archive`. BRARCHIVE first backs up the corresponding offline redo log files (as usual) and then, it backs up all logs, including BRBACKUP logs.

With the one-run strategy, the maximum number of offline redo log files that can be backed up is the number that can still fit on the BRBACKUP tape after the database backup. If more offline redo log files are generated daily than can be backed up, for example, because the database has grown or the number of offline redo log files is increasing, the archiver gets stuck (and, therefore, the database). This situation is called "archiver stuck". Therefore, you must regularly check whether the tape capacity is sufficient. If required, you should use larger tapes, an extra tape station, or another backup strategy.

> Caution:
> The one-run strategy cannot be used to resolve an archiver stuck because BRBACKUP attempts to connect to the database. If an archiver stuck is to be resolved using BRARCHIVE, tapes must be available in tape pool volume_archive.

Consistent Online Backups

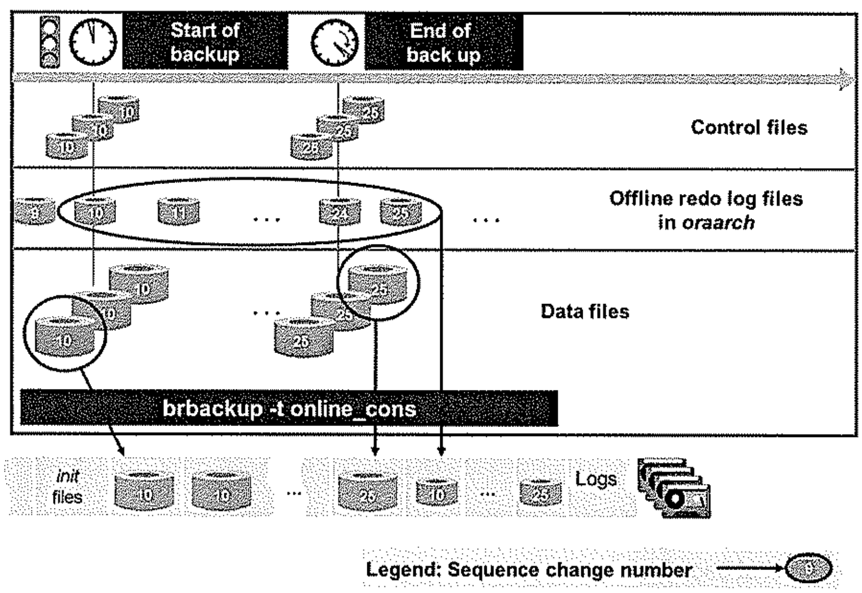


Figure 79: Consistent Online Backups

A consistent online backup is a database backup in online mode that contains logically consistent data. In this case, the offline redo log files generated during the backup are saved to the same volume as the database files that are backed up with BRBACKUP.

After backing up all data files online, BRBACKUP performs a log switch. BRBACKUP waits until the archiver process has finished copying the last redo log file into the directory `oraarch` and then copies the offline redo log files created during the online backup to tape. The last files on tape are the BRBACKUP summary and detail log.

The backup of the offline redo log files in a consistent online backup is completely controlled by BRBACKUP. Therefore, this run is independent of the BRARCHIVE backups and does not affect them. No entries are created in the `arch<DBSID>.log` summary log.

A consistent online backup can be performed either as a whole backup, a full backup, or as an incremental backup. This cannot be scheduled in the DBA Cockpit.

A consistent online backup can be used to reset the database to its status at the end of the backup. This is done by restoring data files and offline redo log files, and performing a point-in-time recovery.

> Hint:
> A consistent online backup is used for special backups that are done once a month or per quarter and put in long-term storage. It is also recommended to perform this backup before an SAP database upgrade.

Checking Backup Logs

Regularly check the result of all backups, using the following methods:

- Use the log viewer for DBA operations (transaction `DB14`). The log viewer is the main tool for checking the results of all backups because it enables you to view logs of all DBA activities.
- Use the DBA Planning Calendar in the DBA Cockpit (`DB13`). In the calendar, you see the scheduled actions, with colors indicating whether the actions have warnings or errors.
- Use transaction `DB12` to view the backup logs only, to create a recovery report, to view the status of the archiving directory, or to get an overview of archived redo log files.

Scheduling Backups from the DBA Planning Calendar

For regular backups (and other regular database actions), use the DBA Planning Calendar (transaction `DB13`) in the DBA Cockpit.

From the action templates, you can schedule any useful combination of the following backups:

- Whole backups
- Online backups
- Offline backups
- Partial backups
- Full backups
- Incremental backups
- Redo Log backups

All the templates offering a backup plus a redo log backup perform BRBACKUP and BRARCHIVE in one-run.

Actions for which backups are planned, such as tape initialization, determining compression rates, or the preparation run for RMAN backup can also be scheduled with transaction `DB13`.

> Hint:
> To use the DBA Planning Calendar, check that parameters in `init<DBSID>.sap` are correctly maintained. While parameters such as tape names can be specified when planning a backup with `DB13`, parameters like `device_type` or `tape_adress` cannot be selected from `DB13`.

Unit 2
Exercise 7

Perform Backups

Business Example

You want to learn how to perform different types of backup.

Perform various types of backups.

1. Perform a complete offline backup of the database and check that it is successfully performed.
2. Perform a complete online backup of the database.
3. Create some additional offline redo log files.
4. Perform a backup of offline redo log files. As the data is backed up on disk, save the offline redo log files, and delete them.

Unit 2
Solution 7

Perform Backups

Business Example

You want to learn how to perform different types of backup.

Perform various types of backups.

1. Perform a complete offline backup of the database and check that it is successfully performed.
   a) Start BRGUI or BRTOOLS and choose *Backup and database copy → Database backup.*
   b) In the *BRBACKUP main options for backup and database copy* input menu, notice the following values:

Parameter	Value
Backup type (type)	*offline*
Files for backup (mode)	*all*

   c) Choose *Continue* two times.
   d) Confirm the *Database instance T99 will be shut down now* message by choosing *Continue.*
      Monitor that the backup has been successfully performed by checking the log file.

2. Perform a complete online backup of the database.
   a) Start BRGUI or BRTOOLS and choose *Backup and database copy → Database backup.*
   b) In the *BRBACKUP main options for backup and database copy* input menu, enter the following data:

Parameter	Value
Backup type (type)	`online`
Files for backup (mode)	`all`

   c) Choose *Continue* two times.
   d) Confirm the *BR0106I Files will be saved on disk in directory:* <directory path> message by choosing *Continue.*
      Monitor that the backup has been successfully performed by checking the log file.

3. Create some additional offline redo log files.
   a) Start BRGUI or BRTOOLS and choose *Instance Management → Alter database instance.*

b) Choose *Continue* to open the BRSPACE menu *Alter database instance main menu*.

c) Choose *Switch redolog file*.

d) Choose *Continue*.

The output informs that the SQL command `alter system switch logfile` is performed.

> Note:
> You can repeat this step several times.

4. Perform a backup of offline redo log files. As the data is backed up on disk, save the offline redo log files, and delete them.

a) Start BRGUI or BRTOOLS and choose *Backup and database copy → Archivelog backup*.

b) In the *BRARCHIVE main options for archivelog backup and verification* input menu, enter **`save_delete`** in the *BRARCHIVE function (function)* parameter.

c) Choose *Continue* two times.

d) Confirm the *BR0106I Files will be saved on disk in directory:* <directory path> message by choosing *Continue*.

Check the log file to make sure the backup has been successfully performed.

LESSON SUMMARY

You should now be able to:

- Perform backups to the system

Unit 2
Lesson 4

Performing Restore and Recovery

LESSON OVERVIEW

This lesson introduces several restore and recovery scenarios. The lesson also explains how to restore and recover a database using BR*Tools.

Business Example

Because of a disk crash on a non-mirrored disk containing data files, you cannot start the database. After you replace the disk, you must restore the missing files and recover the database. For this reason, you require the following knowledge:

- An understanding of the problems that may lead to a restore or recovery scenario
- An understanding of how to perform a complete recovery of the database
- An understanding of how to perform a Point-In-Time-Recovery (PITR) of the database
- An understanding of how to perform a disaster restore or recovery of the database

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Perform restore and recovery to the system

Restore and Recovery Tools

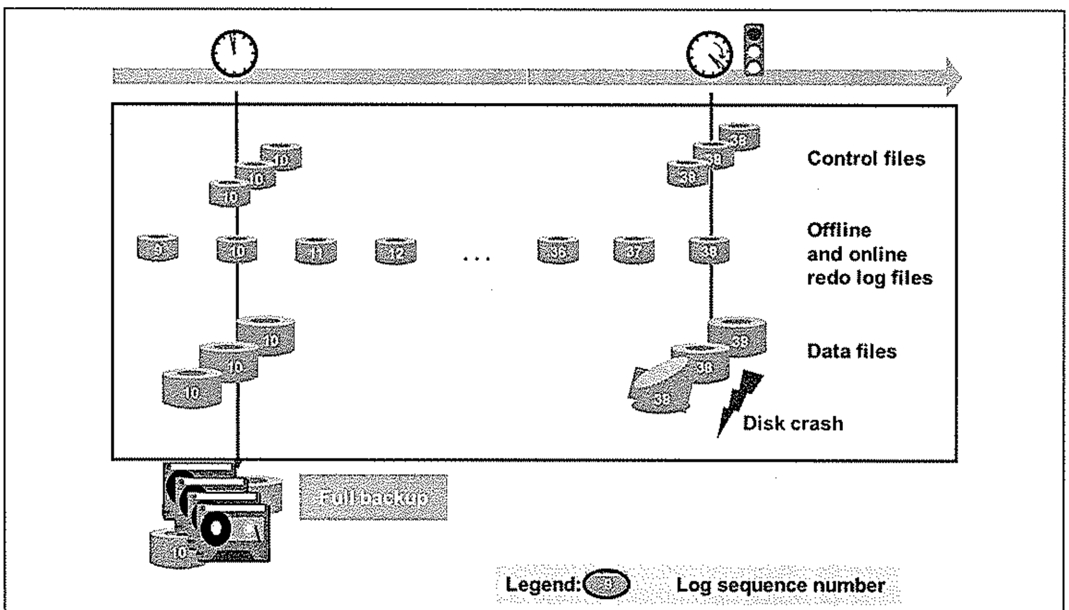


Figure 80: Introduction

In an SAP system with an Oracle database, all data files have online and read or write status. For a functioning and consistent database, all data files and control files must be synchronized; that is, their times must match.

In Oracle, files are synchronized using timestamps. Timestamps are integers that are increased during certain database actions, and entered in all data and control file headers by the log writer or checkpoint process at the checkpoint event.

An example of synchronization data is the log sequence number (LSN), which is increased by 1 during every log switch. At a more sophisticated level, Oracle defines synchronization on the transaction level, using the system change number (SCN), which is increased, for example, after the COMMIT in a modifying transaction or at the checkpoint.

The figure shows an example of a database that was fully saved without errors at time point LSN=10. At time point LSN=38, the database was destroyed by a media or user error in such a way that the database instance failed, or the database became inconsistent. The offline and online redo log files that were created between the beginning of the backup and the occurrence of the error are available. These files are indispensable for re-creating the data in the database.

Problem Solving

If a database problem occurs, you must analyze the problem and create a problem-solving strategy. You must not make any sudden decisions. For typical problem situations, you must have escalation plans ready and tested.

Before restoring any files, you must check for following details:

- The cause of the problem
- The availability of disk space to save and restore files
- The need for a hardware extension
- The file system and mount points
- The availability of backups
- The availability of offline redo log files

To analyze the database problem, check the database alert log and trace files belonging to the background processes in the directory `$ORACLE_HOME/saptrace/background`.

Your problem-solving strategy depends on the answers to the following questions:

- Is the database available?
- Is software or hardware mirroring available?
- Is the error the result of a user error or media error?
- Which files are destroyed?
- Which file types (data files, control files, online redo log files) are affected?

Using the backup strategies recommended by SAP, you have many database backups and offline redo log file backups for a restore and recovery. Your problem-solving strategy determines which backup and offline redo log files are copied back, and how they are applied.

To avoid problems (if you have the time), perform a complete offline backup before the system copies the files back in the restore phase, using BRBACKUP, if the database is running properly, or operating system (OS) backup tools. This backup is important when you perform a PITR or a database reset, because these strategies always involve data loss. In addition, save all offline redo log files in `oraarch` using BRARCHIVE, but do not delete them.

In the event of a hard disk problem, such as a head crash, perform the following actions:

- Replace the hardware
- Create volume(s) on hard disks
- Create file systems and mount them at the old locations

> Caution:
> If you make mistakes, you can drastically aggravate the restore and recovery situation. The costs incurred by a consulting session provided by SAP or an SAP partner are negligible compared to the business consequences of data loss, even for a single day of production operation.

Recovery Report

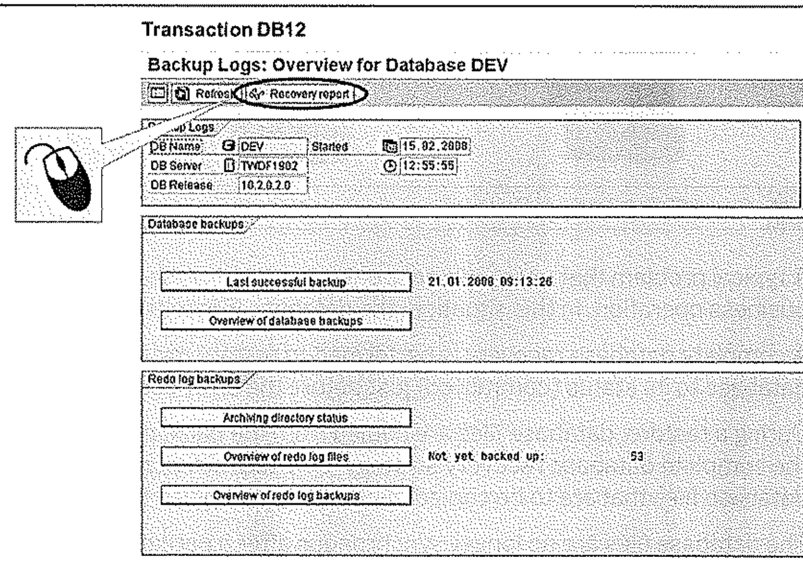


Figure 81: Recovery Report

To test the reliability of your backup strategy, run *recovery report* in SAP transaction `DB12` on a regular basis. The recovery report provides important information that can be used in the event of an Oracle database failure requiring database recovery.

When you start the report, the system displays information about the last successful backup, including backup type and tape names. This information tells you which backup to use for a recovery. The report also checks whether the required redo log files are available (backed up on tape or in the archiving directory). Therefore, you know which files must be restored in the event of a recovery.

Checking the recovery report regularly helps you to detect possible gaps in your backups as follows:

- Missing redo log files can cause problems, because if an error occurs, the database can no longer be restored to the current point in time. You must perform a complete database backup as soon as possible to resolve this critical situation.
- If the list of redo log files is too long, recovery to the current status may take a long time. In such cases, you must perform a full database backup as soon as possible.

Complete Database Recovery

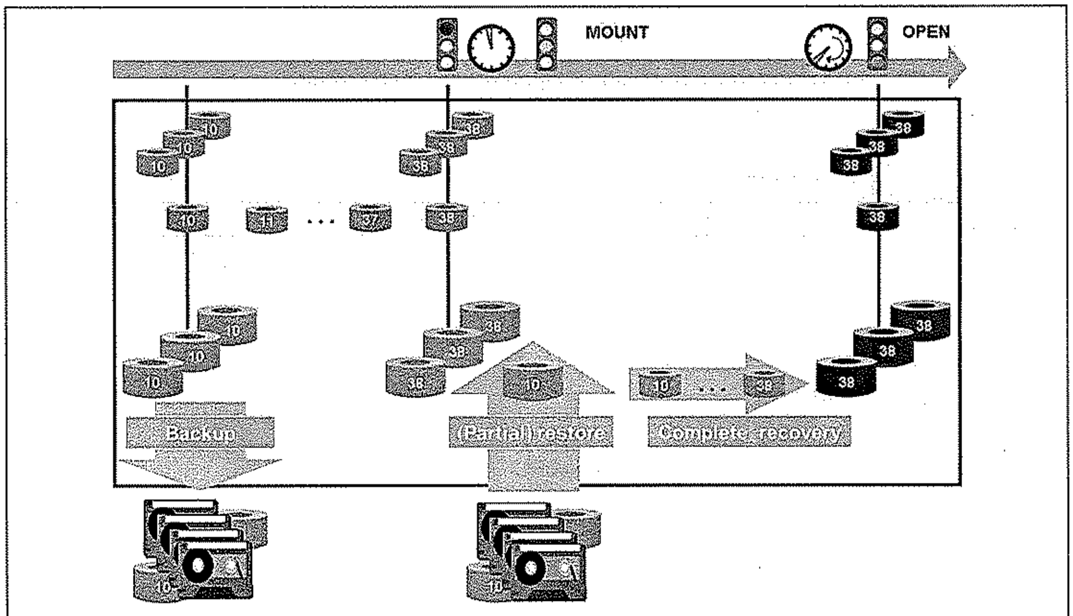


Figure 82: Full Database Recovery

A typical problem that users can encounter is a head crash, in which data is lost during a business operation. In this situation, the database is inconsistent and no longer runs properly.

A complete database recovery is performed to restore missing data files and to recover the database to its status (committed) just before the error occurred.

During a restore, database files are copied from the backup medium back to the disk. Using the complete database recovery strategy, only the minimum required data is copied. The database files that are to be copied back can be combined from different backups.

The database files are no longer synchronous after a partial restore; therefore, the database is inconsistent and cannot run properly after the copy-back procedure terminates.

To synchronize the files, the database evaluates the synchronized data that is saved in the file headers. The database requests all offline redo log files that have accumulated since the oldest database file (in logical terms), in an uninterrupted sequence. During a recovery, all data changes logged by these offline redo log files are replicated in the files that have been copied back from a backup medium.

Using complete database recovery, all changes are performed until all the data files are at the same SCN. This procedure is called media recovery.

When the database is subsequently started up, all open transactions are first rolled forward during the instance recovery. Transactions that are not committed in the redo log are taken (rolled) back, using the undo space (which is likewise recovered).

After the instance recovery, the database shows the following characteristics:

- The database is consistent.
- The database is capable of running.
- The database returns to its committed data status.

Point-in-Time Recovery

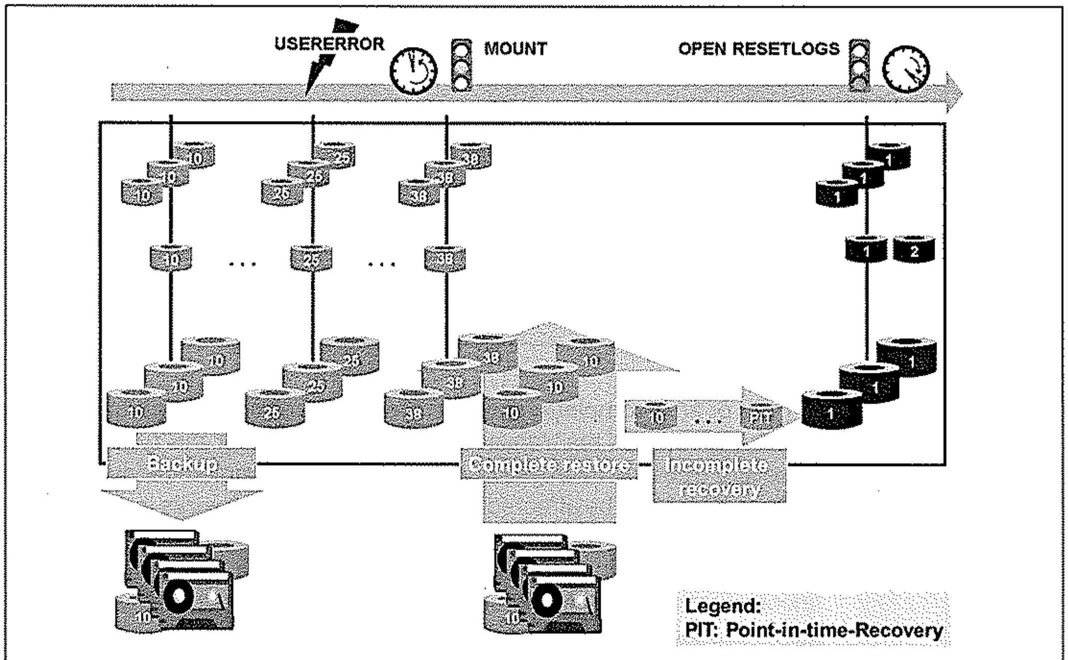


Figure 83: Database Point-in-Time Recovery

A typical problem scenario is that during an upgrade, a user accidently drops a table. As a result, the upgrade must be terminated. A complete backup is available, but it is not created immediately before the upgrade process begins.

A point-in-time recovery (PITR) is performed to reset the database to the status at a certain point in time before the upgrade, using a complete backup, and then recover the data up to a later, appropriate point in time (for example, up to the start of the upgrade or table drop).

Initially, all data files are replaced by copies from a complete online and offline backup (or from a group of partial backups that cover the whole database). The termination point of the recovery determines whether the control files are also to be replaced. The names of all data files and online log files, including their corresponding paths, are in the control file. The file names in the control files must match the file structure after the recovery finishes at the OS level.

During the recovery phase, the changes to the dataset are performed again. Incomplete recovery refers to the end point of recovery, which can be anywhere between the end of the copied backup and the last entry in the current online redo log. The recovery end point can be defined by the redo LSN or SCN, or by specifying a point in time.

A PITR always results in data loss. The data generated between the chosen point in time and the time of last shutdown is lost.

> Caution:
> After a PITR, unless a complete recovery is performed, the database is opened using Oracle command `ALTER DATABASE OPEN RESETLOGS` (called by BRRECOVER), which resets the online redo log files and the LSN to an initial status. Therefore, the old redo logs and the new redo logs do not form a sequence of logs that can be used for a complete recovery, based on the same backup. A complete backup must be performed immediately in a production database before you can use the database.

A PITR can be performed for the whole database or only for a set of tablespaces. For databases that do not have Multiple Components in One Database (MCOD), a database PITR must be performed. Tablespace recovery allows you to restore a tablespace for an individual component in MCOD databases without damaging the remaining components in the database.

> Caution:
> A PITR or an incomplete recovery refers to resetting the database to a previous time. After a PITR, the transactions are consistent for this type of database recovery from the database point of view. An incomplete recovery always causes data to be lost in the affected system. If this system works in a system group with other systems, this also causes data inconsistencies among the systems. Therefore, an incomplete recovery of a database causes application inconsistencies among systems and you cannot use a database to check for or eliminate these inconsistencies.
>
> Due to the consequences described, SAP recommends not to perform an incomplete recovery of a production system. Check whether you can use alternative means, such as those described in SAP Note 434645 – Point-in-time recovery: What must I be aware of? Do not use an incomplete recovery for important systems within a system group unless you are completely clear about how you will work around the inconsistencies at the application level.
>
> Before you perform an incomplete recovery for an important production system, create a support message and clarify the consequences of this action. If you cannot avoid PITR in the production environment, see SAP Note 434647 – Point-in-time recovery in an SAP system group.

Whole Database Reset

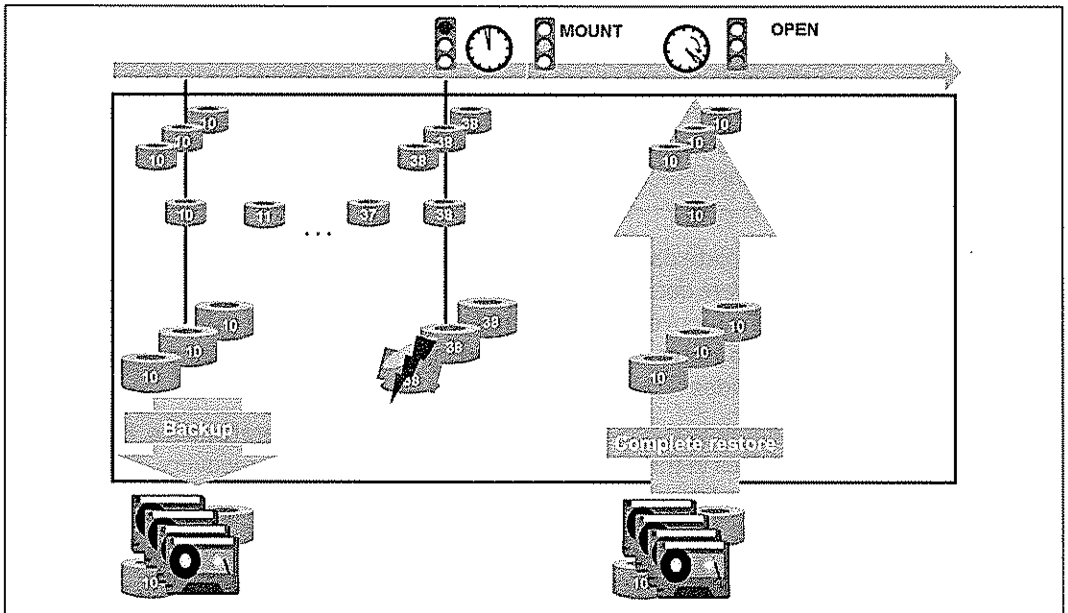


Figure 84: Whole Database Reset

Another typical problem scenario is that during an upgrade, extensive software or hardware problems arise. As a result, the upgrade must be terminated. The database is inconsistent and no longer runs properly. A complete (offline or consistent online) backup is available, created immediately before the upgrade process began.

A whole database reset is performed to reset the database to its status at the end of the complete backup, that is, to the status immediately before the upgrade.

When the database is reset, all data files, online redo log files, and control files are copied from the backup medium. If all these files come from the same valid offline backup, the database is consistent and ready for operation after the copy process finishes. A recovery is not required and the database can be started immediately. If you reset from a consistent online backup, recovery is automatically performed up to the end-point of the backup.

Like a PITR, a database reset always results in data loss. The data that is generated after the applied complete backup is lost. The database, however, remains consistent.

Whole Database Recovery using BR*Tools

When a complete database recovery is performed, the BRRECOVER tool replaces lost data files, using appropriate backups, and recovers the restored data file status using redo log files. To use this function, your online redo log files and control files must be valid.

> Note:
> Since BRRECOVER restores only necessary files, a complete database recovery using BRRECOVER is considered a safe procedure, which means performing this procedure cannot cause more damage to the database by than has already been done. Therefore, you can use this procedure for other problem scenarios, as well; even for scenarios that do not require a restore but only a recovery, such as when a database crashes during an online backup, or when a tablespace goes offline.

To perform a complete database recovery, start BRTOOLS or BRGUI and choose *Restore and recovery → Complete database recovery*. In the menus provided by BRTOOLS, enter the parameters for the recovery. If you do not enter anything here (which is the normal procedure), you can later enter all required information in the list and selection menus that BRRECOVER provides.

The complete database recovery procedure consists of several phases. BRRECOVER presents the phases in the main menu of complete database recovery. These phases must be executed in the predetermined sequence; that is, you can select a particular phase only after the previous phase has been successfully completed.

Complete Database Recovery Using BR*Tools

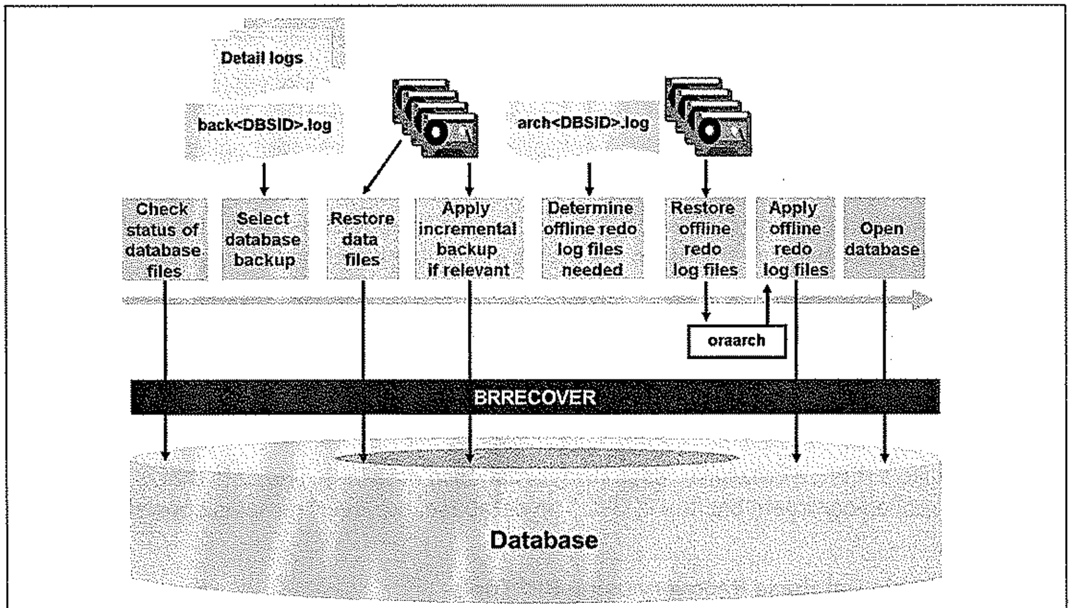


Figure 85: Complete Database Recovery Using BR*Tools

The following control menu displays the phases of a complete database recovery:

```
BR0655I Control menu 101 - please decide how to proceed
------------------------------------------------------------------------
--------
Complete database recovery main menu

 1 = Check the status of database files
 2 * Select database backup
 3 * Restore split/standby control files
```



```
 4 * Restore data files
 5 * Restore split incremental control files
 6 * Restore and apply incremental backup
 7 * Restore and apply archivelog files
 8 * Open database and post-processing
 9 * Exit program
10 - Reset program status

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-----------------------------------------------------------------------
--------
BR0662I Enter your choice:
```

The phases of a whole database recovery are as follows:

1. Check the status of database files

   BRRECOVER checks the status of all files in the database, that is, the control files, online redo log files, and data files.

   BRRECOVER performs the following tasks:

   - Oracle's dynamic V$ views are reloaded during startup to NOMOUNT and MOUNT status. If the database instance is already started, BRRECOVER stops it and then restarts it to MOUNT status.
   - Refers to the entries in some V$ views, such as V$DATAFILE and V$RECOVER_FILE, to determine the status of database files.
   - Logs any errors concerning data files to the detail log created in the `sapbackup` directory. This log gets the `crv` suffix (function ID) for complete recovery.

2. Select database backup

   BRRECOVER determines the eligible backups, using the entries in BRBACKUP summary log file `back<DBSID>.log` (eligible backups are those with return code 0 or 1). The associated detail logs show whether the data files required to restore of missing files are in the backup.

   Missing data files can be restored from various backups during a recovery process. To minimize the subsequent recovery time, BRRECOVER always suggests the most recent backup.

   You can also select an incremental backup to be restored before applying offline redo log files. In this case, BRRECOVER automatically selects the corresponding full backup to restore missing files.

   BRRECOVER also performs a cursory check on the availability of offline redo log files.

3. Restore data files

   BRRECOVER calls BRRESTORE to restore the data files to their original location.

> Caution:
> Neither BRRECOVER nor BRRESTORE creates missing sapdata directories automatically, so you must create them manually at the OS level before you start restoring missing files. However, BRRESTORE automatically creates missing sapdata subdirectories during this phase.

4. Restore and apply incremental backup

   If you select an incremental backup during the Select database backups phase, BRRECOVER calls BRRESTORE to restore and apply the selected incremental backup.

5. Restore and apply archivelog files

   BRRECOVER determines the offline redo log files required for a complete recovery.

   The BRARCHIVE summary log file `arch<DBSID>.log` lists the backups of the offline redo log files.

   BRRECOVER takes into consideration existing offline redo log files in `oraarch` (or `saparch` in older releases), as well as online redo log files. BRRECOVER then calls BRRESTORE to restore the offline redo log files found in backups back to the `oraarch` (or `saparch`) directory.

   Finally, BRRECOVER calls SQL*Plus to apply offline redo log files to the database (Oracle statement RECOVER DATABASE).

   Points to consider while restoring and applying phases are as follows:

   - Offline redo log files are applied to the database in groups of 100 files, at most. If you have more than 100 files to apply, the restore and apply phase repeats automatically, as necessary.
   - The restore and apply phases can be executed in parallel to minimize the total recovery time.

   BRRECOVER can reprocess a structural change in the database, such as an extension of a Tablespace by a new file. Unlike older versions of Oracle, making a new backup after a structural change is not necessary.

6. Open database

   During the last phase, BRRECOVER opens the database and checks the status of the database files and tablespaces.

Database PITR with BR*Tools

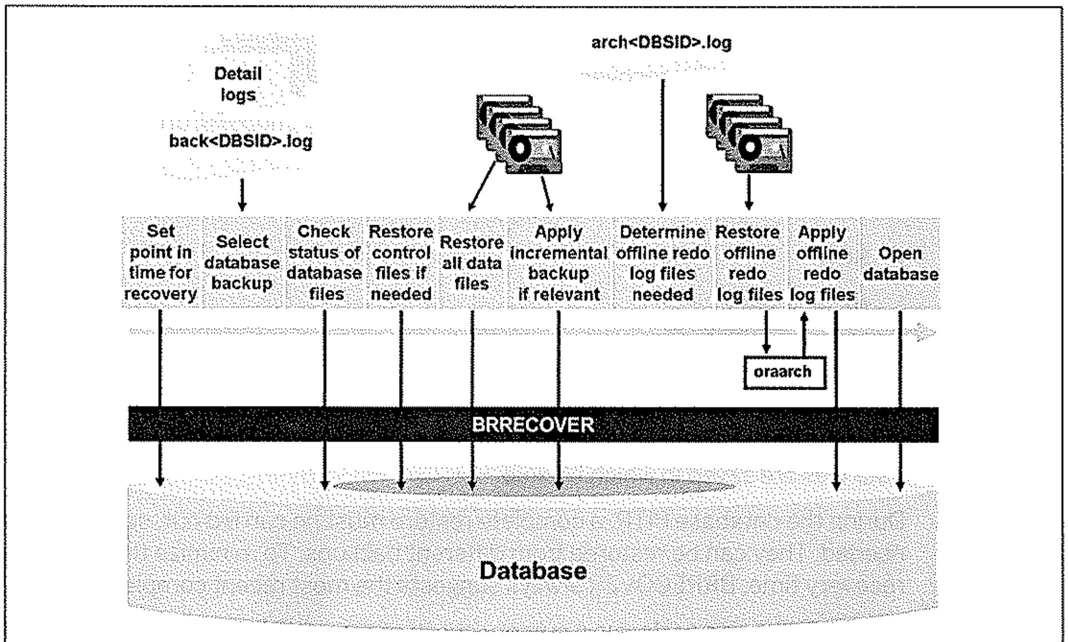


Figure 86: Database PITR with BR*Tools

To perform a PITR, start BRTOOLS or BRGUI and choose *Restore* and *recovery* → *Database point-in-time recovery*. From the subsequent menus provided by BRTOOLS, enter parameters for the database reset. If you do not enter any parameters for the database reset at this point, which is the normal procedure, you can enter all required information through the list and selection menus provided later by BRRECOVER.

The database PITR procedure consists of several phases. BRRECOVER presents the phases in the main menu of PITR, and you must execute them in the predetermined sequence; that is, you can select a particular phase only after the previous phase has been completed successfully.

The following control menu displays the phases of PITR:

```
BR0655I Control menu 103 - please decide how to proceed
-------------------------------------------------------------------------
--------
Database point-in-time recovery main menu

 1 = Set point-in-time for recovery
 2 * Select database backup or flashback
 3 * Check the status of database files
 4 * Restore control files
 5 * Restore data files
 6 * Restore split incremental control files
 7 * Restore and apply incremental backup
 8 * Restore and apply archivelog files
 9 * Restore archivelog files and flashback
10 * Open database and post-processing
11 * Exit program
12 - Reset program status

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-------------------------------------------------------------------------
```

```
--------
BR0662I Enter your choice:
```

Phases of PITR:

1. Set point-in-time for recovery.

   BRRECOVER lets you enter the recovery end point by choosing one of the following processes:

   - Redo LSN
   - SCN
   - Point in time

2. Select database backup.

   BRRECOVER determines the eligible backups using the entries in BRBACKUP summary log file `back<DBSID>.log` (return code 0 or 1). The associated detail logs show which data files are saved in which backup.

   During the database PITR, a complete restore must be carried out, so all data files are needed. They can be restored from different backups. To minimize the subsequent recovery time, BRRECOVER always suggests the most recent complete backup.

   BRRECOVER also performs a cursory check of the availability of offline redo log files.

   You can also select an incremental backup to be restored before applying offline redo log files. In this case, BRRECOVER automatically selects the corresponding full backup to restore all data files.

3. Check the status of database files.

   BRRECOVER checks the status of all files in the database, that is, control files, online redo log files, and data files, to determine which files will be overwritten and which re-created. To update the V$ views, BRRECOVER stops the database instance if it is started and starts it to the MOUNT status if control files are available.

4. Restore control files.

   BRRECOVER calls BRRESTORE to restore control files if needed; that is, if they are unavailable or unsuitable for the selected backups.

5. Restore data files.

   BRRECOVER calls BRRESTORE to restore the data files to their original location.

6. Restore and apply incremental backup.

   If you select an incremental backup during the Select database backups phase, BRRECOVER calls BRRESTORE to restore and apply the selected incremental backup.

7. Restore and apply archivelog files.

   BRRECOVER determines the offline redo log files required for the recovery up to the indicated time point. The BRARCHIVE summary log file `arch<DBSID>.log` lists the backups of the offline redo log files. BRRECOVER takes into consideration existing offline redo log files in `oraarch` (or `saparch`), as well as online redo log files.

   BRRECOVER then calls BRRESTORE to restore the offline redo log files that are found in backups back to the `oraarch` (or `saparch`) directory.

Finally, BRRECOVER calls SQL*Plus to apply redo log files to the database (Oracle statement RECOVER DATABASE UNTIL redo log files are completely applied).

8. Open database

   During the last phase, BRRECOVER performs the following tasks:

   a. Opens the database using the RESETLOGS option (this is required because of incomplete recovery)

   b. Creates missing temporary files

   c. Checks the status of database files and tablespaces

   d. Deletes unnecessary files that are no longer used by the database

Whole Database Reset with BR*Tools

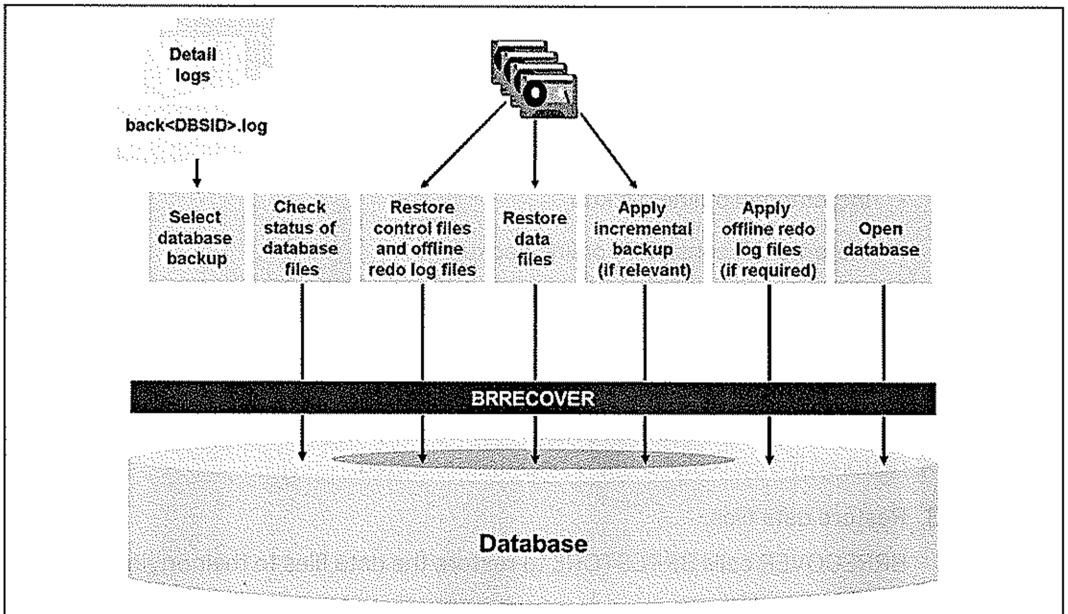


Figure 87: Whole Database Reset with BR*Tools

To perform a whole database recovery, start BRTOOLS or BRGUI and choose *Restore and recovery* → *Whole database reset*. The subsequent menus provided by BRTOOLS, enter the parameters for the database reset. If you do not enter any parameters for database reset at this point, which is the normal procedure, you can enter all required information through the list and selection menus provided later by BRRECOVER.

The whole database reset procedure consists of several phases. BRRECOVER presents the phases in the main menu, and they must be executed in the predetermined sequence, that is, a particular phase can only be selected after the previous phase is completed successfully.

The following control menu displays the phases of a whole database reset:

```
BR0655I Control menu 109 - please decide how to proceed
-----------------------------------------------------------------------
--------
Whole database reset main menu

1 = Select database backup or restore point
2 * Check the status of database files
```

```
3 * Restore control files and redolog files
4 * Restore data files
5 * Restore and apply incremental backup
6 * Apply archivelog files
7 * Restore archivelog files and flashback
8 * Open database and post-processing
9 * Exit program
10 - Reset program status

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
------------------------------------------------------------------------
--------
BR0662I Enter your choice:
```

A BRRECOVER whole database reset goes through the following steps:

1. Select consistent database backup.

   BRRECOVER determines the most suitable backups using entries in BRBACKUP summary log `back<DBSID>.log` (return code 0 or 1) in BRBACKUP. You can select the following backup types:

   - Whole offline backup
   - Whole consistent online backup
   - Incremental offline backup
   - Incremental consistent online backup

   If you choose an incremental backup, BRRECOVER automatically selects the corresponding full backup to restore all data files.

2. Restore control files and redo log files.

   BRRECOVER calls BRRESTORE to restore control files. Offline redo log files are also restored if a consistent online backup is selected.

3. Restore data files.

   BRRECOVER calls BRRESTORE to restore the data files to their original location.

4. Apply incremental backup.

   If you select an incremental backup during the Select database backup phase, BRRECOVER calls BRRESTORE to restore and apply the selected incremental backup.

5. Apply archivelog files.

   If a consistent online backup is selected, BRRECOVER calls SQL*Plus to apply the restored offline redo log files to the database.

6. Open the database.

   During this phase, BRRECOVER performs the following tasks:

   a. Opens the database (if necessary, using the RESETLOGS option)

   b. Creates missing temporary files

   c. Checks the status of database files and tablespaces

   d. Deletes unnecessary files that are no longer used by the database

Disaster Recovery with BR*Tools

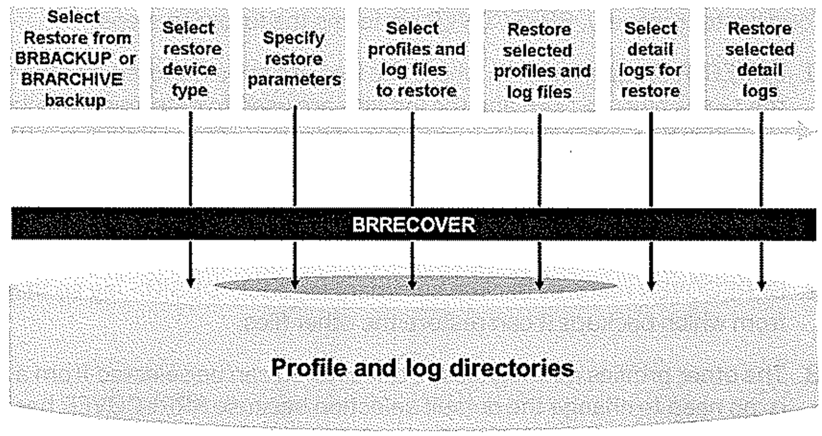


Figure 88: Disaster Recovery Using BR*Tools

If you lose your entire Oracle database system, including the hardware, and have not taken any special security precautions, such as setting up an Oracle standby database, you must recover the whole system, step by step. The restore and recovery procedures depend on the existence of configuration profiles and backup log files. Therefore, a special procedure called disaster recovery is embedded in BRRECOVER. This disaster recovery procedure enables you to restore these files when they are missing.

> Note:
> Disaster recovery is only a preparation step for a subsequent database recovery using database PITR or whole database reset.

The prerequisites for using disaster recovery are as follows:

- SAP and Oracle software must be correctly installed.
- File systems with the `sapdata` directories must exist and be configured as before the disaster.

To perform a disaster database recovery, start BRTOOLS or BRGUI and choose *Restore* and *recovery* → *Disaster recovery*. From the subsequent menus provided by BRTOOLS, enter parameters for the database reset. If you do not enter any parameters for the database reset at this point, which is the normal procedure, you can enter all required information through the list and selection menus provided later by BRRECOVER.

The following control menu displays the phases of disaster recovery:

```
BR0656I Choice menu 136 - please make a selection
-----------------------------------------------------------------------
--------
Disaster recovery main menu

 1 = Restore profiles and log files from BRBACKUP backup
 2 - Restore profiles and log files from BRARCHIVE backup
 3 * Exit program
 4 - Reset program status

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
```

```
----------------------------------------------------------------------
--------
BR0662I Enter your choice:
```

For disaster recovery, BRRECOVER performs the following steps:

1. BRRECOVER queries from where the files are to be restored. In other recovery scenarios, the `init<DBSID>.sap` and the log files are used to determine which data is to be restored. In the disaster recovery scenario, these files are missing, therefore, BRRECOVER must know where to find these files. The `init<DBSID>.sap` and log files are located in either the latest BRBACKUP or BRARCHIVE backup. These options are found in the disaster recovery main menu.
2. BRRECOVER restores backup summary log back<DBSID>.log so that the system knows from which backups it can restore the other files.
3. The other profiles and log files to be restored can be selected from a list. Normally, there is no need to change the default selection because BRRECOVER has already determined which files need to be restored.
4. In the *Restore of BRBACKUP detail logs* selection, a list of backup detail logs is displayed to be selected for restore. Choose only the detail logs of the backup or backups you need to restore the database later, because for every selected backup, the corresponding backup must be mounted to restore the detail log.

Other Functions of BRRECOVER

When calling BRTOOLS or BRGUI and choosing Restore and recovery, two additional backup and restore functions are displayed.

Whole database reset

Point-in-Time Recovery

Disaster recovery

> Caution:
> Use these functions carefully and only if you fully understand the procedure. These functions are expert functions, which are to be used only in exceptional cases.

The other functions of BRRECOVER are as follows:

- *Restore of individual backup files*

  Use this function to restore individual files from a backup, for example, to perform a manual restore and manual recovery using the *Restore and application of archivelog files* function.

- *Restore and application of archivelog files*

  Use this function to perform a manual recovery, for example, after performing a manual restore using the *Restore of individual backup files* function.

> Hint:
> For any other scenarios, which cannot be resolved using BRRECOVER, use the procedures described in SAP Library – SAP Database Guide: Oracle.

Oracle Flashback Database

Oracle flashback database provides an alternative to the normal scenario PITR to reset the database to the status at a certain point in time. Oracle flashback database is a new Oracle feature as of Oracle Database Release 10g. Flashback database is part of the Oracle flashback technology.

Flashback database can be performed by choosing the rewind button on a tape recorder. Flashback database allows a time reset of the database without a time-consuming reload of all database files. The duration of the rewind of the database using flashback database is independent of the size of the database.

The status of the database after a flashback database operation is identical to a PITR of the entire database.

A normal PITR consists of the following steps:

1. The reload of an entire database backup
2. The import of archive logs or recovery up until the required point in time.

During a flashback database operation, however, the last changes, starting from the current state of the database, are reversed.

Points to consider during flashback are as follows:

- Reverting the database to an earlier state is necessary in the event of logical data corruption resulting from an application error, user error, or administrator error. In such cases, activate normal flashback logging.
- Flashback database also lends itself to cases where you anticipate a database reset, for example, if the import of an SAP Support Package or an SAP or Oracle upgrade to a new patch set fails, in which case the database must be reset so that you can repeat the process.

  In this case, it makes sense to define one or more guaranteed restore points, but no normal flashback logging.

Depending on the situation, you must either activate normal flashback logging only, or create guaranteed restore points only. Because of space restrictions, and for performance reasons, make sure that both variants are not be active at the same time.

The following table compares normal PITR to flashback database:

PITR	Flashback Database
You must reload data prior to a backup using PITR.	Reloading data prior to a backup is not necessary when using flashback database.
The duration of a PITR depends largely on the time required to reload all database files from the backup, and on the duration of the subsequent recovery when using archive logs.	With flashback database, the duration is proportional to the time to which the database is reset. The effort is proportional to the amount of changes made during this time that now must be reversed. As a general rule, it takes approximately the same amount of time to undo the changes in the database by reversing the time that it took to make the changes in the first place (for example, a 1 h flashback database takes approximately 1 h).

PITR	Flashback Database
With a PITR, you must execute a new PITR in cases where you did not select the correct point-in-time, combined with another complete reload of the backup.	You can use flashback database to check whether you have reset the database to the correct point in time. If you did not select the correct point in time, you can repeat flashback database with earlier or later times until you find the correct point in time.

Architecture of Flashback Database

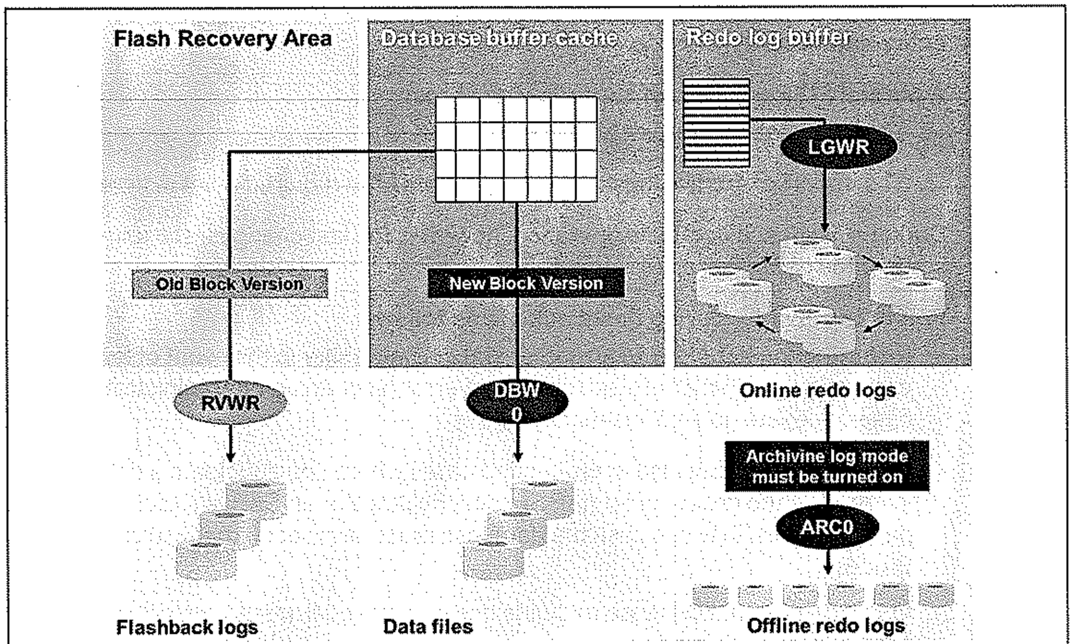


Figure 89: Flashback Database

The flashback database is based on a new type of log files, known as flashback logs. When flashback logging is active, flashback logs of the RVWR Oracle background process are exclusively written to flash recovery area during operation. There, the logs are automatically administrated by Oracle.

Flashback logging is deactivated by default.

When flashback logging is active, Oracle regularly writes all blocks of database files that were changed to flashback logs.

With a flashback database, the data blocks that were changed since the time set for the flashback are read from flashback logs and written back to the database file. After that, the changes that occurred in the database between the time the block was copied to flashback log and the required recovery target time are also implemented using archive logs.

Therefore, flashback database not only requires flashback logs but also the relevant archive logs for the desired period of time (flashback retention target). Therefore, the database must run in the ARCHIVELOG mode (this is always the case for SAP databases).

If the RVWR background process detects problems when writing flashback logs, for example, if flashback log cannot be written correctly for a guaranteed restore point (see below), the Oracle instance terminates. This behavior becomes clear when you consider the purpose of guaranteed restore points and the possible usage scenarios. For flashback logging activated in the usual way, an input/output problem of the RVWR results in the deactivation of flashback logging, but the instance keeps running (standby database is an exception).

During FLASHBACK DATABASE, only the database files are returned to an earlier state, not the auxiliary files such as Oracle password files, Oracle profile files, Oracle wallet files, or Oracle control files.

Flashback logs are not backed up. Therefore, you do not need to adjust the database backup strategy to also back up flashback logs. Flashback logs are deleted automatically from flash recovery area by the Oracle server when they are no longer needed, or if space is required in flash recovery area for more important files.

Flashback logging generates an additional 2% of input/output load. To keep the additional input/output load generated by flashback database as low as possible, changed blocks from the Oracle buffer cache are written to the flashback logs in certain, fixed intervals. Therefore, not every version of a data block is written to flashback logs.

Restrictions

Consider the following restrictions when using flashback database:

- Flashback database is not a substitute for database backups.
- To use flashback database, the database must be physically intact. If there are corrupt or missing database files, flashback database cannot revert the database to an intact state.
- Flashback database can only undo changes in Oracle database files. Flashback database does not reset profile files (`spfile`, `init<DBSID>.ora`), password files, and Net Services configuration files (such as `sqlnet.ora`).
- Certain database operations cannot be undone using flashback database. Some of these operations are as follows:
  - Datafile Shrink
  - Drop Tablespace
  - Drop Datafile

  When the operations are complete, the flashback database time window restarts.

Administration

The following administrative tasks belong to flashback database:

- Creates the flash recovery area
- Activates and deactivates normal flashback logging
- Creates and deletes guaranteed restore points (GRPs)
- Monitors performance
- Monitors the free space in the flash recovery area

Note:
For more information about flashback database, see SAP Note 966117 – Oracle Flashback Database technology. SAP Note 966073 – Oracle Flash Recovery Area/ Fast Recovery Area describes the configuration and administration of the flash recovery area for use with flashback databases with SAP BR*Tools.

The support of SAP BR*Tools for flashback database includes the activation and deactivation of database flashback and the management of restore points. It also includes enhancements in PITR of the database and database reset. For more information, see SAP Note 1125923 – Support for Oracle database flashback in BR*Tools.

Unit 2
Exercise 8

Perform Restore and Recovery to the System

Business Example

A disk has become unusable. You want to restore and recover the database to its most recent state.

In this exercise, various restore and recovery scenarios are performed. Before deleting any files to perform the exercises, make sure that you have proper offline and online backup, and the database runs in the archive log mode.

Windows does not allow you to delete a file in use. Therefore, you must shut down Oracle before deleting a file to simulate a disk crash.

1. Simulate a disk crash by shutting down your database and deleting data file `D:\oracle\<DBSID>\sapdata3\<DBSID>_1\<DBSID>.data1`.

   Start the database and check the error message. Decide which scenario you want to use to recover the database and perform the recovery as follows:

   After performing the recovery, check that the database is recovered successfully.

2. To simulate a user error, start the script `usererror.bat`, located in `G:\oracle\<DBSID>\scripts`. Note the current system time that the script displays. After waiting for 60 seconds, the script displays the number of rows of the table *DBCHECKORA* and drops *DBCHECKORA*. To show that *DBCHECKORA* is dropped, execute the command to show the number of rows in *DBCHECKORA*. An error message, stating that *DBCHECKORA* does not exist, appears.

   Which is the correct scenario to recover the database? Perform the recovery.

   After performing the recovery, check that the database is recovered successfully as follows:

3. (Optional) Repeat the last exercise. Recovery is not possible until the point in time of the user error. Why?

4. You have possibly corrupted the database; therefore, you decide to reset the database to the point in time of the offline backup taken in the last exercise.

   Perform the correct scenario.

5. (Optional) Simulate restore and recovery after a disk crash without the BRBACKUP logs.

   To do this, shut down your database and delete data file `D:\oracle\<DBSID>\sapdata3\<DBSID>_1\<DBSID>.data1` and the BRBACKUP logs in directory `D:\oracle\<DBSID>\sapbackup`.

   Then, start the database and check the error message. Decide which scenario you want to use to recover the database and perform the recovery.

Hint:
Before starting with the simulation of this error scenario, perform a complete offline backup.

After performing the recovery, check that the database is recovered successfully as follows:

Unit 2
Solution 8

Perform Restore and Recovery to the System

Business Example

A disk has become unusable. You want to restore and recover the database to its most recent state.

In this exercise, various restore and recovery scenarios are performed. Before deleting any files to perform the exercises, make sure that you have proper offline and online backup, and the database runs in the archive log mode.

Windows does not allow you to delete a file in use. Therefore, you must shut down Oracle before deleting a file to simulate a disk crash.

1. Simulate a disk crash by shutting down your database and deleting data file D:\oracle\<DBSID>\sapdata3\<DBSID>_1\<DBSID>.data1.

   Start the database and check the error message. Decide which scenario you want to use to recover the database and perform the recovery as follows:

   After performing the recovery, check that the database is recovered successfully.

   a) Shut down the database using, for example, brspace -c force -f dbshut.

      Delete data file

      D:\oracle\<DBSID>> del sapdata3\<DBSID>_1\<DBSID>.data1.

   b) Start the database using, for example, brspace -c force -f dbstart.

      The following error message appears:

```
BR0613I Database instance T99 is shut down
BR0786I Database instance T99 will be opened now in mode 'normal'

BR0280I BRSPACE time stamp: 2011-06-01 14.56.32
BR0304I Starting and opening database instance T99 ...
BR0278E Command output of 'D:\oracle\DEV\112\BIN\sqlplus /nolog
&lt; D:\ORACLE\T99\sapreorg\.sefzyyvl.spi':

SQL*Plus: Release 11.2.0.1.0 Production on Mi Jun 1 14:56:32 2011

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

SQL> SQL> SQL> Connected to an idle instance.
SQL>

ORACLE instance started.

Total System Global Area  133644288 bytes
Fixed Size                  2173360 bytes
Variable Size             109053520 bytes
Database Buffers           16777216 bytes
Redo Buffers                5640192 bytes
Database mounted.
```

```
ORA-01157: cannot identify/lock data file 4 - see DBWR trace file
ORA-01110: data file 4: 'D:\ORACLE\T99\SAPDATA3\T99_1\T99.DATA1'

SQL> Disconnected from Oracle Database 11g Enterprise Edition
Release
11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options
BR0280I BRSPACE time stamp: 2011-06-01 14.56.39
BR0279E Return code from 'D:\oracle\DEV\112\BIN\sqlplus /nolog
&lt; D:\ORACLE\T99\sapreorg\.sefzyyvl.spi': 0
BR0302E SQLPLUS call for database instance T99 failed
BR0306E Start and open of database instance T99 failed

BR0280I BRSPACE time stamp: 2011-06-01 14.56.39
BR0669E Cannot continue due to previous warnings or errors

BR0280I BRSPACE time stamp: 2011-06-01 14.56.39
BR0700E Fatal errors occurred - terminating processing...
BR1018I Number of instances processed: 0
BR1004E BRSPACE function 'dbstart' failed

BR1008I End of BRSPACE processing: sefzyyvl.dbr 2011-06-01
14.56.39
BR0280I BRSPACE time stamp: 2011-06-01 14.56.40
BR1007I BRSPACE terminated with errors
```

c) The correct scenario to recover the database is complete database recovery.

   a) To recover the database, start BRGUI or BRTOOLS and choose *Restore and recovery → Complete database recovery*.

   b) Choose *Continue*.

      In *Complete database recovery main menu*, check the actions provided by BRRECOVER.

      The following menu appears:

```
Complete database recovery main menu

 1 = Check the status of database files
 2 * Select database backup
 3 * Restore split/standby control files
 4 * Restore data files
 5 * Restore split incremental control files
 6 * Restore and apply incremental backup
 7 * Restore and apply archivelog files
 8 * Open database and post-processing
 9 * Exit program
10 - Reset program status
```

      Perform the necessary steps to recover the database.

      Choose *Continue* at any prompt to successfully complete the recovery without any additional input.

      In the last step, you must restart the database.

d) Check that the database is recovered successfully by performing the following steps:

a) Check the status by starting BRGUI or BRTOOLS and choose *Instance Management → Show instance status*.

b) Choose *Continue* twice.

In the *Information about the status of database instance T99* menu, check the *Instance status (status)* parameter.

2. To simulate a user error, start the script `usererror.bat`, located in `G:\oracle\<DBSID>\scripts`. Note the current system time that the script displays. After waiting for 60 seconds, the script displays the number of rows of the table *DBCHECKORA* and drops *DBCHECKORA*. To show that *DBCHECKORA* is dropped, execute the command to show the number of rows in *DBCHECKORA*. An error message, stating that *DBCHECKORA* does not exist, appears.

   Which is the correct scenario to recover the database? Perform the recovery.

   After performing the recovery, check that the database is recovered successfully as follows:

   a) Start the script `usererror.bat` to delete the *DBCHECKORA* table.

   The following error message appears:

```
D:\oracle\T99>cd scripts

D:\oracle\T99\scripts>usererror.bat
This script will first display the number of table entries in
table DBCHECKORA.
Then it will drop DBCHECKORA - accessing the table after the drop
will display an error message.
For point-in-time recovery make a note of the current time which
is:
15:19
Script continues in 60 seconds - please wait.

SQL*Plus: Release 11.2.0.1.0 Production on Mi Jun 1 15:20:49 2011

Copyright (c) 1982, 2010, Oracle.  All rights reserved.


  COUNT(*)
----------
       123

select count(*) from DBCHECKORA
                     *
ERROR at line 1:
ORA-00942: table or view does not exist


Disconnected from Oracle Database 11g Enterprise Edition Release
11.2.0.1.0 - 64
bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

D:\oracle\T99\scripts>
```

   b) The correct recovery procedure is to perform a database PITR as follows:

a) Start BRGUI or BRTOOLS and choose *Restore and recovery → Database point-in-time recovery*. In *Database point-in-time recovery main menu*, check the actions provided by BRRECOVER.

The following menu appears:

```
Database point-in-time recovery main menu

 1 = Set point-in-time for recovery
 2 * Select database backup or flashback
 3 * Check the status of database files
 4 * Restore control files
 5 * Restore data files
 6 * Restore split incremental control files
 7 * Restore and apply incremental backup
 8 * Restore and apply archivelog files
 9 * Restore archivelog files and flashback
10 * Open database and post-processing
11 * Exit program
12 - Reset program status
```

b) Choose *Continue* until BRRECOVER input menu *Options for point-in-time recovery of database <DBSID>* appears.

```
Options for point-in-time recovery of database T99

 1 # Database instance of archivelog thread (instance) . []
 2 ~ Last archivelog sequence to apply (last_seq) ...... []
 3 ~ Last system change number to apply (last_scn) ..... []
 4 ~ End point-in-time for recovery (end_pit) .......... []
```

c) In the parameter *End point-in-time for recovery (end_pit)*, enter the date and time that you noted in step 2.a).

c) In the *Select database backup or flashback* menu, choose a backup for database PITR.

Note the actions performed by BRRECOVER. Choose *Continue* at any prompt to successfully complete the recovery without any additional input.

d) To check whether the table *DBCHECKORA* is back, run script `checktable.bat`.

The following message appears:

```
D:\oracle\T99\scripts>checktable.bat
This script will display the number of table entries in
table DBCHECKORA.

SQL*Plus: Release 11.2.0.1.0 Production on Mi Jun 1 15:33:57 2011

Copyright (c) 1982, 2010, Oracle.  All rights reserved.


  COUNT(*)
----------
       123

Disconnected from Oracle Database 11g Enterprise Edition Release
11.2.0.1.0 - 64
bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

D:\oracle\T99\scripts>
```


e) To verify that an `alter database open resetlogs` is performed, check the status of the database instance by performing the following steps:

   a) To check the status, start BRGUI or BRTOOLS and choose *Instance Management → Show instance status*.

   b) Choose *Continue* two times.

      In the *Information about the status of database instance T99* menu, check that the option*Current redolog sequence (redoseq)* is set to 1.

3. (Optional) Repeat the last exercise. Recovery is not possible until the point in time of the user error. Why?

   a) Between the latest backup and the point in time to which you want to recover the database, a database reset is performed by the previous PITR. It is not possible to recover a database over a database reset. Therefore, you must create a complete backup immediately after any PITR (or whole database reset).

4. You have possibly corrupted the database; therefore, you decide to reset the database to the point in time of the offline backup taken in the last exercise.

   Perform the correct scenario.

   a) To reset the database to the point in time of a backup, offline backup or consistent online backup, use the whole database reset scenario as follows:

      a) Start BRGUI or BRTOOLS and choose *Restore and recovery → Whole database reset*.

         In Whole database reset main menu, check the actions provided by BRRECOVER.

         The following menu appears:

```
Whole database reset main menu

 1 = Select database backup or restore point
 2 * Check the status of database files
 3 * Restore control files and redolog files
 4 * Restore data files
 5 * Restore and apply incremental backup
 6 * Apply archivelog files
 7 * Restore archivelog files and flashback
 8 * Open database and post-processing
 9 * Exit program
10 - Reset program status
```

   b) Choose *Continue* until the list of backups suitable for the whole database reset appears. Choose a backup. Watch the actions performed by BRRECOVER. Choose *Continue* at any prompt to successfully complete the recovery without any additional input.

5. (Optional) Simulate restore and recovery after a disk crash without the BRBACKUP logs.

   To do this, shut down your database and delete data file `D:\oracle\<DBSID>\sapdata3\<DBSID>_1\<DBSID>.data1` and the BRBACKUP logs in directory `D:\oracle\<DBSID>\sapbackup`.

   Then, start the database and check the error message. Decide which scenario you want to use to recover the database and perform the recovery.

> **Hint:**
> Before starting with the simulation of this error scenario, perform a complete offline backup.

After performing the recovery, check that the database is recovered successfully as follows:

a) Shut down the database using, for example, `brspace -c force -f dbshut`.

   Delete data file

   `D:\oracle\<DBSID>> del sapdata3\<DBSID>_1\<DBSID>.data1.`

b) Start the database using, for example, `brspace -c force -f dbstart`.

   The following error message appears:

```
BR0613I Database instance T99 is shut down
BR0786I Database instance T99 will be opened now in mode 'normal'

BR0280I BRSPACE time stamp: 2011-06-01 14.56.32
BR0304I Starting and opening database instance T99 ...
BR0278E Command output of 'D:\oracle\DEV\112\BIN\sqlplus /nolog
&lt; D:\ORACLE\T99\sapreorg\.sefzyyvl.spi':

SQL*Plus: Release 11.2.0.1.0 Production on Mi Jun 1 14:56:32 2011

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

SQL> SQL> SQL> Connected to an idle instance.
SQL>

ORACLE instance started.

Total System Global Area  133644288 bytes
Fixed Size                  2173360 bytes
Variable Size             109053520 bytes
Database Buffers           16777216 bytes
Redo Buffers                5640192 bytes
Database mounted.

ORA-01157: cannot identify/lock data file 4 - see DBWR trace file
ORA-01110: data file 4: 'D:\ORACLE\T99\SAPDATA3\T99_1\T99.DATA1'


SQL> Disconnected from Oracle Database 11g Enterprise Edition
Release
11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options
BR0280I BRSPACE time stamp: 2011-06-01 14.56.39
BR0279E Return code from 'D:\oracle\DEV\112\BIN\sqlplus /nolog
&lt; D:\ORACLE\T99\sapreorg\.sefzyyvl.spi': 0
BR0302E SQLPLUS call for database instance T99 failed
BR0306E Start and open of database instance T99 failed

BR0280I BRSPACE time stamp: 2011-06-01 14.56.39
BR0669E Cannot continue due to previous warnings or errors

BR0280I BRSPACE time stamp: 2011-06-01 14.56.39
BR0700E Fatal errors occurred - terminating processing...
```

```
BR1018I Number of instances processed: 0
BR1004E BRSPACE function 'dbstart' failed

BR1008I End of BRSPACE processing: sefzyyvl.dbr 2011-06-01
14.56.39
BR0280I BRSPACE time stamp: 2011-06-01 14.56.40
BR1007I BRSPACE terminated with errors
```

c) The correct scenario to recover the database is complete database recovery as follows:

   a) To recover the database, start BRGUI or BRTOOLS and choose *Restore and recovery → Complete database recovery*.

   b) Choose *Continue*.

      In *Complete database recovery main menu*, check the actions provided by BRRECOVER.

      The following menu appears:

```
Complete database recovery main menu

 1 = Check the status of database files
 2 * Select database backup
 3 * Restore split/standby control files
 4 * Restore data files
 5 * Restore split incremental control files
 6 * Restore and apply incremental backup
 7 * Restore and apply archivelog files
 8 * Open database and post-processing
 9 * Exit program
10 - Reset program status
```

      Perform the necessary steps to recover the database.

      The recovery interrupts when performing the step *Select database backup*.

```
BR0699I Reading log file D:\ORACLE\T99\sapbackup
\backT99.log ...
BR0252E Function fopen() failed for 'D:\ORACLE\T99\sapbackup
\backT99.log' at loc
ation BrbRunGet-2
BR0253E errno 2: No such file or directory
BR0121E Processing of log file D:\ORACLE\T99\sapbackup
\backT99.log failed

BR0280I BRRECOVER time stamp: 2011-06-01 16.21.03
BR0700E Fatal errors occurred - terminating processing...
BR0715E Complete database recovery recovery failed

BR0797I Number of restored/recovered database files: 0/0
BR0798I Number of restored/applied incremental files: 0/0
BR0799I Number of restored/applied archivelog files: 0/0

BR0706I End of database recovery: vefzzgeu.crv 2011-06-01
16.21.03
BR0280I BRRECOVER time stamp: 2011-06-01 16.21.04
BR0704I BRRECOVER terminated with errors

######################################################################
```

```
################
BR0292I Execution of BRRECOVER finished with return code 5
```

The error message indicates that the processing of the file backT99.log, which contains the summary log information, failed.

BRBACKUP needs the log information to guide you through the recovery.

Restore the BRBACKUP logs to proceed with the complete database recovery.

d) The correct scenario to recover the BRBACKUP logs is disaster recovery, as follows:

a) To recover the database, start BRGUI or BRTOOLS and choose *Restore and recovery → Disaster recovery*.

b) Choose *Continue*.

In *Disaster recovery main menu*, check the actions provided by BRRECOVER.

The following list appears:

```
Disaster recovery main menu

 1 = Restore profiles and log files from BRBACKUP backup
 2 - Restore profiles and log files from BRARCHIVE backup
 3 * Exit program
 4 - Reset program status
```

c) Choose *Restore profiles and log files from BRBACKUP backup*.

d) Choose *Local disk*.

e) Proceed the menu *Parameters for restoring profiles and log files from local BRBACKUP disk* by choosing *Continue*.

Choose *Continue* at any prompt to successfully complete the recovery without any additional input.

In the last step, exit BRRECOVER.

e) Proceed with the complete database recovery as follows:

a) Start BRGUI or BRTOOLS and choose *Restore and recovery → Complete database recovery*.

b) Choose *Continue*. In *Complete database recovery main menu*, check the actions provided by BRRECOVER.

The following menu appears:

```
Complete database recovery main menu

 1 = Check the status of database files
 2 * Select database backup
 3 * Restore split/standby control files
 4 * Restore data files
 5 * Restore split incremental control files
 6 * Restore and apply incremental backup
 7 * Restore and apply archivelog files
 8 * Open database and post-processing
```



```
 9 * Exit program
10 - Reset program status
```

Perform the necessary steps to recover the database.

Choose *Continue* at any prompt to successfully complete the recovery without any additional input.

In the last step, restart the database.

f) Check that the database is recovered successfully as follows:

a) To check the status, start BRGUI or BRTOOLS and choose *Instance Management → Show instance status*.

b) Choose *Continue* two times.

In the menu *Information about the status of database instance T99*, check the parameter *Instance status (status)*.

LESSON SUMMARY

You should now be able to:

- Perform restore and recovery to the system

Unit 2
Lesson 5

Working with Advanced Backup Techniques

LESSON OVERVIEW
In this lesson, you will learn about advanced backup scenarios.

Business Example

Your database contains 900 GB of data. Because your users are working on the SAP system 24 hours per day from different countries, you are considering an alternative to normal backups that will not interrupt or slow down normal operation. For this reason, you require the following knowledge:

- An understanding about the various backup strategies supported by SAP
- An understanding of how to decide which strategy fits your needs

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Apply advanced backup techniques

Advanced Backup Techniques

This lesson provides an overview of methods for reducing the length of the backup, restore and recovery processes, and describes how to ensure that backups affect live operation as little as possible. These topics are particularly relevant in systems that require high availability, or have only small windows for downtime.

Consider the following high availability problem that can occur when developing a strategy for a backup.

You want to perform an offline backup for a large system, but your Oracle database and SAP system must remain online for most of the required backup time.

There are some advanced backup techniques that can help to solve this availability problem. These techniques also provide additional security in case of a hardware failure.

These improvements will add higher costs in the following areas:

- Hardware
- Training of the administrator(s)
- Additional administration work required for the implementation and production operation

Methods for Accelerating the Backup and Restore Process

Various possibilities for optimizing backup time are illustrated as follows:

- Hardware

Hardware used during the backup (tape drives, disks, system, and input/output buses) play a key role in the data throughput during a backup.

- Parallel backup

  Using several tape drives in parallel greatly reduces backup time.

- Using DD

  If you use BRBACKUP to save a database, do not use CPIO to copy data files. Instead, use DD, which offers better performance. If you use DD to copy data, use the BRBACKUP parameter `DD_FLAGS` to configure the largest possible block size (for example, 64 kB). The larger the block size, the better the general performance.

- Using the BACKINT interface

  The BACKINT interface allows you to connect external backup tools to BR*TOOLS.

- Optimizing BEGIN BACKUP runtimes

  Setting tablespaces in the backup mode using BEGIN BACKUP can take a long time. To minimize this time, see SAP Note 875477.

- Incremental backup

  Incremental backups are possible using RMAN. An incremental backup only backs up the blocks of data that have changed since the last full backup. Because RMAN does not backup the entire database, the volume of data is greatly reduced. Therefore, incremental backups can reduce runtime. However, because all blocks must be scanned, the reduction in runtime is not as great as the reduction of data volume. As of Oracle 10g, you can activate block change tracking, which can save a considerable amount of runtime. See SAP Note 964619.

- Partial backup

  If a backup of the complete database takes too long, you can carry out several partial backups. However, you must ensure that each file of the database is included in at least one partial backup.

- Two-phase backup

  Instead of saving the database directly to tape, a quick backup to disk can be performed as the first step of the two-phase backup. You can then save the database to tape in the second step of the two-phase backup.

- Split-mirror backup

  During the split-mirror backup, the system sets the tablespaces to backup mode (online backup), or the database is stopped (offline backup). Then the mirrored disks are separated. Finally, after a very brief period, the backup mode is ended, or the database is restarted. The separated disks can then be saved to tape. To ensure a fast restore, one or more mirrors from the last few hours or days can be stored on disk. When a restore is required, the earlier state can be restored much faster by mounting the mirror disks instead of using the time-consuming tape method.

- Snapshots

  In the snapshot method, all data from a time point are frozen at the input/output system level. When data is changed, the new version is stored, as well. Creating snapshots (with the database offline or in backup mode) allows you to create backups on disk very quickly. For details on using snapshots, consult with your hardware partner.

- Standby database

  A standby database is created, in addition to the primary database and is recovered in real time with the data from the primary database or with a defined time offset. If required, a standby database can be started as the primary database, without having to carry out a restore.

Several backup methods can be combined to eliminate the need for a time-consuming restore from tape.

> Note:
> You can NEVER completely rule out the need for a restore from tape, so you must ensure that a sufficient number of backup tapes are available.

Split-Mirror Disk Backup

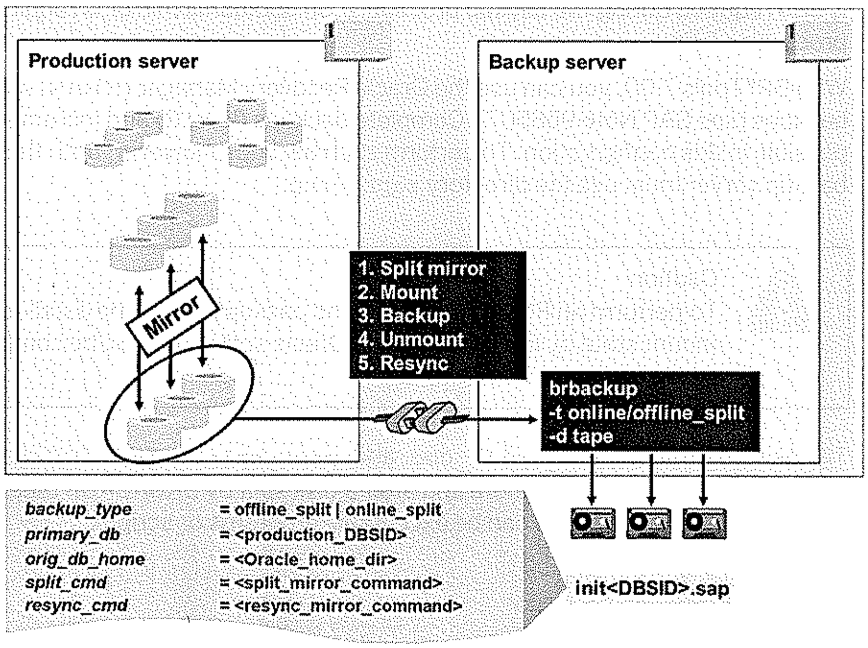


Figure 90: Split-Mirror Disk Backup

Split-mirror disk backups can significantly reduce backup time. At the start of a backup, the disk mirror where the data files are located is broken up by a predefined command. The mirror half is backed up from a separate server, while the production half is still running, without impairing performance.

After the backup is completed, the disk mirror can be resynchronized immediately or with a delay.

To perform online backup, proceed as follows:

1. Change the tablespaces to backup mode.
2. Break up the disk mirror.
3. End the backup mode in the production half.

4. Perform an online backup from the mirror.
5. Resynchronize the mirror.

To perform offline backup, proceed as follows:

1. Stop the database.
2. Break up the disk mirror.
3. Start the database in the production half.
4. Perform an offline backup of the mirror.
5. Resynchronize the mirror.

The configuration is performed by maintaining additional parameters in `init<DBSID>.sap` and must enable BRBACKUP, running on the backup server, to connect to the database on the production server.

During normal operation, disk mirroring protects against database failure. If such protection is also required during the backup procedure, an additional mirror is required for the production half.

With BR*Tools 7.00, you can use the Oracle Recovery Manager (RMAN) with split-mirror backups (see SAP Note 968507). Complete backups (level 0) and incremental backups are possible in this configuration. To perform such backups, however, both the database and the backup server must use the directory `sapbackup`.

> Caution:
> To perform an RMAN backup on the backup server, the database is set to the mount state by BRBACKUP. The Oracle software must be completely installed on the backup server.

The new parameters, `pre_spilt_cmd` and `post_split_cmd`, in BR*Tools 7.00 allow you to run external commands before and after a disk split by BRBACKUP (see SAP Note 968507). These commands can be executable programs or scripts.

SAP Tools and the Oracle Standby Database

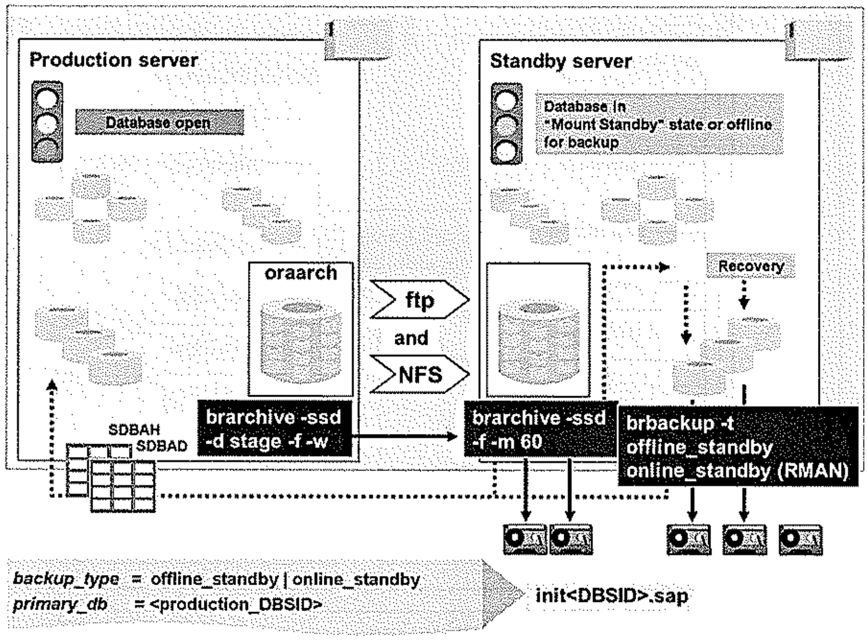


Figure 91: SAP Tools and the Oracle Standby Database

An Oracle standby database consists of two database servers. The production database has the status OPEN. During normal operation, the standby database has the status MOUNT, and is continually applying the offline redo log files from the production server. In case of a production server failure, the standby database can be opened, and can take on the role of the production database.

Data files are saved to tape on the standby server, using BRBACKUP with `offline_standby` as backup type. These actions are logged on the production server, in tables *SDBAH* and *SDBAD*, as well as in log files in directory `sapbackup` (tables *SDBAH* and *SDBAD*, along with the log files in the directory `sapbackup`, must be accessible from the standby server).

BRARCHIVE runs on both servers. From the production server, a continuous backup to a local NFS mounted, or remote disk is performed (using a verification, with BRARCHIVE option –`w`|`-verify`). On the standby server, backup to tape is performed from the `oraarch` directory, which can be mounted on the standby server using Windows shares or NFS on UNIX.

The offline redo log files can be applied to the standby database by using the option `-m`|-`modify <delay>`. The optional entry `<delay>` determines whether the connection is "hot" (that is, replicated with no delay) or "warm" (that is, replicated with a delay). The latter makes it possible to stop applying offline redo log files before a user error is replicated on the standby server.

With BR*Tools 7.00, you can use the Oracle RMAN with split-mirror backups (see SAP Note 968507) to perform online backups of standby databases, without importing offline redo log files, by choosing the backup type `backup_type=online_standby`. Offline backups of a standby database are also possible with RMAN. However, you must import the offline redo log files. You can perform complete backups (level 0) or incremental backups of the standby database.

Structure-Retaining Database Copy

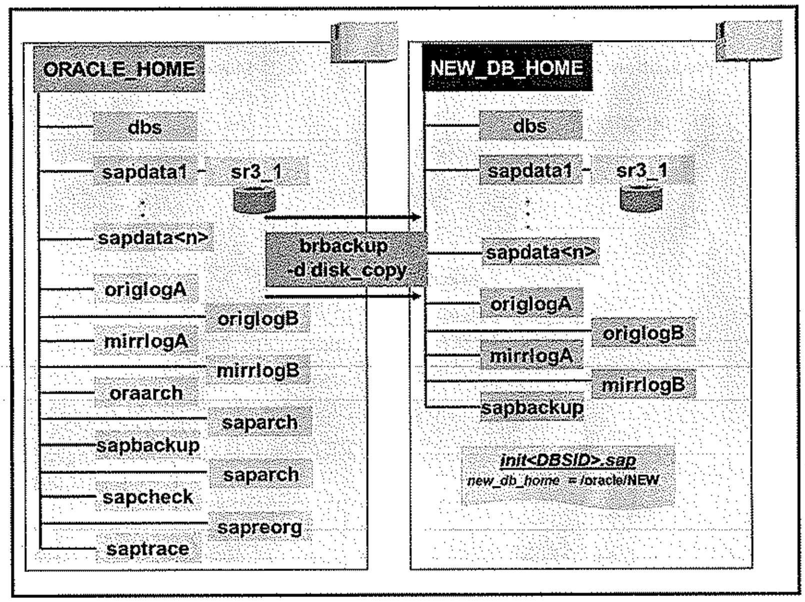


Figure 92: Structure-Retaining Database Copy

A structure-retaining database copy can be used to create a copy of the database on the same server with another <DBSID>, or to create a database copy on a remote server with the same or another <DBSID>.

In a typical scenario, a structure-retaining database copy is used to perform a homogenous system copy, in which the source and the target system use the same operating system (OS) and the same database.

Using a structure-retaining database copy is usually faster than using a system copy performed with SAP installation tools using R3load. However, by using R3load, the database is automatically reorganized, as it first exports the data into a database and OS-independent format, and then imports the data on the new server.

System Copy with BR*Tools

BR*Tools 7.00 contains important enhancements that provide additional support and automation for homogenous (within a hardware platform) and heterogeneous (between different hardware platforms) database copies. Support for homogenous database copies is based on the BRRECOVER functions, Database reset and Database point-in-time recovery, as opposed to the functions for heterogeneous database copies - the BRSPACE function Recreate database and the Oracle 10g feature Cross-Platform Transportable Tablespaces.

Enhanced Support for Homogenous Database Copies

This new function allows a fully-automated (no operator intervention) restructuring of a database, based on a complete offline or online backup, including on a computer other than the computer of the original database. You can also change the Oracle SID (`ORACLE_SID`) and the directories (`ORACLE_HOME`) and SAPdata home (`SAPDATA_HOME`).

You can structure a database copy with BRRECOVER based on the following backup types:

- A complete or incremental offline, or consistent online (`online_cons`) backup of the database (without additionally importing redo log files)
- A complete or incremental online backup of the database, or by additionally importing redo log files (forward recovery)

BRRECOVER and BRRESTROE automatically recognize the new environment (`ORACLE_SID`, `ORACLE_HOME`, and `SAPDATA_HOME`) . The database files are reloaded to the new directories. The files are automatically renamed in the control file after the restore. At the end of the process, BRRECOVER creates new control files with the new database name.

Enhanced Support for Heterogeneous Database Copies

This new function allows you to copy an SAP Oracle database between different hardware platforms. This is an alternative to procedures that are based on Oracle export and import, or SAP R3 load utilities, and can save time. These time-saving advantages are reduced. However, if database files must be converted because of different endian formats, you can also change the Oracle SID (`ORACLE_SID`), the directories (`ORACLE_HOME`), and SAP data home (`SAPDATA_HOME`). This procedure is supported as of Oracle 10g. To do this, you require the Oracle 10g feature Cross-Platform Transportable Tablespaces.

The system performs a heterogeneous database copy in the following steps:

1. Exports user tablespaces on the source system.
2. Copies the files from the user tablespaces to the target system.
3. Copies the scripts and dumps files to the target system.
4. Re-creates the database on the target computer.
5. Converts endian format of the database files (if required).
6. Imports user tablespaces to the new database.

The conversion of database files with different Endian formats is performed by Oracle RMAN, using BR*Tools.

> Hint:
> For more information about homogenous and heterogeneous system copies using BR*Tools, see SAP Notes 1003028.

LESSON SUMMARY

You should now be able to:

- Apply advanced backup techniques

Unit 2

Learning Assessment

1. Which of the following statements are correct with respect to Recovery Manager (RMAN)?

   Choose the correct answers.

   - [ ] A Using Recovery Manager (RMAN), you can perform online and offline backups.
   - [ ] B Using RMAN, you can perform complete, incremental, and partial backups.
   - [ ] C Using RMAN, you can perform whole and full backups.
   - [ ] D Without using RMAN, you can perform incremental backups.
   - [ ] E Without using RMAN, you can perform partial backups.

2. Which of the following statements are true when considering a suitable backup strategy?

   Choose the correct answers.

   - [ ] A Perform a complete backup every day.
   - [ ] B Performing incremental backups it is sufficient to perform a full backup initially.
   - [ ] C Perform incremental backups and a full backup once a week.
   - [ ] D Perform partial backups so that the complete database is backed up during a week.
   - [ ] E Ensure that the offline redo log files are backed up twice.

3. Which of the following statements are correct concerning backups?

   Choose the correct answer.

   - [ ] A When a daily offline backup is performed, backup of offline redo log files is not necessary.
   - [ ] B When performing an incremental backup, only changed blocks are backed up.
   - [ ] C Performing a daily consistent online backup is a sufficient backup strategy; no additional backups need to be performed.
   - [ ] D An online backup also backs up the offline redo log files.

4. Which of the following tools are used to save the BR*Tools profile init<DBSID>.sap?

   Choose the correct answers.

   - [ ] A BRBACKUP
   - [ ] B BRARCHIVE
   - [ ] C BRRESTORE

5. When a backup is directly started from the command line using BRBACKUP, the result and logs are also shown on the monitor to display DBA Operation Logs of the DBA Cockpit.

   Determine whether this statement is true or false.

   - [ ] True
   - [ ] False

6. The Oracle administration tools delivered by SAP include programs for backing up database files and other files of an SAP system and programs for restoring missing files and recovering data files to a consistent state. Which of the following is true?

   Choose the correct answers.

   - [ ] A The program BRBACKUP backs up Oracle data files, the control file, and online redo log files, where necessary.
   - [ ] B The program BRSPACE can restore all files belonging to the database system from the backups.
   - [ ] C The program BRARCHIVE backs up the online redo log files of the database.
   - [ ] D The program BRARCHIVE backs up the offline redo log files of the database.
   - [ ] E The program BRRESTORE can restore all files belonging to the database system from the backups.

7. An offline redo log file was backed up once by BRARCHIVE. The log file now has the status ____________.

   Choose the correct answer.

   - [ ] A COPIED
   - [ ] B DELETED
   - [ ] C SAVED
   - [ ] D SELECTED

8. To perform a complete database reset, you need which of the following backups?

   Choose the correct answers.

   - [ ] A Complete offline backup
   - [ ] B Complete online backup
   - [ ] C Consistent online backup
   - [ ] D Backup of offline redo log files

9. On a full backup, the database itself, including the Oracle software directories, are backed up.

   Determine whether this statement is true or false.

   - [ ] True
   - [ ] False

10. Which of the following are the various possibilities for optimizing backup time?

    Choose the correct answers.

    - [ ] A Hardware
    - [ ] B Parallel backup
    - [ ] C Software
    - [ ] D Split-mirror backup

11. A database copy can be structured with BRRECOVER based on which of the following?

    Choose the correct answers.

    - [ ] A Complete offline backups of the database without additionally importing redo log files
    - [ ] B Consistent online backup of the database without additionally importing redo log files
    - [ ] C Complete offline backups of the database with additionally importing redo log files
    - [ ] D Incremental online backup of the database or by additionally importing, redo log files

Learning Assessment - Answers

1. Which of the following statements are correct with respect to Recovery Manager (RMAN)?

   Choose the correct answers.

   [X] A Using Recovery Manager (RMAN), you can perform online and offline backups.

   [X] B Using RMAN, you can perform complete, incremental, and partial backups.

   [X] C Using RMAN, you can perform whole and full backups.

   [ ] D Without using RMAN, you can perform incremental backups.

   [X] E Without using RMAN, you can perform partial backups.

2. Which of the following statements are true when considering a suitable backup strategy?

   Choose the correct answers.

   [X] A Perform a complete backup every day.

   [ ] B Performing incremental backups it is sufficient to perform a full backup initially.

   [X] C Perform incremental backups and a full backup once a week.

   [X] D Perform partial backups so that the complete database is backed up during a week.

   [X] E Ensure that the offline redo log files are backed up twice.

3. Which of the following statements are correct concerning backups?

   Choose the correct answer.

   [ ] A When a daily offline backup is performed, backup of offline redo log files is not necessary.

   [X] B When performing an incremental backup, only changed blocks are backed up.

   [ ] C Performing a daily consistent online backup is a sufficient backup strategy; no additional backups need to be performed.

   [ ] D An online backup also backs up the offline redo log files.

4. Which of the following tools are used to save the BR*Tools profile init<DBSID>.sap?

   Choose the correct answers.

   [X] A BRBACKUP

   [X] B BRARCHIVE

   [ ] C BRRESTORE

5. When a backup is directly started from the command line using BRBACKUP, the result and logs are also shown on the monitor to display DBA Operation Logs of the DBA Cockpit.

   Determine whether this statement is true or false.

   [X] True

   [ ] False

6. The Oracle administration tools delivered by SAP include programs for backing up database files and other files of an SAP system and programs for restoring missing files and recovering data files to a consistent state. Which of the following is true?

   Choose the correct answers.

   [X] A The program BRBACKUP backs up Oracle data files, the control file, and online redo log files, where necessary.

   [ ] B The program BRSPACE can restore all files belonging to the database system from the backups.

   [ ] C The program BRARCHIVE backs up the online redo log files of the database.

   [X] D The program BRARCHIVE backs up the offline redo log files of the database.

   [X] E The program BRRESTORE can restore all files belonging to the database system from the backups.

7. An offline redo log file was backed up once by BRARCHIVE. The log file now has the status ____________.

   Choose the correct answer.

   [ ] A COPIED

   [ ] B DELETED

   [X] C SAVED

   [ ] D SELECTED

8. To perform a complete database reset, you need which of the following backups?

   Choose the correct answers.

   [X] A Complete offline backup

   [ ] B Complete online backup

   [X] C Consistent online backup

   [ ] D Backup of offline redo log files

9. On a full backup, the database itself, including the Oracle software directories, are backed up.

   Determine whether this statement is true or false.

   [ ] True

   [X] False

10. Which of the following are the various possibilities for optimizing backup time?

    Choose the correct answers.

    [X] A Hardware

    [X] B Parallel backup

    [ ] C Software

    [X] D Split-mirror backup

11. A database copy can be structured with BRRECOVER based on which of the following?

    Choose the correct answers.

    [X] A Complete offline backups of the database without additionally importing redo log files

    [X] B Consistent online backup of the database without additionally importing redo log files

    [ ] C Complete offline backups of the database with additionally importing redo log files

    [X] D Incremental online backup of the database or by additionally importing, redo log files

UNIT 3 Monitors and Tools

Lesson 1

Introducing Oracle Data Management 220
Exercise 9: Manage Oracle Tablespace Using SAP Tools 235

Lesson 2

Managing Database System Check 239
Exercise 10: Perform Database System Checks 255

Lesson 3

Explaining Computing Center Management System (CCMS) Alert Monitor 261

UNIT OBJECTIVES

- Describe how Oracle manages data
- Perform a database system check
- Explain Computing Center Management System (CCMS) alert monitors

Unit 3
Lesson 1

Introducing Oracle Data Management

LESSON OVERVIEW

This lesson introduces you to Oracle data management. This lesson also explains different types of tablespaces and their naming conventions.

Business Example

SAP tools and transactions make the monitoring and administration of an Oracle database easier because the database administrator (DBA) does not need to know the exact Oracle commands to get the monitoring information and to administrate the Oracle database. Some background knowledge is needed to interpret the monitoring results and how to know which actions must be performed on the database, as well as when and why. You need to learn more about these SAP tools and transactions. For this reason, you require the following knowledge:

- An understanding of how Oracle stores data
- An understanding of the difference between dictionary and locally-managed tablespaces
- An understanding of the purpose of undo and temporary tablespaces

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Describe how Oracle manages data

Oracle Data Management

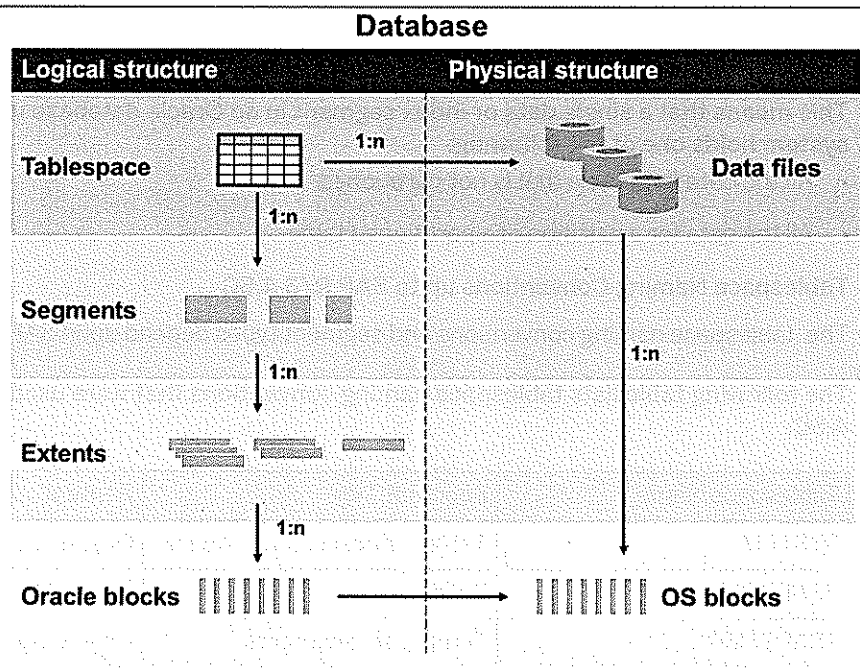


Figure 93: Tablespaces, Data Files, and Segments

Oracle stores its data in tablespaces. A tablespace consists of one or more files. Data files are usually stored in the local file system, but they can also be stored on raw disks. A data file contains the actual data.

Oracle has the following main segment types:

- Data

  A data segment contains table data in rows.

- Indexes

  Indexes are used for faster access to table data and to enforce unique constraints. Each table in SAP has one primary index and optional secondary indexes. The index contains the key fields of a table and points to the data block where the corresponding table row is stored.

- Temporary segments

  Temporary segments are used to sort tables and to create indexes.

- Undo/Rollback segments

  Undo/rollback segments are used to provide read consistency, the ability to roll back or undo changes to tables, and for recovery.

A segment is stored in single tablespace. Within the tablespace, a segment can be spread across several data files.

To meet the demands of very large databases, database designers create partitioned tables and indexes. Partitioned tables allow data to be split into smaller and easily manageable units, called partitions. Each partition in a table is stored in its own data segment and can be managed individually. You can create a partition in a separate tablespace, however, doing so is not recommended for SAP databases.

Operations on partitioned tables and indexes can also be performed in parallel. Partitioned tables are used, for example, in SAP Business Information Warehouse (SAP BW) systems, to store the fact tables of InfoCubes.

This means that a single data or index segment in an Oracle database used in an SAP system holds one of the following:

- All the data for a table that is not partitioned
- All the data for a partition of a partitioned table

Tablespace Naming Conventions up to SAP R/3 4.6C

The tablespace naming conventions and segment types depend upon SAP version.

The following table lists tablespace naming conventions that were used up to SAP R/3 4.6C:

Tablespace Name	Segment Type	Content
SYSTEM	Tables, indexes, and rollback	Oracle data dictionary
PSAPROLL	Rollback	Only rollback segments
PSAPTEMP	Temporary segments	Only temporary segments
PSAP<NAME>D	Tables	SAP tables – tables are logically grouped into different PSAP<NAME>D tablespaces
PSAP<NAME>I	Indexes	SAP indexes – all indexes to tables in a specific PSAP<NAME>D tablespace are stored in a corresponding PSAP<NAME>I tablespace

The following table lists the tablespace names used to store SAP tables and indexes up to R/3 4.6C:

Tablespace Name	Content
PSAPBTABD/I	Transaction data (frequently changed data)
PSAPSTABD/I	Master data (rarely changed data)
PSAPPOOLD/I	SAP pool tables
PSAPPROTD/I	Log information
PSAPLOADD/I	SAP loads (compiled ABAP programs)
PSAPSOURCED/I	SAP sources (ABAP)
PSAPDOKUD/I	Documentation tables
PSAPCLUD/I	SAP cluster data
PSAPDDICD/I	SAP data dictionary

Tablespace Name	Content
PSAPUSER1D/I	Customer data
PSAPEL<REL>D/I	Release-dependent SAP loads
PSAPES<REL>D/I	Release-dependent SAP sources

The advantages of the tablespace layout as of R/3 4.6C are as follows:

- Data files can be stored on different physical disks and controllers, according to their use (for example, PSAPBTABD has much higher input and output than PSAPES<REL>D) and segment type (tables on different disks than indexes).
- Reorganizational units are smaller.

The disadvantages of the tablespace layout as of R/3 4.6C are as follows:

- Greater administration effort
- Less effective space management, that is, free space in one tablespace cannot be used by any segment in another tablespace

Tablespace Naming Conventions Starting with SAP Web APPLICATION SERVER (AS) 6.10

The following table lists the tablespace naming conventions that came into use starting with SAP Web AS 6.10:

Tablespace Name	Segment Type	Contents
SYSTEM	Tables, indexes, and rollback	Oracle data dictionary
SYSAUX	Tables, indexes	Mandatory help tablespace for the SYSTEM tablespace
PSAPROLL or PSAPUNDO	Rollback/undo	Only rollback/undo segments
PSAPTEMP	Temporary segments	Only temporary segments
PSAP<SCHEMA-ID>	Tables, indexes	Objects of the SAP NetWeaver AS ABAP schema
PSAP<SCHEMAID><REL>	Tables, indexes	Release-dependent data (ABAPs, loads)
PSAP<SCHEMA-ID>USR	Tables, indexes	Customer objects
PSAP<SCHEMA-ID>DB	Tables, indexes	Objects of the SAPNetWeaver AS Java schema (starting with SAPNetWeaver AS 6.40)

This new tablespace layout is also called the Multiple Components in One Database (MCOD) layout. Using this layout, it is possible to store the data of several SAP systems in a single database.

To distinguish between data for SAP systems C11 and C12 stored in the same MCOD database, SAP processes no longer connect to the database with the user `SAPR3`, but with the user `SAP<SCHEMA-ID>`. Each user has its own schema, which has the same name as user, and all database objects belong to the schema named for the user who created it. By default, every access to a database object accesses the object belonging to the schema of the calling user.

> Note:
> Oracle database 10g has a new mandatory tablespace, the SYSAUX tablespace. SYSAUX is a mandatory help tablespace for the SYSTEM tablespace. It provides a central location for necessary additional metadata outside the SYSTEM tablespace. Some components and products that used the SYSTEM tablespace or their own tablespaces in earlier Oracle versions now occur in this tablespace. This reduces the load on the SYSTEM tablespace and simplifies administration. Only the actual Oracle dictionary is still located in the SYSTEM tablespace (SYS and SYSTEM schemas). The SYSAUX tablespace is always created with new installations or database upgrades. In a normal operation, the SYSAUX tablespace is neither renamed nor deleted. The SYSAUX tablespace can also be transported. If the SYSAUX tablespace is missing (media failure), certain database features are not available, or will be restricted (for example, Automatic Workload Repository (AWR), historical data, and Enterprise Manager).

Database Schema

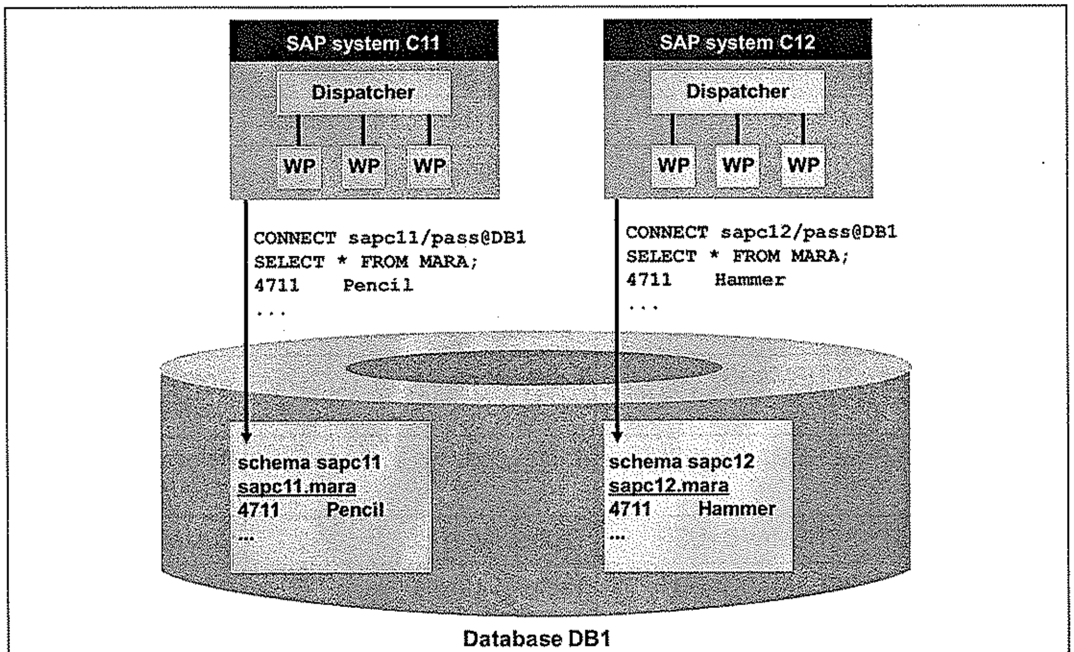


Figure 94: Database Schema

In the example shown in the figure, by using different schema, SAP system C11, sees the contents of the *MARA* table belonging to schema SAPC11 (SAPC11.MARA) because C11 connects as SAPC11. System C12 will see a completely different table (SAPC12.MARA).

In the old layout, the SAP system identifier (SID) and the Oracle SID were always equal. However, in the new layout, the two SIDs can be different. For the connection of the SAP instance to Oracle, a third SID-like parameter, SCHEMA-ID, is introduced.

The different types of SID parameters are as follows:

- SAP SID (SAPSID)

  This parameter is the system identifier for an SAP system. This parameter is set during central instance installation and stored in the SAP profile parameter `SAPSYSTEMNAME` (usually in the default profile).

- Database SID (DBSID)

  This parameter is the system identifier for the data base. This parameter can be different from the SAP SID. The database SID is set during database creation and can be read from the *V$DATABASE* view, the *NAME* field.

- Schema ID

  This parameter is used as part of tablespace names (for example, SAP<SCHEMA-ID>USR) and as part of the database schema and username that SAP uses to connect to the database (SAP<SCHEMA-ID>). The Schema ID parameter is set by the SAP installation tool on the database installation. The schema, used by the SAP instance and various other tools, such as R3trans, is read from the environment variable `DBS_ORA_SCHEMA`.

All IDs can be different, but they do not need to be different because of following reasons:

- If MCOD is not used (every SAP system has its own database), all IDs can be identical.
- If MCOD is used, at least the second SAP system storing its data in the MCOD database needs a SCHEMA-ID different from the DBSID.

Points to be remembered while using MCOD are as follows:

- Make sure to understand when the MCOD tablespace layout is needed. Having multiple components in one database requires the MCOD tablespace layout, however, having one database per SAP system does not necessarily require the MCOD tablespace layout.
- As of SAP Basis 4.6C SR2, SAP installation tools create the MCOD tablespace layout and support the load of the new data into another schema of an existing database. However, you can still load the data into a new database.
- BR*Tools still supports the old tablespace layout. Currently, there is no need to switch to the MCOD tablespace layout for existing databases if you do not plan to use multiple components in one database.

Data Files

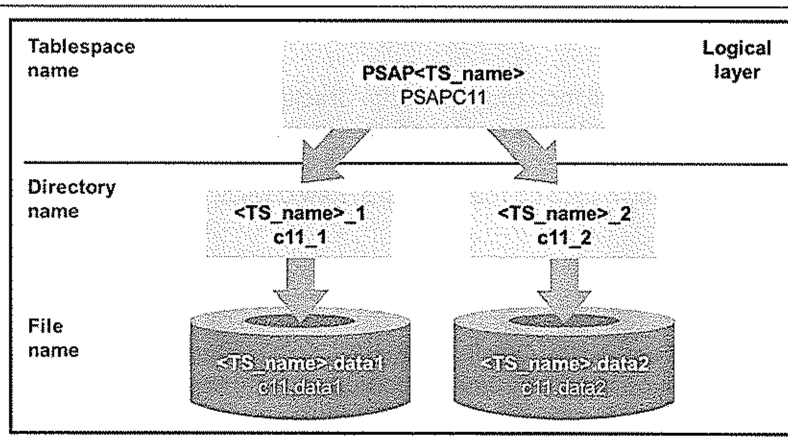


Figure 95: Naming Conventions for Data Files

Each tablespace consists of one or more data files that reside in the file system.

The naming conventions for each data file are as follows:

- Each data file is created in a separate subdirectory of the `sapdata<n>` directory. The directory is named like the tablespace, without the PSAP prefix and with a number that represents the <n>th data file of the tablespace.
- Each data file has same name as the tablespace, without the PSAP prefix and with the extension `.data<n>`.

If a tablespace is full, you can extend the tablespace in the following ways:

- Add another data file to the tablespace.
- Manually resize an existing data file.
- Change the properties of an existing data file to be autoextensible. In this case, the file grows automatically when more space is needed in the tablespace. The maximum file size (`MAXBYTES`) and the size by which the data file is enlarged (`INCREMENT_BY`) can be specified.

There is no general recommendation as to whether the size of tablespaces should be increased manually or automatically. If tablespaces are increased manually, the free space in each tablespace must be monitored and, if there is a lack of free space, increased manually. If all data files are autoextensible, the file systems must be monitored to make sure they have enough free space for the data files to grow.

Although the size of a data file is less important because there is no longer a 2 GB size limit on any supported operating system (OS), the number of data files is approximately 100 when the database reaches its expected size. As a general rule, the data file size must be the expected database size divided by 100, but not smaller than 2 GB.

The following table lists the expected database size and file size:

Expected database size	File size
Up to 200 GB	2 GB
200 to 400 GB	4 GB
400 to 800 GB	8 GB
Larger than 800 GB	16 GB

Raw Devices

The SAP R/3 installation tools store the Oracle data files by default in a file system. It is possible to store data files on raw devices in versions of Oracle up to Oracle 10g. In such cases, no file systems are created on the disks storing the data files, and Oracle is responsible for storing its own data on the raw disks. Raw devices are supported only on the UNIX OS.

> Hint:
> Installing new Oracle 11.2 databases on raw devices is not recommended because the Oracle install tools do not support raw devices. Oracle plans to stop supporting raw devices completely with the next version following 11.2.

The advantages of using raw devices are as follows:

- Faster input and output because input and output is not buffered in the OS buffer cache
- Less disk space is required (missing file system overhead)
- Faster crash recovery because no file system checks have to be performed

The disadvantages of using raw devices are as follows:

- Raw devices must be well documented, as standard UNIX tools, such as MOUNT or DF, show only file systems, not raw devices.
- Storage configuration is inflexible because only one data file is permitted for each raw device.
- Backup is possible using only Recovery Manager (RMAN) or Data Description (DD) (both supported by BR*Tools).

Dictionary-Managed Tablespaces

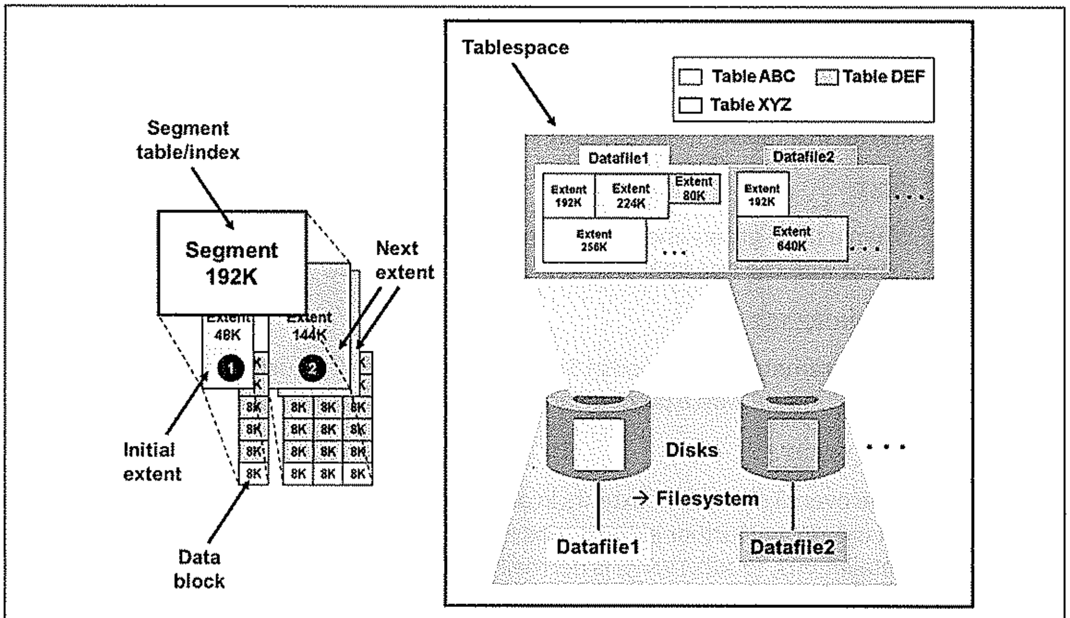


Figure 96: Dictionary-Managed Tablespaces Storage Parameters

All tablespaces created by SAP installation tools in the old tablespace layout were created as dictionary-managed tablespaces. SAP recommends using locally-managed tablespaces. Locally-managed tablespaces are the default tablespaces when the new MCOD tablespace layout is created by the installation tool.

To switch from dictionary-managed to locally-managed tablespaces on existing databases, reorganization is necessary. Whether this reorganization can be done online or must be done with export or import depends on the database version.

Dictionary-managed tablespaces and their administration are introduced to support DBAs who still must manage these tablespaces, and understand the advantages of locally-managed tablespaces.

Each segment consists of one or more extents. An extent is a collection of Oracle blocks and must be allocated contiguously in a single data file.

In dictionary-managed tablespaces, Oracle checks and updates data dictionary tables whenever a new extent must be allocated (for example, due to table growth) or deleted. The extent size and other parameters are called storage parameters, and are stored in the Oracle data dictionary.

Storage Parameters

Storage parameters can be set during the creation of a tablespace. These parameters are then used for all tables in the tablespace as defaults. The default tables can be overridden when a table is created or changed with BR*Tools for existing tables.

The storage parameters and their meanings are as follows:

- `MINEXTENTS`

  This parameter displays the minimum number of extents allocated when a table is created. SAP installation tools usually create tables with a single initial extent. In special cases (rollback segments, reorganization, and so on), other values can be chosen.

- `INITIAL`

  This parameter displays the size of the initial extent, which is allocated when a table is created. The SAP installation tools create the initial extent large enough to import the table into a single initial extent if the table size is known. If the table is initially empty, the `INITIAL` parameter is calculated from information in the SAP data dictionary (the *IAORA* and *IGORA* tables, maintained in transaction `SE11`, *technical settings → size category*).

- `NEXT`

  When a table grows and the existing extents are full, Oracle automatically creates a next extent with the size specified in the `NEXT` parameter.

- `PCTINCREASE`

  This parameter is not used in SAP systems (set to 0). Further extents are created with `PCTINCREASE` with a percent larger than the previous next extent.

- `MAXEXTENTS`

  If the overall number of extents in a segment corresponds to `MAXEXTENTS`, Oracle does not create any further extents, and issues the error message *ORA-01553*. In SAP installations, `MAXEXTENTS` is normally set to values between 100 and 300 (see the *IGORA* table).

Dictionary-Managed Tablespaces Extent Allocation

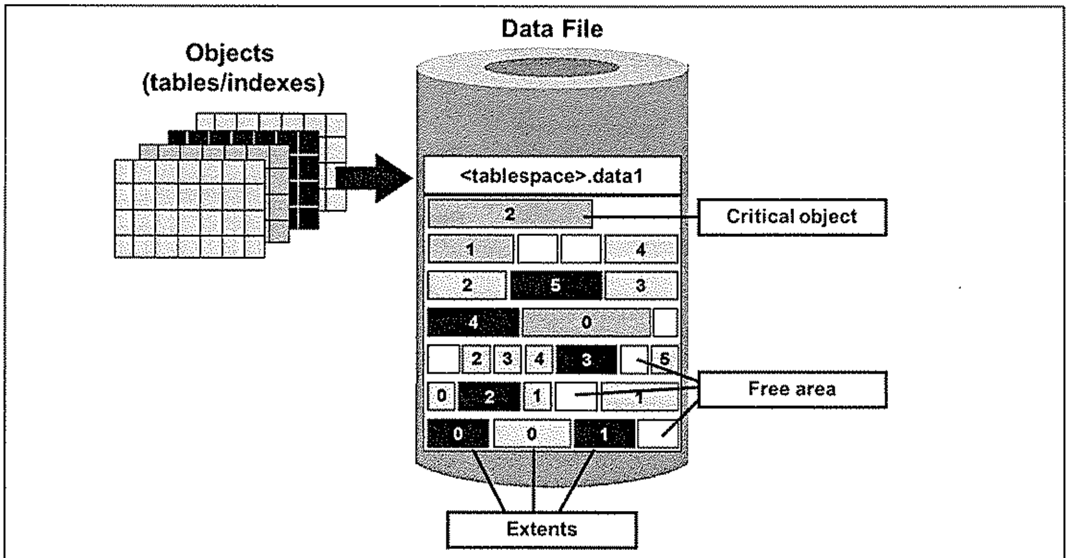


Figure 97: Dictionary-Managed Tablespaces Extent Allocation

This way of allocating space for segments in tablespaces has several disadvantages and requires the following additional administration efforts:

- The DBA must allocate extents contiguously within one data file. To check whether a tablespace has enough free space, it is not enough to sum up all free areas. If the check finds any segment having a `NEXT` size larger than the largest free area, a tablespace overflow (*ORA-01653* for tables and *ORA-01654* for indexes) can occur if Oracle will have to create a next extent for this segment.
- When extents are dropped, they are marked as free and their blocks can be used by new extents. But adjacent free extents are not combined into one large free extent immediately. Therefore, large free areas are not detected and not used for large next extents. The DBA must coalesce free extents in this case, which means that several adjacent free extents are combined into one large free extent.
- To avoid Oracle error *ORA-01553* (maximum number of extents reached), the DBA must regularly check whether any segment has a number of extents allocated that is close to `MAXEXTENTS`. In this case, `NEXT` must be increased so that Oracle creates less but larger extents.
- Optimally, the `NEXT` parameter for all tables must be regularly adapted to the current table growth to make sure that Oracle does not need to create a next extent more often than, for example, one month.

The SAP tool BRCONNECT can perform these checks and automatically adapt the `NEXT` storage parameter. The tool BRSPACE can be used to coalesce free extents manually, while `brconnect -f check` coalesces free extents automatically. By using locally-managed tablespaces, these problems will be eliminated.

Locally-Managed Tablespaces

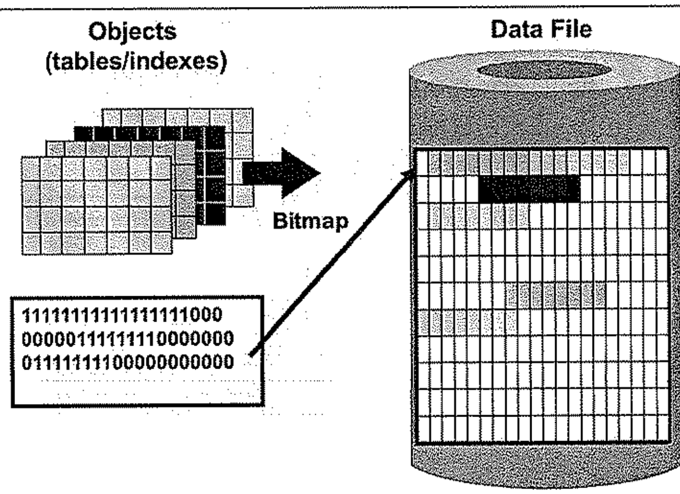


Figure 98: Locally-Managed Tablespaces Extent Allocation

All tablespaces created by SAP installation tools in the new MCOD tablespace layout are created as locally-managed tablespaces.

With locally-managed tablespaces, each data file has a bitmap listing and free blocks in the data file. When a new extent must be allocated, Oracle selects a data file and checks the bitmap to make sure that enough contiguous blocks exist to create a new extent. This is done for each data file of the tablespace, for as long as there is enough free space for the allocation of the extent.

The extent size is no longer determined by storage parameters. The extent size is either identical for all extents (`UNIFORM`) or automatically chosen (`AUTOALLOCATE`), depending on the segment size. The kind of extent allocation (`UNIFORM` or `AUTOALLOCATE`) is determined at the creation of the tablespace and cannot be changed.

SAP uses locally-managed tablespace with automatic extent allocation as of Oracle 9.20 and recommends that you create new tablespaces of this type as of Oracle 8.1.7 and SAP kernel 4.6D.

> Note:
> For details, see SAP Note 214995.

The exceptions to using locally-managed tablespaces are as follows:

- The temporary tablespace PSAPTEMP is created locally managed but with a `UNIFORM` extent size.
- The rollback tablespace PSAPROLL is created as a dictionary-managed tablespace. For rollback, a new PSAPUNDO tablespace should replace PSAPROLL.
- In SAP BW systems, tablespaces holding fact tables or aggregates should be created with a uniform extent size of 1 MB.

For automatic extent allocation (`AUTOALLOCATE`), fixed rules are used to determine the size of the next extent, depending on the total size of the segment.

The following table lists the segment size and its corresponding extent size:

Segment size	Next extent size	Maximum number of extents with this size
Less than 1 MB	64 KB	16
Between 1 MB and 64 MB	1 MB	63
Between 64 MB and 1 GB	8 MB	126
More than 1 GB	64 MB	Unlimited

This fixed rule is a compromise between having too many extents and wasting disk space for segments of different sizes in tablespaces.

The characteristics of the segment size are as follows:

- A small segment with less than 1 MB will have a maximum of 16 extents; wasted space for a new extent is less than 64 KB.
- A large table with 20 GB will have 461 extents; wasted disk space due to extent allocation is less than 64 MB, which is about 0.3% of the total used space.

The advantages of using locally-managed tablespaces are as follows:

- Adaption of `NEXT` or `MAXEXTENTS` is not necessary anymore. Oracle error *ORA-01553* (maximum number of extents reached) no longer occurs.
- Less fragmentation of a data file occurs and the available space in the data file is better used.
- Better performance when dropping or creating tables, and on parallel loads, because no data in the dictionary tables have to be updated when new extents are created or deleted.

The disadvantage of using locally-managed tablespaces is as follows:

- Check for used and free space is more expensive

Undo Management

Besides storing data, the Oracle database must provide the following consistencies:

- Transactional Consistency

  Normally, an individual database transaction performs inserts, updates, or deletes in many different tables. In SAP systems, these database transactions are performed by the update and background work processes. If such a database transaction terminates before all changes are performed in the database and the work process commits its changes, Oracle has to be able to roll back or undo these changes.

- Read Consistency

  It often happens that a transaction reads data from a database while it is being changed or deleted by another transaction. While the changing transaction is not yet finished, any reading process must get the information existing before the changing transaction has modified the data.

To provide transactional consistency and read consistency, Oracle uses, as of Oracle 9i, undo segments in an undo tablespace or the former rollback segments in a rollback tablespace. SAP recommends that you switch to the undo tablespace when using Oracle 9i because the undo tablespace is easier to handle and provides some new functionality.

When a database transaction starts, Oracle selects one existing undo or rollback segment for this transaction. Whenever table data is changed, the data is changed in its original block, after the old data was copied into the assigned undo or rollback segment. Whenever a rollback is necessary (for example, a transaction is cancelled or an instance recovery is performed after an instance crash), the original data is recovered from the undo or rollback segments (transactional consistency). Whenever a process reads data that is actually changed by another process, the reading process reads from the undo or rollback segments (read consistency).

Rollback Tablespace

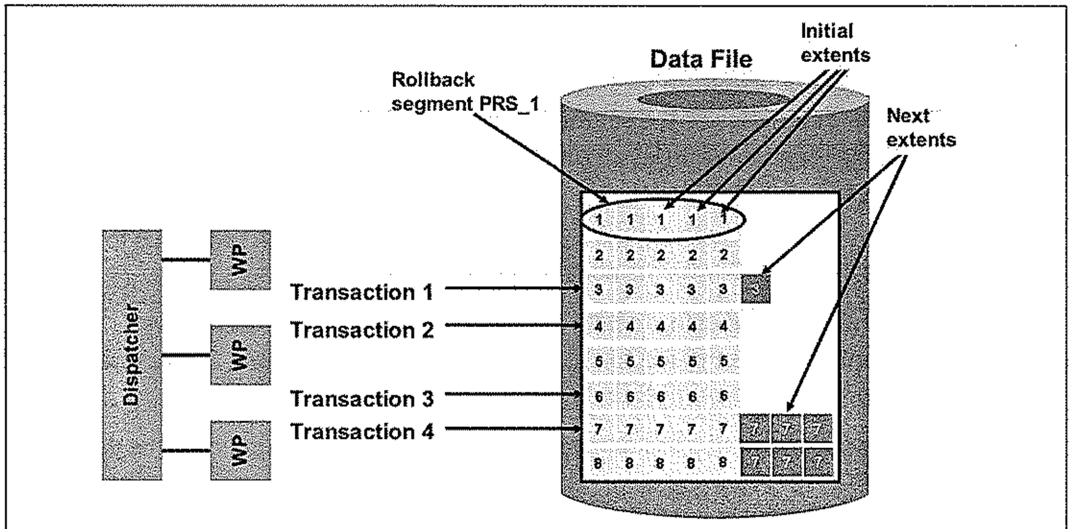


Figure 99: Rollback Segments and Extents

The internal extent management of the rollback tablespace is very similar to the internal extent management of dictionary-managed tablespaces. Rollback segments are created by the SAP installation tool using the storage parameters `INITIAL`, `MINEXTENT`, `MAXEXTENTS`, and `NEXT`.

For rollback segments, the special storage parameter `OPTIMAL` is set. Whenever a rollback segment is fully occupied, Oracle automatically allocates a `NEXT` extent. When any transaction using this rollback segment commits, the freed extents are marked to be overwritable by other transactions. In certain cases, Oracle also tries to regain free space in the rollback tablespace by shrinking the rollback segments to their `OPTIMAL` size.

This behavior causes a common error *ORA-01555* (snapshot too old).

The process to manage extents is as follows:

1. A process, for example, a background job, starts changing or deleting data.
2. Another process reads the data being changed by the other process.
3. The changing process commits because the commit within the rollback segment extents are overwritten by a third process or the rollback segments shrink to their optimal size.
4. The reading process cannot read anymore from the rollback segments – *ORA-01555*.

As with data dictionary tablespaces, this way of managing extents has several disadvantages.

To overcome, the system needs the following additional administrative efforts:

- Setting good storage parameters for the rollback segments depends on the database size and the actions performed in an SAP system. Experimenting with the parameters is necessary.
- Due to the fact that Oracle can select any rollback segment for a transaction, the storage parameters must always be changed for all rollback segments.
- For special actions, such as large client copies, the storage parameters used for normal operation are not good. In this case, SAP recommends switching to a special, larger rollback tablespace (usually called PSAPROLLBIG) with special storage parameter settings. After this special action, a switch back to the standard rollback segments must be performed.

Undo Tablespace

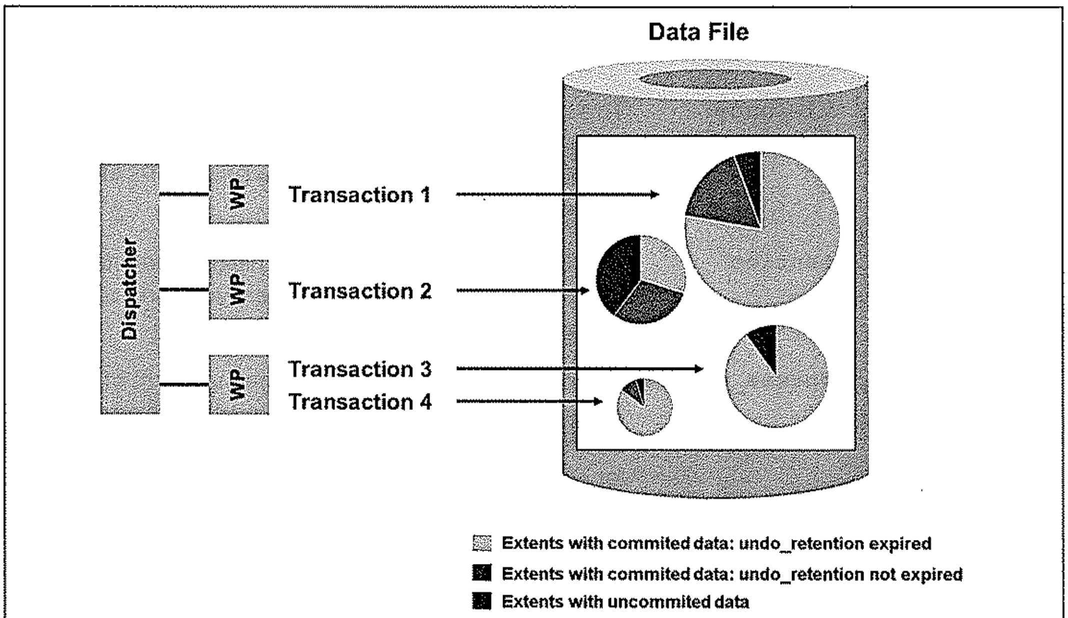


Figure 100: Automated Undo Management

As of Oracle 9i, Oracle introduced a new feature called automatic undo management (AUM). This feature uses a new tablespace type called undo tablespace (SAP convention – PSAPUNDO) with undo segments instead of rollback segments.

There are currently no disadvantages known when using AUM.

The advantages of using AUM are as follows:

- Undo segments are managed automatically. There are no storage parameters to maintain them; even the number of undo segments is configured automatically and can change dynamically.
- A transaction can use more than one undo segment.
- The criteria for undo segments being overwritten is no longer the commit of the transaction, but a time period defined by the profile parameter `undo_retention`. Setting the undo retention time equal to the run time of the longest running background job avoids *ORA-01555* completely (as long as the undo tablespace is large enough).

- The conversion to larger undo tablespaces for special actions, such as large client copies, is simpler.
- If more space is needed than is currently available in the expired part of tablespaces storage, then the following changes occur:
  - An extent is allocated from the free space.
  - An expired extent is stolen from another undo segment.
  - An unexpired but committed extent is overwritten.
  - An unexpired but committed extent from another undo segment is stolen.
  - An out-of-space error is fired (current transaction is rolled back).

For AUM, the following new profile parameters are available:

`undo_management`	This parameter can be set to `MANUAL` or `AUTO`. To turn AUM on, the parameter must be set to `AUTO`.
`undo_tablespace`	This is a special tablespace created with the Oracle command `CREATE UNDO TABLESPACE 'PSAPUNDO'`. You must specify the name of the undo tablespace (usually PSAPUNDO) in this parameter.
`undo_retention`	Oracle keeps the undo data for this time period, in seconds (called undo retention time). Then, you can overwrite the undo data.
`undo_suppress_errors`	When activating AUM, manual management of rollback segments is no longer possible and causes errors. To suppress errors concerning manual management of rollback segments, for example, in scripts, this parameter can be set to *true*.

> Caution:
> Oracle 10g no longer supports this procedure.

> Hint:
> SAP recommends that you switch from rollback tablespace to undo tablespace, as of Oracle 9.2.0. The procedure to switch is described in detail in SAP Note 600141. Only a general description is given in the lesson.

To switch from manual management to AUM, perform the following steps:

1. Create the new undo tablespace, PSAPUNDO.
2. Change or insert the four new Oracle parameters with appropriate values.
3. Drop all rollback segments, except the SYSTEM rollback segment.
4. Delete or comment out the Oracle parameter `rollback_segments`.
5. Drop the rollback tablespace.

Unit 3
Exercise 9

Manage Oracle Tablespace Using SAP Tools

Business Example

As the DBA analyzing the architecture of an existing installation, you want to check the tablespace layout of your database.

Check the tablespace types of your database.

1. Check which tablespaces are dictionary managed and which are locally managed. Check the detailed information of tablespace PSAP<Schema-ID>.

Unit 3
Solution 9

Manage Oracle Tablespace Using SAP Tools

Business Example

As the DBA analyzing the architecture of an existing installation, you want to check the tablespace layout of your database.

Check the tablespace types of your database.

1. Check which tablespaces are dictionary managed and which are locally managed. Check the detailed information of tablespace PSAP<Schema-ID>.
   a) To check the tablespace types, start BRGUI or BRTOOLS and choose *Space management → Additional space functions → Show tablespaces*.
   b) Choose *Continue* two times. The following list displays space information for all tablespaces of your database:

```
List of database tablespaces

Pos.  Tablespace      Type   Status     ExtMan.   SegMan.   Backup
Files/AuExt.
Compr.  Encr.
      Total[KB]    Used[%]     Free[KB]  MaxSize[KB]  ExtSize[KB]
FreeExt.
Largest[KB]

  1 - PSAPT99         DATA   ONLINE     LOCAL     AUTO
NO          1/1
NO      NO
         20480       5.00        19456          51200
30720          1
30720+:19456:0:0:0
  2 - PSAPT99USR      DATA   ONLINE     LOCAL     AUTO
NO          1/0
NO      NO
         10240      43.75         5760          10240
0          1
5760:0:0:0:0
  3 - PSAPTEMP        TEMP   ONLINE     LOCAL     MANUAL
NO          1/0
NO      NO
         20480       0.00        20480          20480
0          0
   0:0:0:0:0
  4 - PSAPUNDO        UNDO   ONLINE     LOCAL     MANUAL
NO          1/0
NO      NO
         20480       5.00        19456          20480
0        135
1152:1024:1024:1024:1024
  5 - SYSAUX          DATA   ONLINE     LOCAL     AUTO
NO          1/0
```

```
NO       NO
           204800       92.09        16192          204800
0          21
13248:320:320:256:256
  6 - SYSTEM            DATA   ONLINE    DICT       MANUAL
NO          1/0
NO       NO
           409600       45.12       224800          409600
0           1
224800:0:0:0:0
```

c) To get more details, choose the row with tablespace PSAP<Schema-ID>.

d) Choose *Continue*. The following information displays:

```
Information about tablespace PSAPT99

 1 - Tablespace type (type) .................. DATA
 2 - Tablespace status (status) .............. ONLINE
 3 - Extent management (extent) .............. LOCAL
 4 - Segment space managenent (space) ......... AUTO
 5 - Backup status (backup) .................. NO
 6 - Number of files in tablespace (files) .... 1
 7 - Number of autoextensible files (autoext) . 1
 8 - Tablespace compression (compression) ..... NO
 9 - Tablespace encryption (encryption) ....... NO
10 - Total tablespace size in KB (total) ...... 20480
11 - Used space in tablespace in % (used) ..... 5.00
12 - Free space in tablespace in KB (free) .... 19456
13 - Maximum tablespace size in KB (maxsize) .. 51200
14 - Maximum extension size in KB (extsize) ... 30720
15 - Used space as to max. size in % (used) ... 2.00
16 - Free space as to max. size in KB (free) .. 50176
17 - Allocated disk space in KB (space) ....... 20488
18 - Number of free extents (freeext) ......... 1
19 - Largest free extents (largest) .......... 30720+:
19456:0:0:0
```

LESSON SUMMARY

You should now be able to:

- Describe how Oracle manages data

Unit 3
Lesson 2

Managing Database System Check

LESSON OVERVIEW

This lesson explains how to use and customize the database system checks.

Business Example

Your SAP system worked fine till last night, but in the morning, you had complaints from several users. One user told you that she had received an ABAP short dump indicating a broken update, other users complained that they did not see the data they had entered shortly before. Still others report broken background jobs.

You checked the ABAP short dump, which indicated Oracle error ORA-01653. After checking the Oracle alert log file, you found the same error saying that a tablespace is full and needs to be enlarged.

If you had run the regular checks introduced in this lesson, you would have been warned two weeks ago that the tablespace was using more than 95% full. You must ensure, in future, that you enlarge the tablespace before the tablespace overflow occurs. For this reason, you require the following knowledge:

- An understanding of the purpose of running regular database system checks
- An understanding of how to run regular checks with SAP tools
- An understanding of how to interpret the results of database system checks
- An understanding of how to create a strategy to prevent possible problems and errors

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Perform a database system check

Database System Checks

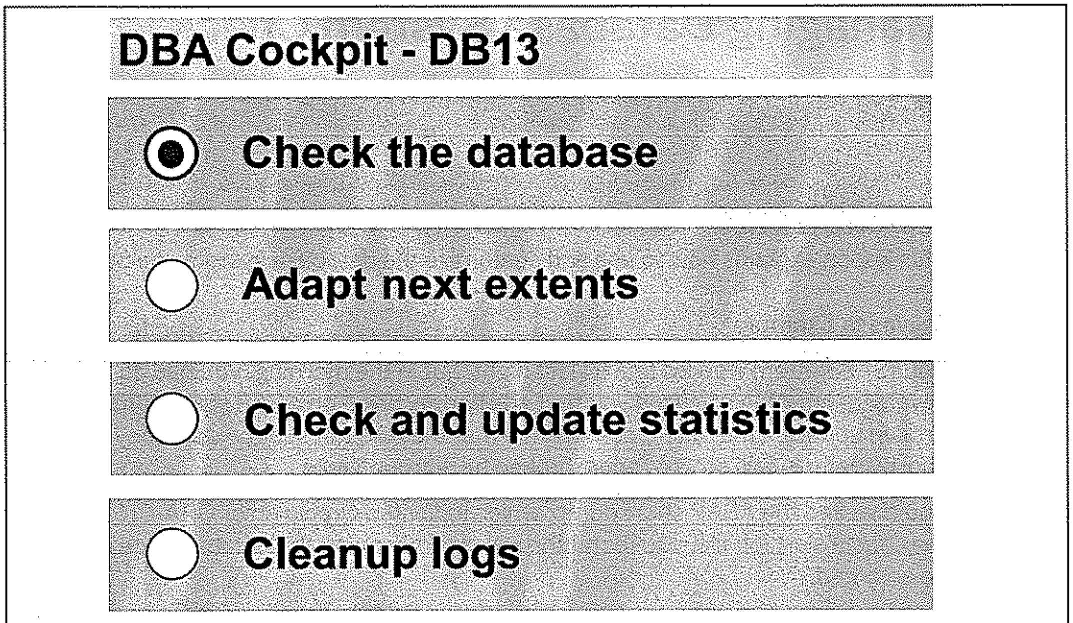


Figure 101: BRCONNECT Activities – Check Database

BRCONNECT is the main tool used to check the database.

Besides checking the database, BRCONNECT can also perform the following tasks:

- Adapt the NEXT extent size
- Create the optimizer statistics
- Clean up DBA logs

brconnect -f check

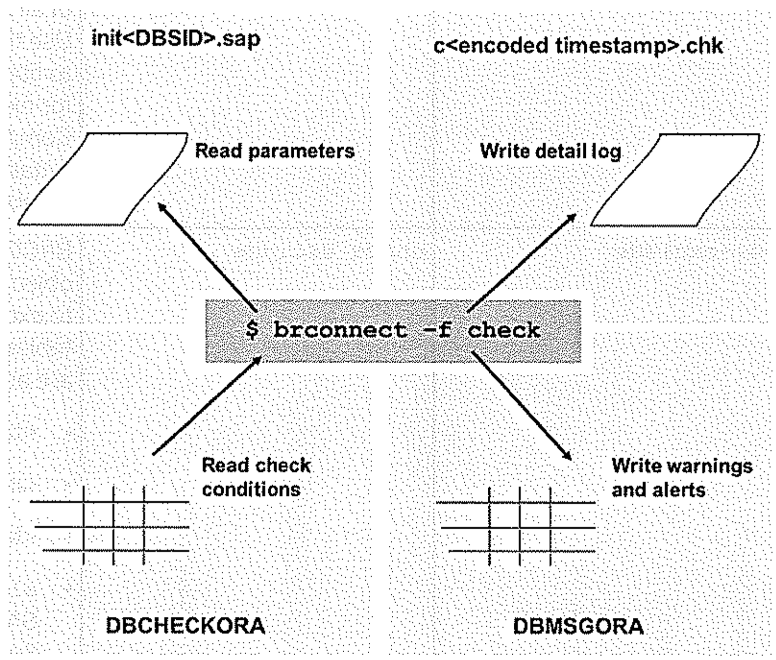


Figure 102: brconnect -f check

You schedule the tool `brconnect -f` check daily.

The default behavior of BRCONNECT can be adapted with parameters in the file `$ORACLE_HOME/dbs/init<DBSID>.sap` in UNIX or `%ORACLE_HOME%\database\init<DBSID>.sap` in Windows. To override these defaults, command line options can be used.

To check the database, BRCONNECT reads which checks it should perform and conditions for the checks, from the DBCHECKORA table.

If any warnings or errors are detected, they are written into the table DBMSGORA and a detailed log file, `$SAPDATA_HOME/sapcheck/c<encoded timestamp>.chk`, is created.

> Note:
> Normally, BRCONNECT is started using `brconnect -u / -c -f check`.

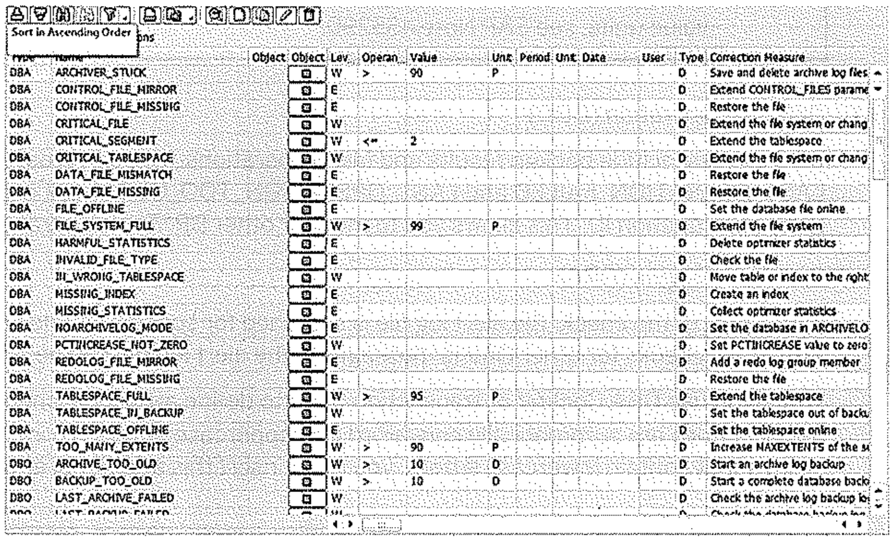
Parameters for the Database System Check

To perform the database system check, start BRGUI or BRTOOLS and choose *Check and verification → Database system check*. Alternatively, run the command `brconnect -f check` in the command line. The long names for `brconnect -f check` are identical to the input menus of BRTOOLS.

The following table lists command line options for brconnect – f check and the corresponding parameters in init<DBSID>.SAP for the database system check:

Option	Parameter	Meaning
`-o|-owner`	`check_owner`	BRCONNECT checks objects of all SAP owners, by default. Use this to restrict the checks to objects of certain owners, for example, in Multiple Components in One Database (MCOD) databases.
`-e|-exclude`	`check_exclude`	BRCONNECT checks all objects, by default. Use this to exclude checks on certain tables. Possible arguments are as follows: • <tables> • <indexes> • <tablespaces> • `all_part` to exclude non-SAP partitions (such as in SAP Business Information Warehouse (SAP BW) and SAP Advanced Planner and Optimizer (SAP APO)) • `non_sap` to exclude non-SAP objects (such as Oracle data dictionary tables)
`-d|-default`	None	Normally, check conditions are read from the *DBCHECKORA* table. Use this option so that BRCONNECT performs the check with the default conditions in the source code.

Check Conditions for BRCONNECT



Type	Name	Object	Object	Lev.	Operan.	Value	Unit	Period	Unit	Date	User	Type	Correction Measure
DBA	ARCHIVER_STUCK			W	>	90	P					D	Save and delete archive log files
DBA	CONTROL_FILE_MIRROR			E								D	Extend CONTROL_FILES parame
DBA	CONTROL_FILE_MISSING			E								D	Restore the file
DBA	CRITICAL_FILE			W								D	Extend the file system or chang
DBA	CRITICAL_SEGMENT			W	<=	2						D	Extend the tablespace
DBA	CRITICAL_TABLESPACE			W								D	Extend the file system or chang
DBA	DATA_FILE_MISMATCH			E								D	Restore the file
DBA	DATA_FILE_MISSING			E								D	Restore the file
DBA	FILE_OFFLINE			E								D	Set the database file online
DBA	FILE_SYSTEM_FULL			W	>	99	P					D	Extend the file system
DBA	HARMFUL_STATISTICS			E								D	Delete optimizer statistics
DBA	INVALID_FILE_TYPE			E								D	Check the file
DBA	IN_WRONG_TABLESPACE			W								D	Move table or index to the right
DBA	MISSING_INDEX			E								D	Create an index
DBA	MISSING_STATISTICS			E								D	Collect optimizer statistics
DBA	NOARCHIVELOG_MODE			E								D	Set the database in ARCHIVELO
DBA	PCTINCREASE_NOT_ZERO			W								D	Set PCTINCREASE value to zero
DBA	REDOLOG_FILE_MIRROR			E								D	Add a redo log group member
DBA	REDOLOG_FILE_MISSING			E								D	Restore the file
DBA	TABLESPACE_FULL			W	>	95	P					D	Extend the tablespace
DBA	TABLESPACE_IN_BACKUP			W								D	Set the tablespace out of backu
DBA	TABLESPACE_OFFLINE			E								D	Set the tablespace online
DBA	TOO_MANY_EXTENTS			W	>	90	P					D	Increase MAXEXTENTS of the s
DBO	ARCHIVE_TOO_OLD			W	>	10	D					D	Start an archive log backup
DBO	BACKUP_TOO_OLD			W	>	10	D					D	Start a complete database back
DBO	LAST_ARCHIVE_FAILED			W								D	Check the archive log backup lo

Figure 103: Customizing DBCHECKORA

BRCONNECT reads the checks to be performed from the table *DBCHECKORA*. To adapt the checks and their conditions, you can customize the table *DBCHECKORA* using transaction `DB17`.

You can use the following options to customize the check conditions:

- Add new conditions of type DBO, ORA, or PROF
- Exclude individual conditions from the check
- Specify threshold values for the conditions
- Create object-specific conditions to exclude them from the check
- Create object-specific conditions to set individual threshold values
- Specify corresponding corrective actions (this field is only for information and is displayed when the error occurs)
- Maintain the description of the conditions

To customize the table *DBCHECKORA*, you can use DBA Cockpit, choose *Alerts → Check conditions*, or directly call transaction `DB17`.

In the DBCHECKORA table, different types of checks defined are as follows:

- *Database administration (DBA)*

  DBA checks include database configuration, space management, and state of the database. Checks of this type cannot be added, because the actions these checks perform are hardcoded into BRCONNECT. However, you can copy existing check conditions and, therefore, specify special thresholds for specific tablespaces specified in the *OBJECT* field. To exclude objects from the check, use the `check_exclude` parameter in `init<DBSID>.sap`. To exclude whole checks, deactivate in the Customizing table *DBCHECKORA*.

> Note:
> For more information about possible test operands, threshold values, and value units, see SAP Note 435290.

- *Database operation* (DBO)

  DBO checks include backup results and failed operations. You can create new LAST_OPERATION_FAILED and LAST_OPERATION_TOO_OLD check conditions for any BR*Tools function ID. This allows you to monitor other BR*Tools operations.

- *Oracle messages* (ORA)

  In this part of the check, the alert log file is searched for specific Oracle error messages. Additional checks for other Oracle messages in the alert log file can be added.

  When you use DBA Cockpit or transaction `DB17` to define additional checks in the alert log file, all entries are automatically converted into uppercase letters. To include checks for strings that contain lowercase letters, you must insert this additional entry with SQL*Plus.

> Hint:
> See the online help for a detailed description of the SQL command.

- *Oracle profile parameters* (PROF)

  Profile parameters of Oracle are checked. These checks can be adapted to the current recommendations from SAP and new checks can be included.

> Hint:
> You can find current parameter recommendations for Oracle11g in SAP Note 1431798 – Oracle 11.2.0: Database Parameter Settings. For Oracle 10g, you can find these recommendations in SAP Note 830576 – Parameter recommendations.

Viewing the Output of the Database System Check

The detailed log of the database system check is written into the directory `$SAPDATA_HOME/sapcheck` (UNIX) or `%SAPDATA_HOME%\sapcheck` (Windows). The filename is an encoded timestamp and has the extension `.chk`.

You can view the detailed log with any text viewer or editor. In addition, BRTOOLS can be used to display the logfiles. Call BRTOOLS or BRGUI and choose *Additional functions → Show profiles and logs → BRCONNECT logs*.

The following information displays on the screen:

```
BR0658I List menu 52 - please select one entry
----------------------------------------------------------------------
--------
Display of BRCONNECT logs

Pos.  Log           Start                Function    Rc    Object

 1 =  cdwroebt.chk  2011-06-06 10.16.45  check        1
```



```
 2 - cdwroebh.cln  2011-06-06 10.16.33  cleanup    0
 3 - cdwroeao.sta  2011-06-06 10.16.14  stats      0   ALL
 4 - cdwrodki.chk  2011-06-06 10.09.12  check      1
 5 - cdwroczu.chk  2011-06-06 10.04.38  check      1
 6 - cdwrocuz.nxt  2011-06-06 10.02.33  next       0   ALL
 7 - cdwrobht.chk  2011-06-06 09.45.33  check      1
 8 - cdwrjngl.chk  2011-06-06 11.35.31  check      1
 9 - cdwrjmgk.chk  2011-06-06 11.24.14  check      3
10 - cdwrjmco.chk  2011-06-06 11.22.34  check      3
11 - connT99.log   <summary log>
```

To view the BRCONNECT logs from within the SAP system, use the DBA Cockpit (*Jobs → DBA Logs*) or transaction `DB14`. Viewing the logs with DBA Cockpit or `DB14` has the advantage that warnings and errors are automatically highlighted.

In addition to the output in the log file, the database system check also writes warnings and errors into the database table *DBMSGORA*. This table can be viewed with transaction `DB16` and is also read by the Computing Center Management System (CCMS) alert monitor (transaction `RZ20`) to check and report Oracle database alerts.

Interpreting the Detail Log of the Database System Check

The detail log of the database system check first contains information about the brconnect parameters, then information about the database objects.

The database objects are as follows:

- **Tablespaces and data files**

  This object is a list of all tablespaces and their data files. This list shows the status and the current size of all data files.

  - A plus sign (+) behind the status of a tablespace indicates that the corresponding tablespace is locally managed. A pound sign (#) behind the status of a tablespace indicates that this is a temporary tablespace. A minus sign (-) behind the status of a tablespace indicates that this is an undo tablespace.

  - A plus sign (+) behind the status of a data file indicates that the corresponding data file is auto extensible.

- **Redo log files**

  This object is a list of all redo files, their status, and size.

- **Control files**

  This object is a list of control files and their size.

- **Database disk volumes**

  This object is a list of disk volumes containing database data. For each main directory, the total and free space is shown, including the percentage of used space.

- **Tablespace fragmentation**

  This object lists statistical information for each tablespace, like number of data (files), tables, indexes, and extents, for each tablespace and current total, used, and free space within the table space.

  The columns *MaxSize[KB]*, *Used[%]*, and *Free[KB]* display the information about the tablespaces, keeping the auto extensibility of data files in mind. This information is identical to the current sizes for tablespaces not having auto extensible data files.

Tablespaces having auto extensible data files are indicated by a plus sign [+] behind the numbers. These numbers display the following information:

- Space information

  This is displayed, assuming each data file has enough disk space to grow up to its `MAX_SIZE` value.

- Maximum available disk space

  This is displayed if the disk space available is smaller than `MAX_SIZE`. In this case a *CRITICAL_FILE* alert is raised because a file system overflow can occur if Oracle extends the data file.

The last column Largest[KB] lists the five largest extents. A plus sign (+) indicates again that this size will be available if a data file is automatically extended.

- Check Conditions for Database Administration

  This list shows all check conditions from the *DBCHECKORA* table, including their threshold values, descriptions, and severity. It is also indicated whether the check is active.

The detail log ends with a list of all warnings, errors, or alerts detected and summary information.

Adapting the NEXT Extent Size

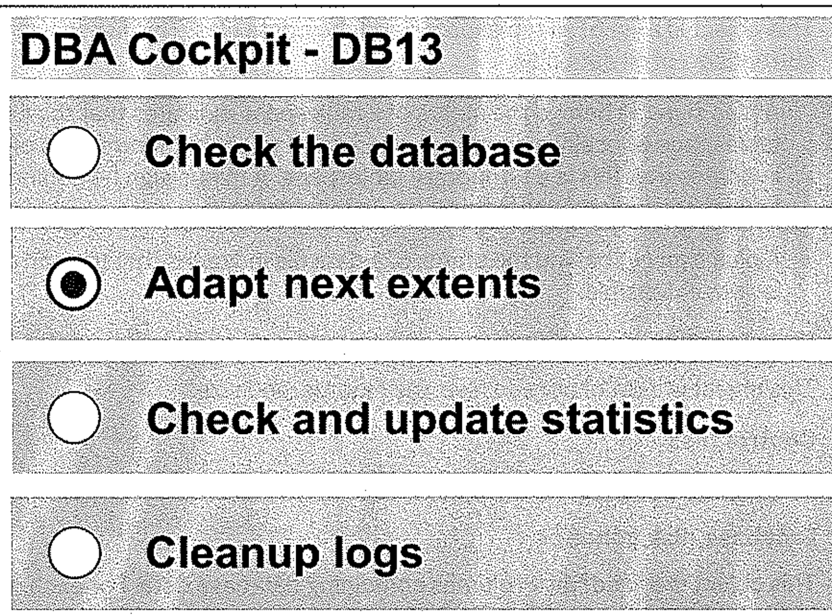


Figure 104: BRCONNECT Activities – Adapt Next Extents

In dictionary-managed tablespaces, the storage parameter `NEXT` determines the size of the next extent when a new extent must be allocated.

Execute the tool `brconnect -f next` regularly, for example, once per week, to adapt the NEXT storage parameter of all segments in dictionary-managed tablespaces.

Regularly executing the command brconnect -f next prevents the following situations:

- Prevents tables from having too many small extents. Tables that have too many small extents can produce the following negative effects:
  - Reduced performance
  - Increased chance that the total number of extents reaches the limit `MAXEXTENTS`
- Prevents tables from having a `NEXT` extent size that is too large. Having tables with a `NEXT` extent size that is too large can produce the negative following effects:
  - Wasted disk space
  - Fragmentation, because smaller contiguous free areas cannot be used
  - A tablespace overflow, if the largest free area is smaller than the NEXT extent size of a table

> Hint:
> By default, `brconnect -f next` checks for all tables and indexes if the `NEXT` storage parameter must be adapted. All objects in locally-managed tablespaces are automatically excluded. It is not necessary to manually restrict it to objects in dictionary managed tablespaces.
>
> Run `brconnect -f next` once to check whether there are tables available in data dictionary tablespaces to adapt the NEXT extent size. If the message `BR0906I All selected tables/indexes are in locally managed tablespaces` appears, you do not need to schedule `brconnect -f next`.

> Note:
> Normally, BRCONNECT is started using `brconnect -u / -c -f next -t all`.

The algorithm used to determine the optimal NEXT extent size ensures the following:

- The `NEXT` extent size is never changed to a value smaller than the current value and as specified in the SAP data dictionary (*TGORA* for tables and *IGORA* for indexes) for the corresponding size category of this table or index (stored in the SAP data dictionary table *DD09L*).
- The `NEXT` extent size is about 10% of the total size of the table or index, if the table is larger than its expected size.
- The `NEXT` size is smaller than the largest contiguous free area of the tablespace, if the tablespace is not autoextensible.

Parameters for Adapt NEXT Extent

In most cases, it is not necessary to specify additional parameters for `brconnect -f next`. Sometimes exceptions (for example, exclusion of tables) or special settings (for example, forcing a specific `NEXT` extent size for specific tables) are necessary. These settings can be made on the command line or made permanent by setting parameters in the profile `init<DBSID>.sap`.

The following table lists optional parameters for adapt NEXT extent:

Option	Parameters	Meaning
`-e|-exclude`	`next_exclude`	A list of tables or indexes for which the `NEXT` extent is not adapted. The keyword `all_part` excludes SAP partitions (such as in SAP BW and SAP APO)
`-s|-special`	`next_special`	Defines special settings for a list of tables or indexes. The syntax is `<object>:<size>[/<limit>]`; the parameters are as follows: • `<object>` is a table or Index • `<size>` is `NEXT` extent size for the `<object>` • `<limit>` is optional and specifies the `MAXEXTENTS` value for the `<object>`
`-m|-max`	`next_max_size`	Defines the maximum size for any `NEXT` extent. No extent will be adapted to be larger than this value.
`-l|-limit`	`next_limit_count`	Defines the maximum number of extents of a segment (`MAXEXTENTS`)
`-f|-force`	None	Forces to reduce the next extent size. The following arguments define the target `NEXT` size for the reduction: • `free` Reduces the `NEXT` extent size to the size of largest contiguous free area in the tablespace • `max` Reduces the `NEXT` extent size to the value of `next_max_size` • `both`

Option	Parameters	Meaning
		(default) Reduces the NEXT extent size to the smaller of the values free and max
-t|-table	next_table	A list of tables, indexes, or tablespaces to adapt the next extent; default – all tablespaces. Keyword special adapts only objects specified in the next_special parameter
-o|-owner	next_owner	Limits the adjustment to tables owned by certain owners; default – tables of all SAPR3 and SAP<SCHEMA-ID> users.

The detailed log of brconnect -f next is written into the directory $SAPDATA_HOME/sapcheck (UNIX) or %SAPDATA_HOME%\sapcheck (Windows). The file name is an encoded timestamp and has the extension .nxt. You can view it in the same way as the log of the database system check.

Checking and Updating of Database Statistics

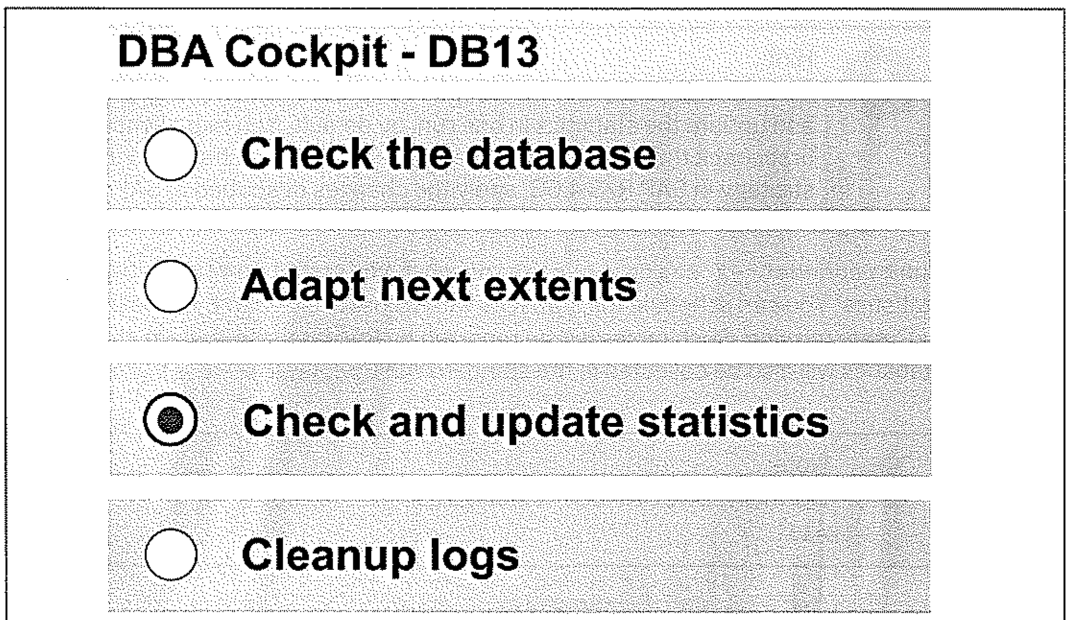


Figure 105: BRCONNECT Activities – Check and Update Optimizer Statistics

Oracle uses the cost-based optimizer to find the optimal access path of an SQL query. The cost-based optimizer relies on statistics about tables, which are stored in Oracle dictionary tables in the SYSTEM tablespace. If these statistics are not up to date, for example, because a table grew much larger, Oracle can use a more expensive access path when querying tables, which results in poor performance.

On SAP systems, the parameter `optimizer_mode` is set to `CHOOSE`. This means Oracle uses the cost-based optimizer when statistics are available. If no statistics are available, they are created when the table is accessed at runtime.

To update the statistics on an SAP system, use the tool `brconnect -f stats`. This tool checks all tables of the database to see if the statistics are out of date and updates the statistics, if required.

> Caution:
> Do not use other tools to update the statistics as these tools may not be aware of the SAP tables that must not have statistics.

To make sure that statistics are up to date, SAP recommends that regularly you run `brconnect -f stats -t all`. It can be scheduled it in the DBA Cockpit (transaction `DB13` – Check and Update Statistics). With Oracle 10, schedule the statistics run once a week. As table monitoring is active for all tables by default, the check phase is very quick and does not require much time.

> Hint:
> With older versions of Oracle, for example Oracle 8.1.7 and 9i, table monitoring is not active. During the statistics run, the check phase requires considerable time. Therefore, the statistics are scheduled once a week.

> Hint:
> Up until SAP R/3 4.6C, transaction `DB13` does not offer a single option to check and update the statistics in one run, by default. You must schedule two jobs – first check the statistics and then update them. To change to the new one-phase strategy in SAP R/3 4.6C, apply the changes described in SAP Note 403704.

Run `brconnect -f stats -t missing` when you create new tables that do not have statistics. You can schedule the statistics with transaction `DB20` by choosing *Global statistics* → *Create missing*. Such tables will be reported by the database system check. Missing statistics will also be collected in the standard run `brconnect -f stats -t all`.

> Note:
> Normally, BRCONNECT is started using `brconnect -u / -c -f stats -t all`.

The detailed log of `brconnect -f stats` is written into the directory `$SAPDATA_HOME/sapcheck` (UNIX) or `%SAPDATA_HOME%\sapcheck` (Windows). The file name is an encoded timestamp and has the extension `.sta`. You can view it in the same way as the log of the database system check.

Clean Up of Old Logs and Traces

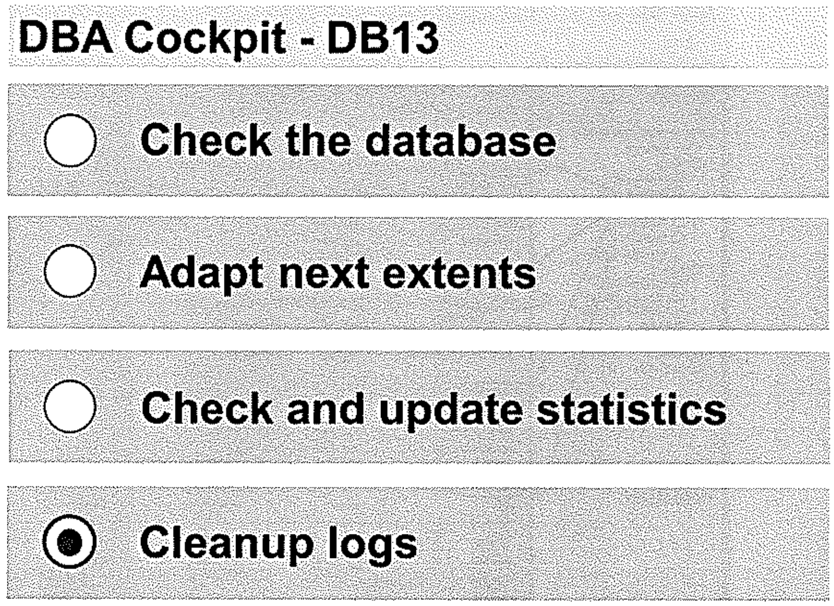


Figure 106: BRCONNECT Activities – Cleanup Logs

Over time, many log files and other intermediate files, such as backups on disks, are created. To clean up these logs, use `brconnect -f cleanup`.

This process cleans the following logs and files:

- Detailed logs of the BR*Tools
- Disk backups created by BRBACKUP and BRARCHIVE
- Export dumps and scripts created by BRSPACE
- Log records and check results in the tables *SDBAH*, *SDBAD*, *DBA**, and *DBMGSGORA*
- Oracle trace and audit files

> Note:
> Normally, BRCONNECT is started using `brconnect -u / -c -f cleanup`.

By default, all files older than 30 days are cleaned up and database log records older than 100 days are deleted. You can separately define another cleanup period for each object in the BR*Tools profile `init<DBSID>.sap` by removing the pound sign (#) from the corresponding `cleanup_*` entry and specifying another cleanup period:

```
...
# retention period in days for brbackup log files
# default: 30
# cleanup_brbackup_log = 30

# retention period in days for brconnect log files
# default: 30
# cleanup_brconnect_log = 30
...
```

Checking Database Growth

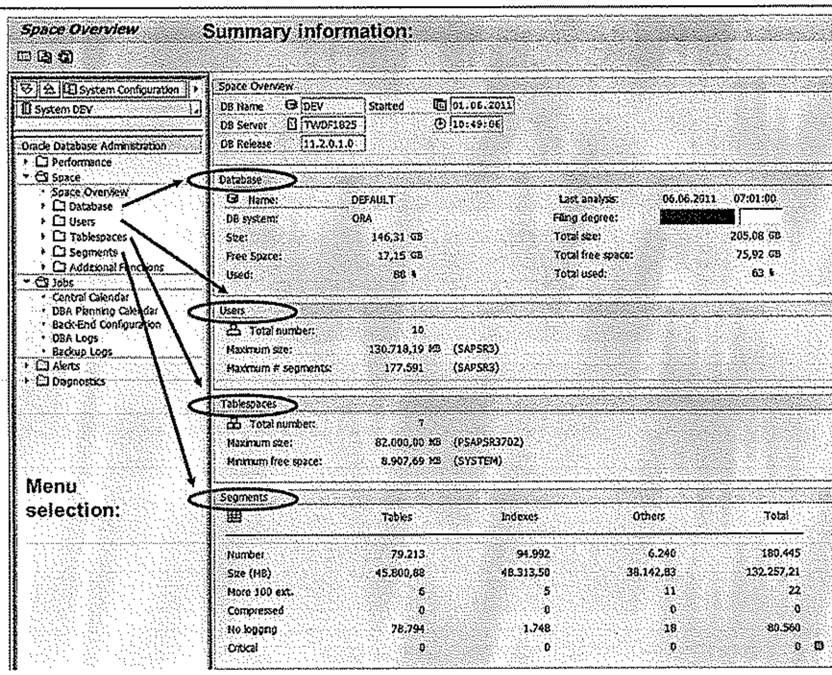


Figure 107: Checking Database Growth

The checks performed by the database system check provide a snapshot of the current database status when BRCONNECT runs. A DBA must know how fast a specific tablespace has grown to be able to plan new disks. Statistical data about the database growth can be displayed using the *Space* menu item in the DBA Cockpit (transaction `DB02`).

The figure shows the menu structure and memory space in the DBA Cockpit (transaction `DB02`). The *Space* menu in DBA Cockpit provides functions that help analyze the database performance. A historical overview is also available to analyze performance.

Overview of functions in the memory space analysis (Space menu) in the DBA Cockpit:

- *Space Overview*

  It displays information about the database size and free space.

- *Database*

  The *Overview* option displays memory space statistics of the entire database. Choose the *History* tab to view the total database growth per week or per month.

- *Users*

  The *Overview* option displays information about the database users and memory objects that are assigned to them. The *Detailed analysis* option displays additional information about the database users, such as user account status, assigned default tablespaces, and whether the user is blocked.

- *Tablespaces*

  The *Overview* option displays the current sizes and free space per tablespace. In addition, tablespace information, such as status (online, offline) and backup mode are also displayed.

When a tablespace has autoextensible data files, the total size to which this tablespace can grow by automatically extending all autoextensible data files is shown. In addition, detailed information about data files, such as the current size and the settings of the automatic enhancement, is displayed.

The *Free Space Analysis* function shows detailed information about free space in tablespaces, including fragments and free areas within the tablespace.

The database views *DBA_DATA_FILES* and *DBA_TEMP_FILES* are displayed.

The *Detailed analysis* option displays the additional detailed information about the individual tablespaces.

- *Segments*

  The *Overview* option displays the following information:

  - Overall size and number of all segment types in the database
  - Largest segments, segments that have the largest allocated number of extents
  - Segments with the largest growth

  All segments are listed whose next extent size is larger than the largest free space in the tablespace (Space Critical Objects).

  You can tell whether there is a critical situation in the following situations:

  - If the corresponding tablespace has at least one autoextensible data file and one of the data files can be automatically extended to a size that the next extent fits into the data file, the situation is considered a warning. As long as there is enough disk space available for any autoextensible data file to grow, no action has to be taken.
  - If the corresponding tablespace does not have autoextensible data files or no autoextensible data file can grow by the next extent size (because `MAXBYTES` is reached or the disk is full), the situation is critical. This is because adding a single row to this table can cause a tablespace overflow. In this case, either you need to enlarge the file system, if possible, or add another data file to enlarge the tablespace.

  The *Detailed analysis* option displays additional detailed information about the individual tablespaces. The *Detailed analysis (aggregated)* option displays compressed information about selected segments including assigned tables and indexes, partitions, and LOBs.

- *Additional Functions*

  The *Collector Logs* option displays information about the job that collects the data.

  The *BW Analysis* option allows you to display SAP Business Information Warehouse (SAP BW) objects that are assigned to the database.

Collecting Regular Statistics

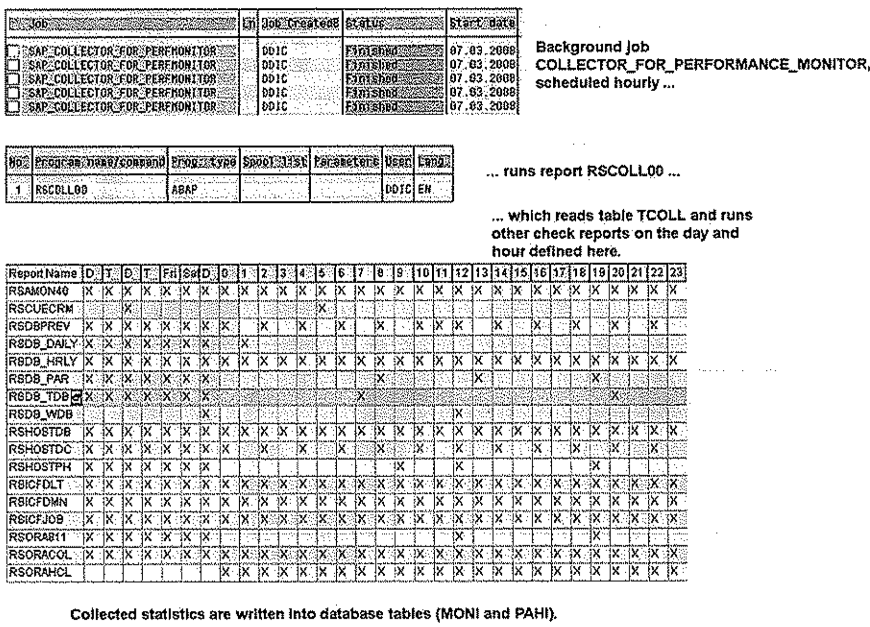


Figure 108: Collecting Regular Statistics

To collect several statistics about the database, the operating system (OS), and the SAP system, and to store the results in database tables for calculating statistics over a certain period of time, the *COLLECTOR_FOR_PERFORMANCE_MONITOR* background job is scheduled hourly by executing the ABAP report *RSCOLL00*.

The report *RSCOLL00* reads the table *TCOLL*, which contains a list of other reports and a matrix specifying the day of the week and hour of the day when this report is to be executed.

These reports collect their data and write information into the tables MONI (statistical data) and *PAHI* (parameters of OS, database, and SAP).

> Note:
> For more information about the reports in the table *TCOLL* as well as necessary customizing of *TCOLL* when several instances of an SAP system are running on one host, see SAP Note 12103.

Unit 3
Exercise 10

Perform Database System Checks

Business Example

You want to perform regular database checks.

In this exercise, a new tablespace is created and filled using SQL scripts. Using BRCONNECT, detect and describe any problems.

Connect to your database for this exercise.

1. Change to the subdirectory `D:\oracle\<SID>\scripts` and execute the command file `tscreate.bat`. The command file creates the tablespace PSAP<Schema-ID>EX and fills it with data, as follows.
2. Run the database check. What problems exist for the new tablespace?
3. Run `brconnect -f next` and check how it solves the issues.
4. Check the database again. Did the action of `brconnect -f next` permanently solve the issue?
5. Insert further rows into the SKEL table by executing the command file tsfill.bat. Check the database again to see whether you get a warning. If you execute tsfill.bat the second time, you can also see the Oracle error message indicating the tablespace overflow.

Perform Database System Checks

Business Example

You want to perform regular database checks.

In this exercise, a new tablespace is created and filled using SQL scripts. Using BRCONNECT, detect and describe any problems.

Connect to your database for this exercise.

1. Change to the subdirectory `D:\oracle\<SID>\scripts` and execute the command file `tscreate.bat`. The command file creates the tablespace PSAP<Schema-ID>EX and fills it with data, as follows.
   a)
   ```
   D:\oracle\T99>cd scripts

   D:\oracle\T99\scripts>tscreate.bat
   Creating the tablespace PSAPT99EX

   D:\oracle\T99\scripts>
   ```
2. Run the database check. What problems exist for the new tablespace?
   a) Run the database check directly from the command line or by using BRGUI or BRTOOLS.
   ```
   D:\oracle\T00>brconnect -c -f check
   ```
   b) Check the BRCONNECT log file. BRCONNECT detected that the largest free area within the data file is smaller than the next extent of the SKEL table.

   A part of the log file is as follows:
   ```
   BR0969I Checking database administration...
   BR0970W Database administration alert - level: WARNING, type:
   CRITICAL_SEGMENT,
   object: (table) SAPT99.SKEL, value: 504 KB * 1 / PSAPT99EX (>
   480/0/0/0/0 KB)
   ```
   c) Check the tablespace layout of the new tablespace PSAP<Schema- ID>EX. The BRCONNECT log file *.chk indicates that PSAP<Schema-ID>EX is a dictionary-managed tablespace (no + sign at tablespace status) with no autoextensible data file (no + sign at the status of any data file of PSAP<Schema-ID>EX).

   The log file displays the following information:
   ```
   BR0118I Tablespaces and data files

   Tablespace      Status      File               Status
   Id.         Size
   PSAPT99         ONLINE+     ...\T99.DATA1      ONLINE+    4
   20979712
   PSAPT99EX       ONLINE      ...\T99EX.DATA1    ONLINE
   6       2105344
   ```



```
PSAPT99USR        ONLINE+     ...\T99USR.DATA1     ONLINE      5
10493952
PSAPTEMP          ONLINE#     ...\TEMP.DATA1       ONLINE     -1
20979712
PSAPUNDO          ONLINE-     ...\UNDO.DATA1       ONLINE      3
20979712
SYSAUX            ONLINE+     ...\SYSAUX.DATA1     ONLINE      2
209723392
SYSTEM            ONLINE      ...\SYSTEM.DATA1     SYSTEM      1
419438592
```

d) While the tablespace PSAP<Schema-ID> is only 76.56% full (*Used[%]* column), the largest free area is 480 KB. The missing + sign in the last *Free[KB]* column indicates that there is no autoextensible data file.

The BRCONNECT log file displays the following information:

```
BR0983I Tablespace fragmentation

Tablespace   Files   Tables   Indexes   Extents     Total[KB]
Used[%]
PSAPT99          1        0         0         0
20480      5.00
PSAPT99EX        1        1         0         3
2048     76.56
PSAPT99USR       1       26        27        54
10240     43.75
PSAPTEMP         1        0         0         0
20480      0.00
PSAPUNDO         1        0         0         0
20480      5.00
SYSAUX           1      451       512      2398
204800     93.13
SYSTEM           1      625       709      2874
409600     45.13
Total:           7     1103      1248      5329
688128     55.75
```

3. Run `brconnect -f next` and check how it solves the issues.

   a) Run the following parameter:

   ```
   D:\oracle\T00>brconnect -c -f next
   ```

   `brconnect -f next` adapts the next extent size to exactly the size of the largest free area within the tablespace. By doing so, the table can grow by another extent.

   The BRCONNECT log file `*.nxt` contains the following message:

   ```
   BR0907I Checking and adapting next extent of tables and
   indexes...
   BR0917I Next extent size adapted for table SAPR3.SKEL, value old/
   new:
    504 KB / 480 KB
   ```

4. Check the database again. Did the action of `brconnect -f next` permanently solve the issue?

   a) No. The database check still issues a warning. The situation has become slightly better as a tablespace overflow now occurs only when the second next extent is added.

   ```
   WARNING, type: CRITICAL_SEGMENT, object: (table) SAPT99.SKEL,
   value: 480 KB * 2 / PSAPT99EX (> 480/0/0/0/0 KB).
   ```

5. Insert further rows into the SKEL table by executing the command file tsfill.bat. Check the database again to see whether you get a warning. If you execute tsfill.bat the second time, you can also see the Oracle error message indicating the tablespace overflow.

a) Run the command file `tsfill.bat`.

```
D:\oracle\T99\scripts>tsfill.bat
Filling the tablespace PSAPT99EX

SQL*Plus: Release 11.2.0.1.0 Production on Mo Jun 6 14:58:38 2011

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

Connected.
INSERT INTO SKEL  SELECT * FROM DBA_INDEXES
           *
ERROR at line 1:
ORA-01653: unable to extend table SAPT99.SKEL by 60 in
tablespace
PSAPT99EX


Disconnected from Oracle Database 11g Enterprise Edition Release
11.2.0.1.0 - 64
bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

D:\oracle\T99\scripts>
```

b) Check the database again to see whether you get the following warning.

```
D:\oracle\T00>brconnect -c -f check
```

The BRCONNECT log file contains the following message:

```
BR0969I Checking database administration...
BR0970W Database administration alert - level: WARNING, type:
TABLESPACE_FULL, o
bject: PSAPT99EX, value: 100.00% (> 95%)
BR0970W Database administration alert - level: WARNING, type:
CRITICAL_SEGMENT,
object: (table) SAPT99.SKEL, value: 480 KB * 1 / PSAPT99EX (>
0/0/0/0/0 KB)
BR0970W Database administration alert - level: ERROR, type:
MISSING_STATISTICS,
object: (table) SAPT99.SKEL
```

c) If you execute `tsfill.bat` the second time, you can also see the Oracle error message indicating the tablespace overflow. If you execute tsfill.bat the second time, you can also see the Oracle error message indicating the tablespace overflow, as shown below.

```
D:\oracle\T99\scripts>tsfill.bat
Filling the tablespace PSAPT99EX

SQL*Plus: Release 11.2.0.1.0 Production on Mo Jun 6 14:58:38 2011

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

Connected.
INSERT INTO SKEL  SELECT * FROM DBA_INDEXES
           *
ERROR at line 1:
ORA-01653: unable to extend table SAPT99.SKEL by 60 in
tablespace
PSAPT99EX
```



```
Disconnected from Oracle Database 11g Enterprise Edition Release
11.2.0.1.0 - 64
bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

D:\oracle\T99\scripts>
```

LESSON SUMMARY

You should now be able to:

- Perform a database system check

Unit 3
Lesson 3

Explaining Computing Center Management System (CCMS) Alert Monitor

LESSON OVERVIEW

This lesson explains the use of the Computing Center Management System (CCMS) alert monitor to monitor the Oracle database.

Business Example

You use the CCMS alert monitor daily to check if there are any warnings or problems within the SAP system. The database is also included in the alert monitor because you want to have a single transaction to view the health of all the components in your SAP system. For this reason, you require the following knowledge:

- An understanding of the purpose of the CCMS alert monitor
- An understanding of how to use the CCMS alert monitor to monitor the Oracle database

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Explain Computing Center Management System (CCMS) alert monitors

CCMS Alert Monitoring

The alert monitor in CCMS (transaction `RZ20`) allows you to centrally monitor different parts of the SAP system, including the Oracle database. You can configure analysis and data collection tools for different types of alerts. The database monitor can be found under the monitor set SAP CCMS Monitor Templates.

The alert monitor is also part of the DBA Cockpit (*Alerts → Alert Monitor*).

Oracle Database Monitoring

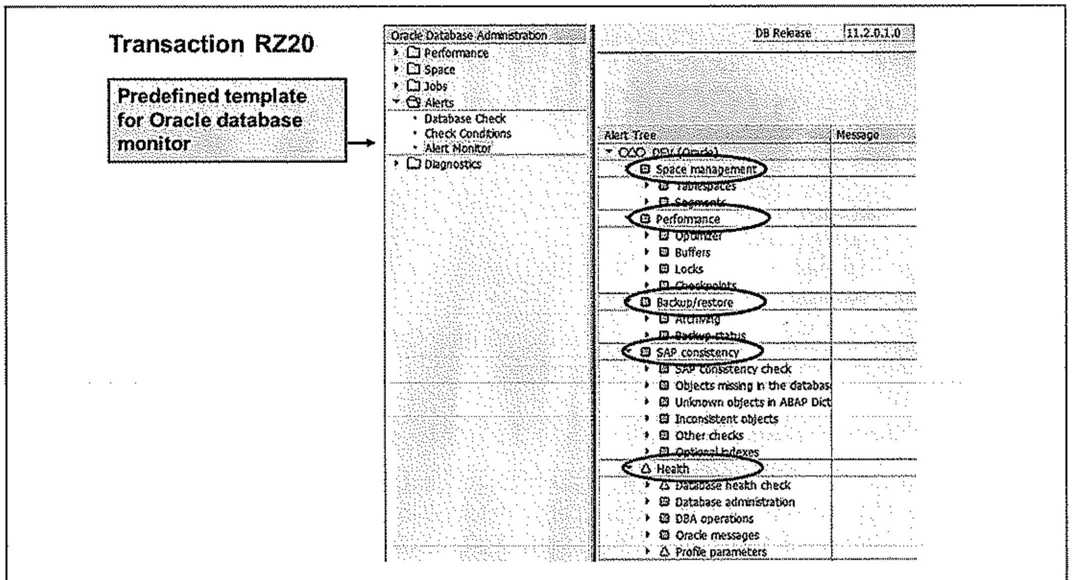


Figure 109: DBA Cockpit – Alert Monitor

You can use the CCMS alert monitor to check the following Oracle database functions:

- *Space management* – Tablespaces and segments
- *Performance* – Optimizer statistics, buffers, logs, and checkpoints
- *Backup/restore* – Database and redo log backups
- *SAP Consistency* – Between database objects in the ABAP and Oracle dictionaries
- *Health* – Database system checks from BRCONNECT

In the CCMS monitoring architecture, the monitors get their data from data collectors. The main data collector for the Oracle database functions within the CCMS alert monitor is BRCONNECT. It writes into the *DBMSGORA* table, which is examined by the CCMS alert monitor to display information, warnings, and alerts.

> Hint:
> When you enter new conditions of type ORA (Oracle error messages) or PROF (profile parameter check) in the table *DBCHECKORA* using transaction `DB17`, you must rebuild the monitoring tree to be able to see corresponding alerts in the CCMS alert monitor by performing the following steps:
>
> 1. Enter the new condition in the *DBCHECKORA* table using transaction `DB17`.
> 2. In transaction `RZ20`, open *SAP CCMS Monitor Templates.*
> 3. Double-click the *Database* monitor.
> 4. Activate the maintenance functions by choosing *Extras → Activate maintenance function*.
> 5. Position the cursor on the Oracle root MTE (Monitoring Tree Element) and reset all open alerts.
> 6. Select *Open alerts*.
> 7. Choose *Edit → Alerts → Reset alerts*.
> 8. Delete the Oracle monitoring tree by choosing *Edit → Delete*.
> 9. On the main application server, run the ABAP program *RSDBMON0* to rebuild the database monitoring tree.

For several other monitoring tree elements (MTEs) within the Oracle part of the CCMS alert monitor, the data collectors directly check the V$- and DBA views from Oracle.

> Caution:
> To avoid the expensive check of the view *DBA_SEGMENTS* and to display the information in the MTE, choose *Database → Oracle → Space management → Segments*. The system retrieves the segment alert information from the latest database system check. If this run is older than one day, the data is retrieved directly from the *DBA_SEGMENTS* view. This has a negative effect on performance. Therefore, you need to schedule at least one daily run of `brconnect -f check`.

LESSON SUMMARY

You should now be able to:

- Explain Computing Center Management System (CCMS) alert monitors

Unit 3

Learning Assessment

1. Oracle stores its data in tablespaces. A tablespace consists of one or more data files. A segment (for example, a table or an index) allocates space in a tablespace using one or more ____.

   Choose the correct answer.

   - [ ] A buffer
   - [ ] B extents
   - [ ] C memory slots
   - [ ] D tracks

2. Each tablespace consists of one or more data files. When creating data files, which of the following conventions are valid in an SAP environment?

   Choose the correct answers.

   - [ ] A Each data file is created in a separate subdirectory below the ORACLE_HOME directory.
   - [ ] B Each data file is created in a separate subdirectory of the sapdata<n> directory.
   - [ ] C Each data file is named like the tablespace, without the PSAP prefix and with the *.data<n> extension.
   - [ ] D Each data file is named like the biggest table it contains.

3. With locally-managed tablespaces, each data file has a _____ listing used and free blocks in the data file. The extent size will no longer be determined by storage parameters.

   Choose the correct answer.

   - [ ] A .TMP
   - [ ] B .GIF
   - [ ] C bitmap
   - [ ] D .TIFF

4. The statistical data about the database growth can be displayed using the menu item Space in the DBA Cockpit. Which of the following information is shown?

   Choose the correct answers.

   - [ ] A Memory space statistics of the entire database
   - [ ] B Information about the overall size of the alert log file
   - [ ] C Information about the database users and memory objects that are assigned to them
   - [ ] D Information about the largest segments, segments that have the largest allocated number of extents, and segments with the largest growth
   - [ ] E Information such as overall size and number of all segment types

5. Which of the following checks are performed when executing a database check using the BRCONNECT?

   Choose the correct answers.

   - [ ] A Database operation checks include backup results and failed operations.
   - [ ] B The system searches the alert log file for specific Oracle error messages.
   - [ ] C Database administration checks include database configuration, space management, and state of the database.
   - [ ] D The developer checks the SAP profile parameters.
   - [ ] E The system scans the Developer Trace files of the work processes for specific Oracle error messages.

6. Which of the following actions must you perform for optimally monitoring the Oracle database within the Computing Center Management System (CCMS) alert monitor?

   Choose the correct answers.

   - [ ] A Maintain check conditions in the DBCHECKORA table
   - [ ] B Schedule the database system check to run daily
   - [ ] C Maintain thresholds in the CCMS alert monitor itself
   - [ ] D Schedule RSDBMON0 report to run daily

Learning Assessment - Answers

1. Oracle stores its data in tablespaces. A tablespace consists of one or more data files. A segment (for example, a table or an index) allocates space in a tablespace using one or more ____.

   Choose the correct answer.

   [ ] A buffer

   [X] B extents

   [ ] C memory slots

   [ ] D tracks

2. Each tablespace consists of one or more data files. When creating data files, which of the following conventions are valid in an SAP environment?

   Choose the correct answers.

   [ ] A Each data file is created in a separate subdirectory below the ORACLE_HOME directory.

   [X] B Each data file is created in a separate subdirectory of the sapdata<n> directory.

   [X] C Each data file is named like the tablespace, without the PSAP prefix and with the *.data<n> extension.

   [ ] D Each data file is named like the biggest table it contains.

3. With locally-managed tablespaces, each data file has a _____ listing used and free blocks in the data file. The extent size will no longer be determined by storage parameters.

   Choose the correct answer.

   [ ] A .TMP

   [ ] B .GIF

   [X] C bitmap

   [ ] D .TIFF

4. The statistical data about the database growth can be displayed using the menu item Space in the DBA Cockpit. Which of the following information is shown?

   Choose the correct answers.

   - [X] A Memory space statistics of the entire database
   - [ ] B Information about the overall size of the alert log file
   - [X] C Information about the database users and memory objects that are assigned to them
   - [X] D Information about the largest segments, segments that have the largest allocated number of extents, and segments with the largest growth
   - [X] E Information such as overall size and number of all segment types

5. Which of the following checks are performed when executing a database check using the BRCONNECT?

   Choose the correct answers.

   - [X] A Database operation checks include backup results and failed operations.
   - [X] B The system searches the alert log file for specific Oracle error messages.
   - [X] C Database administration checks include database configuration, space management, and state of the database.
   - [ ] D The developer checks the SAP profile parameters.
   - [ ] E The system scans the Developer Trace files of the work processes for specific Oracle error messages.

6. Which of the following actions must you perform for optimally monitoring the Oracle database within the Computing Center Management System (CCMS) alert monitor?

   Choose the correct answers.

   - [X] A Maintain check conditions in the DBCHECKORA table
   - [X] B Schedule the database system check to run daily
   - [X] C Maintain thresholds in the CCMS alert monitor itself
   - [ ] D Schedule RSDBMON0 report to run daily

UNIT 4 Space Management

Lesson 1

Administering Tablespaces	270
Exercise 11: Performing Tablespace Administration	285

Lesson 2

Performing Reorganization of Tables	292
Exercise 12: Work with Table Reorganization	313

Lesson 3

Housekeeping and Troubleshooting	318
Exercise 13: Perform Housekeeping and Troubleshooting	327

UNIT OBJECTIVES

- Administer tablespaces
- Reorganize tables
- Perform housekeeping and troubleshooting

Unit 4
Lesson 1

Administering Tablespaces

LESSON OVERVIEW

This lesson explains how to create and administer tablespaces.

Business Example

While monitoring the database, you recognized that the PSAP<Schema-ID> tablespace is more than 98% full. You check the tablespace statistics and realize that if the tablespace growth is the same as in the last month, the tablespace would be 100% full in about four weeks. You decide to install an additional disk and extend the tablespace by adding a new data file because the disks containing tablespaces are already nearly full. For this reason, you require the following knowledge:

- An understanding of how to create new tablespaces
- An understanding of how to extend existing tablespaces
- An understanding of how to move or rename data files
- An understanding of the additional database space management options

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Administer tablespaces

Tablespace Administration

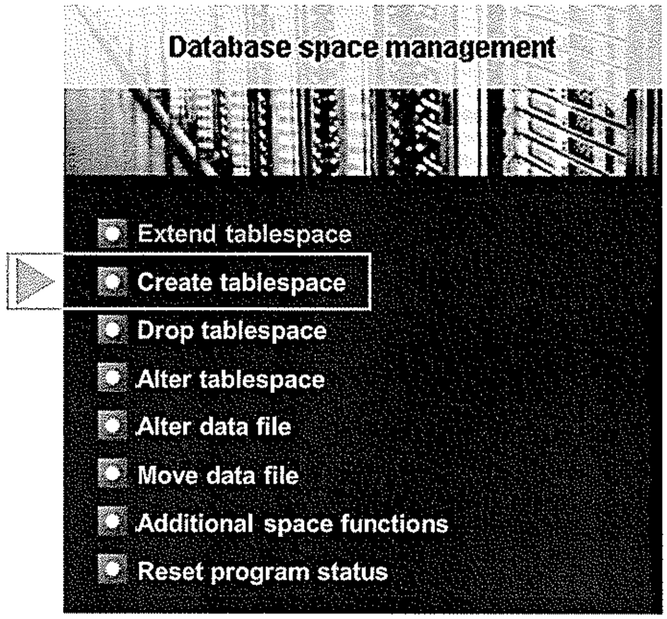


Figure 110: Space Management with BR*Tools

The main activities in tablespace administration are as follows:

- Creation of a new tablespace, for example, as a preparation to move certain tables out of the standard tablespace PSAP<SCHEMA-ID> into a special tablespace
- Deletion of a tablespace, for example, after an upgrade
- Extension of tablespaces by adding a data file or temp file or by resizing existing data files
- Movement of data files, for example, after adding a new disk, a data file that was created on another disk needs to be moved to the new disk

Creating and Dropping Tablespaces

In a normal operation, creating a new tablespace or dropping an existing tablespace is not necessary.

For example, the creation of a new tablespace is necessary for the following tasks:

- In preparation for an upgrade, where the old ABAP programs in PSAP<SCHEMA-ID><Old-Rel> are deleted and the new ABAP programs are imported into the new tablespace PSAP<SCHEMA-ID><New-Rel>
- In preparation for an online reorganization of a tablespace, for example, to switch from a dictionary-managed to a locally-managed tablespace

To create a new tablespace, perform the following steps:

1. Start BRTOOLS or BRGUI and choose *Space management → Create tablespace*.
2. To enter the main menu mode of BRSPACE, choose *Continue* on the next screen.

3. Choose *Create tablespace* from the BRSPACE menu.

   As an alternative, you can start `brspace -f tscreate` and choose *Continue*.

4. Fill in the necessary information by choosing the appropriate menu items.

Creation of a Tablespace Using BR*Tools

The following menu appears when you create a tablespace using BR*Tools:

```
BR0657I Input menu 305 - please enter/check input values
-----------------------------------------------------------------------
--------
Main options for creation of tablespace in database T99

 1 - Tablespace name (tablespace) ......... [PSAPT99NEW]
 2 - Tablespace contents (contents) ....... [data]
 3 - Segment space management (space) ..... [auto]
 4 # Database owner of tablespace (owner) . []
 5 # Table data class / tabart (class) .... []
 6 - Data type in tablespace (data) ....... [both]
 7 # Joined index/table tablespace (join) . []
 8 ~ Uniform size in MB (uniform) ......... []
 9 - Tablespace compression (compress) .... [no]
10 - Tablespace encryption (encrypt) ...... [no]
11 # Encryption algorithm (algorithm) ..... []

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-----------------------------------------------------------------------
--------
BR0662I Enter your choice:
```

The main options and their possible entries for creation of a tablespace are as follows:

- *tablespace*

  Enter the name of the tablespace to be created. According to SAP conventions, the tablespace name should start with the letters PSAP, followed by the SCHEMA-ID, plus a unique name.

- *contents*

  The values for contents are as follows:

  - `data`

    For normal tablespaces containing tables and/or indexes

  - `temp`

    For a temporary tablespace

  - `undo`

    For an undo tablespace

  > Hint:
  > A rollback tablespace has the content `data` but the tablespace should only contain rollback segments.

- *space*

You can select either `auto` for automatic segment space management (default) or `manual` for manual segment space management.

> Hint:
> Do not mix automatic or manual segment space management with locally or dictionary-managed tablespaces.
>
> Automatic segment space management is a new feature of Oracle 9i, which offers better performance on parallel queries on a segment.
>
> You cannot create dictionary-managed tablespaces with BRSPACE due to its disadvantages. BRSPACE always creates locally-managed tablespaces.

- *Class*

  This option is used for the table data type when you create a tablespace.

  The following values trigger data types of all or just specified tablespaces to be transferred during an online reorganization:

  - `all`

    This value transfers all table data types and it is used to reorganize the main tablespace `(PSAP<Schema-ID>)`.

  - `old_tsp | old_tsp_list`

    The new tablespace transfers the table data type(s) from one or several old tablespaces. It is used for the reorganization of one or more tablespaces shifting the tables to a new tablespace.

  This is relevant when reorganizing tables and simultaneously shifting the tables to another tablespace (see SAP Note 646681).

  The explicitly specified data types, `tab_class |tab_class_list` must start with the letters Y or Z (customer data types) and must not be longer than five characters. These data types should be used for customer developments or when shifting individual tables.

  If you do not set the option, a customer data type starting with the letter U is automatically generated by BRSPACE.

- *owner*

  Changing this is only possible if you have multiple components in one database (MCOD). The owner of the database schema for which you create the tablespace is SAP<SCHEMA-ID>.

- *data*

  In the old tablespace layout, a tablespace contained either tables (`table`) or indexes (`index`). In the new MCOD tablespace layout, tablespaces contain both tables and indexes. SAP recommends to create new tablespaces containing data and indexes to simplify administration.

- *join*

  If you have entered either `table` or `index` in the *data* field, here you can enter the corresponding index or table tablespace. If you enter `table`, BRSPACE creates both tablespaces. If you enter `index`, BRSPACE creates only an index tablespace and joins it with existing table tablespace. When you enter `both` in the *data* field, this option is not selectable.

If you choose *Continue*, the menu for the data file properties displays.

Options for Creating a Tablespace

The following menu displays the options for creating a tablespace:

```
BR0657I Input menu 306 - please enter/check input values
-----------------------------------------------------------------------
--------
Space options for creation of tablespace PSAPT99NEW (1. file)

 1 - Tablespace file name (file) .......... [D:\oracle
\T99\sapdata3\t99new_1\t99
new.data1]
 2 # Raw disk / link target (rawlink) ..... []
 3 - File size in MB (size) ............... [2]
 4 - File autoextend mode (autoextend) .... [no]
 5 # Maximum file size in MB (maxsize) .... []
 6 # File increment size in MB (incrsize) . []
 7 - SQL command (command) ................ [create tablespace
PSAPT99NEW
extent
 management local autoallocate segment space management auto datafile
'D:\oracle
\T99\sapdata3\t99new_1\t99new.data1' size 2M autoextend off]

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-----------------------------------------------------------------------
--------
BR0662I Enter your choice:
```

The screenshot shows the output after changing autoextent to yes.

The main options for creation of a tablespace and their possible entries are as follows:

- *file*

  This option is the name of the new data file for the tablespace. The default path and file name follows the SAP naming conventions for data files. You can create a tablespace with up to five data files using BRSPACE.

> Hint:
> You can use the number of the `sapdata<n>` directory as a shortcut for the file name, for example, 4 generates a new tablespace file name, which is located in the `sapdata4` directory.
>
> To create a tablespace with more than five data files, create the tablespace and then extend the tablespace by adding additional data files.

- *rawlink*

  This option is a soft link to a raw disk or to an external (non-SAP data) directory if there is no free space available in the sapdata directory.

- *size*

This option is the size of the data file in MB.

- *autoextend*

  If set to `yes`, the new data file is created autoextensible. In this case, `maxsize` and `incrsize` must be specified. If set to `no`, `maxsize` and `incrsize` cannot be specified and their entries are locked.

- *maxsize*

  For autoextensible data files, this parameter specifies the maximum size, in MB, up to which the data file can be increased.

- *incrsize*

  For autoextensible data files, this parameter specifies the size, in MB up to which the data file is automatically increased when necessary.

Now the tablespace is created. As creating a tablespace is a structural change in the database, a control file backup is created in the directory `$SAPDATA_HOME/sapreorg/<encoded timestamp>` before and after the creation of the tablespace. In addition, the action is reported in `$SAPDATA_HOME/sapreorg/struc<DBSID>.log`.

Dropping a Tablespace Using BR*Tools

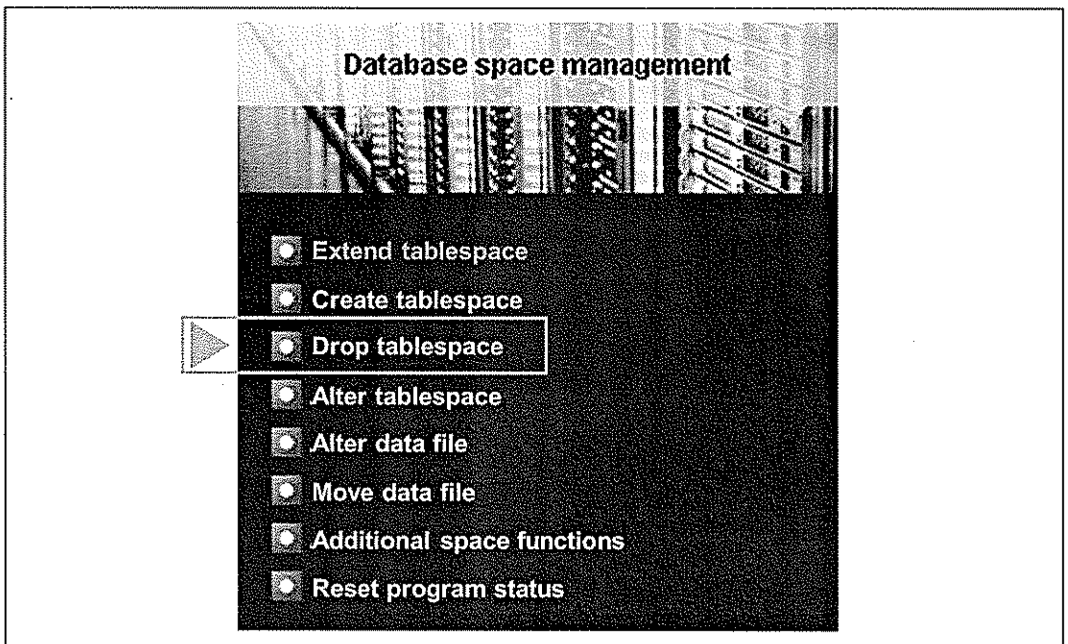


Figure 111: Dropping a Tablespace with BR*Tools

To drop a tablespace, perform the following steps:

1. Start BRTOOLS or BRGUI and choose *Space management* → *Drop tablespace*.
2. To enter the main menu mode of BRSPACE, choose *Continue* on the next screen.
3. Choose *Drop tablespace* from the BRSPACE menu.
4. As an alternative, you can start `brspace -f tsdrop` and enter *continue*.

A list of tablespaces is shown, from where you can select the tablespace to be dropped. After selecting the tablespace, you can set the *force* option (default: no). By default, BRSPACE does not drop a tablespace that is not empty. Using the *force* option, BRSPACE drops the tablespace even if it is not empty.

When a tablespace is dropped, BRSPACE performs the following steps:

1. Creates a control file backup in the directory `$SAPDATA_HOME/sapreorg/<encoded timestamp>` before and after the deletion
2. Checks whether the tablespace is empty (If the tablespace is not empty, it will not be dropped unless you force it by setting the force mode to yes.)
3. Takes the tablespace offline
4. Drops the tablespace including data files
5. Removes all subdirectories for the data files
6. Creates entries for the delete process in `struc<DBSID>.log`

Enlarging a Tablespace

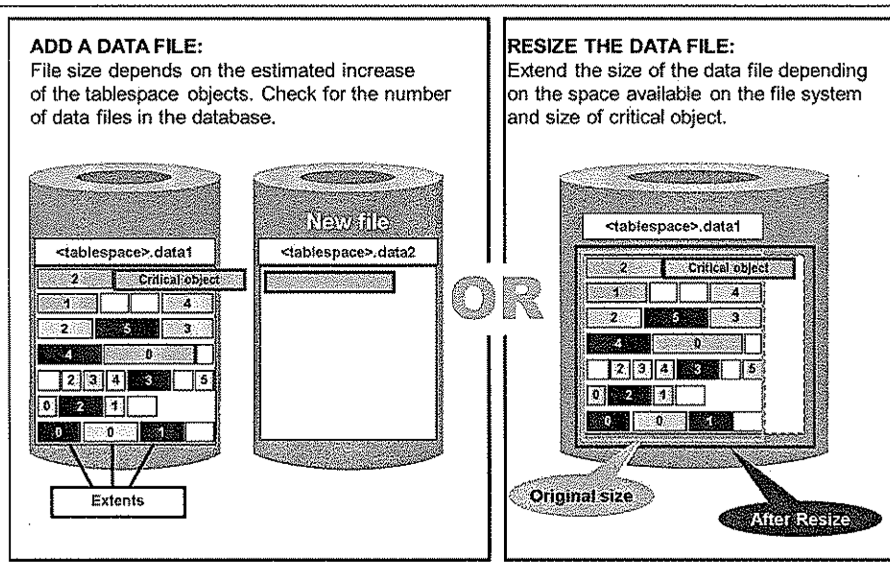


Figure 112: Enlarging a Tablespace

You can enlarge a tablespace in the following ways:

- Add a new data file to the existing tablespace.

  Use this option when the existing data files cannot be resized because there is no disk space available on the disks holding the existing data files, or because the existing data files have already reached their maximum size and you do not want to make them larger.

- Change the properties of an existing data file to be autoextensible.

  Use this option to simplify further tablespace administration in the future. Any autoextensible data file grows automatically, when needed, up to a maximum file size defined.

> Caution:
> When using autoextensible data files, you must monitor the disk space usage. In this case, a tablespace overflow can still occur if the disks holding these data files are full.

- Resize an existing data file.

  Use this option if you want to keep control over the data file growth.

You can create a new data file under the menu option *Extend tablespace* and you can find the properties and size of the data file under the menu option *Alter data file.*

Extending a Tablespace Using BR*Tools

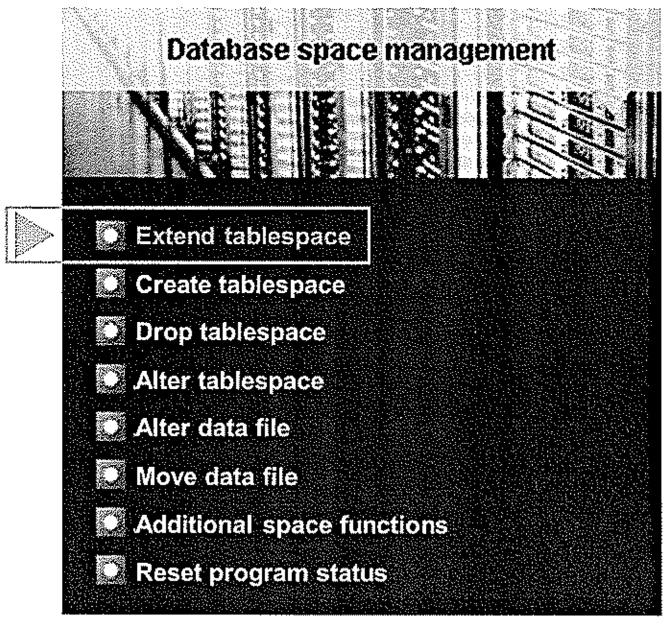


Figure 113: Extending a Tablespace with BR*Tools

To enlarge a tablespace by adding a new data file, perform the following steps:

1. Start BRTOOLS or BRGUI and choose *Space management* → *Extend tablespace*.
2. To enter the main menu mode of BRSPACE, choose *Continue* on the next screen.
3. Choose *Extend tablespace* from the BRSPACE menu.
4. As an alternative, you can start `brspace -f tsextend` and choose *Continue*.

After choosing the tablespace to be extended, a menu is shown, which is similar to the menu when creating a tablespace. Enter the appropriate values as described when creating a tablespace.

Changing the Attributes of Data Files

Figure 114: Changing the Attributes of Data Files Using BR*Tools

The attributes of data files that you can change using BR*Tools are as follows:

- *Resize data file*

  To resize an existing data file, perform the following steps:

  1. Start BRTOOLS or BRGUI and choose *Space management → Alter data file → Resize data file*.
  2. To enter the main menu mode of BRSPACE, choose *Continue* on the next screen.
  3. As an alternative, you can use `brspace -f dfalter` and choose *Resize data file*.
  4. Choose the data file from the list. Specify the new data file size and continue.

- *Turn on and maintain autoextend*

  To switch an existing file to be autoextensible, perform the following steps:

  1. Start BRTOOLS or BRGUI and choose *Space management → Alter data file → Turn on and maintain autoextend*.
  2. To enter the main menu mode of BRSPACE, choose *Continue* on the next screen.
  3. As an alternative, you can use `brspace -f dfalter` and choose *Turn on and maintain autoextend*.
  4. Choose the data file from the list.

  The list only contains all data files because you can change `maxsize` and `incrsize` with this function. On the next screen, the parameters for automatic extension of the data file have to be entered (`maxsize, incrsize`).

- *Turn off autoextend*

  To reset an existing file to be autoextensible, start BRTOOLS or BRGUI and choose *Space management → Alter data file → Turn off autoextend*.

- *Rename data file*

  The existing function for shifting data files is only available when you change the sapdata directory. This means that it is not possible to shift and rename data files in a sapdata directory. Use the new action *Rename data file* in the BRSPACE function *Alter data file* to do this.

> Hint:
> You can also enter the new name interactively in the *Options for alter of data file* menu.

> Caution:
> The new name for the database file should correspond to the SAP naming conventions.

- *Drop empty data file*

  If a data file is created with an invalid size or in the wrong directory, you can make the necessary correction using RESIZE or RENAME. If you want to drop the data file again, the following options are available:

  - Up to and including Oracle 9i, you can only delete a created data file during tablespace reorganization. There are no other possible options.
  - As of Oracle 10g, you can also drop an empty data file.

  BRSPACE now supports this feature using a new action *Drop empty data file* within the function *Alter data file* (see SAP Note 592393).

> Hint:
> If extents still exist in the data file, this command fails with ORA-03262. In this case, the relevant segments must first be stored so that the extents can be released.

Moving or Renaming a Data File Using BR*Tools

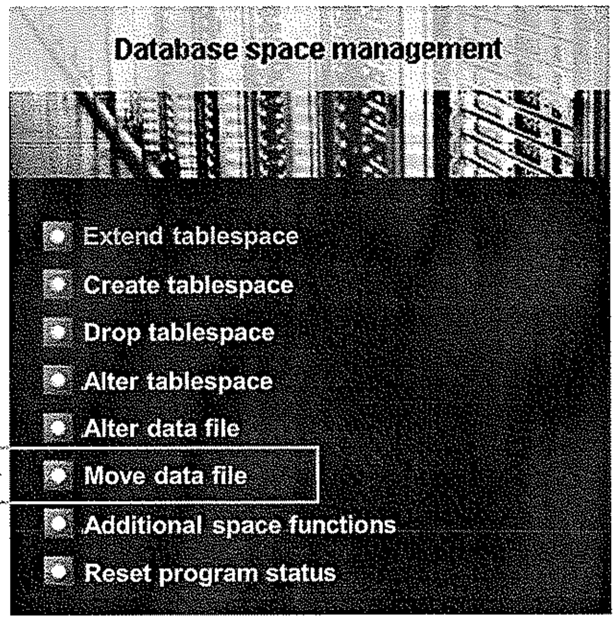


Figure 115: Moving or Renaming a Data File with BR*Tools

When moving or renaming a data file, the new file name or location must be added to the Oracle control file. This must be done using Oracle commands using BRSPACE.

You can move an existing data file to another location for the following reasons:

- To move data files from a file system to raw devices or from raw devices to a file system (on UNIX)
- To extend a tablespace by adding a new data file, but because of lack of disk space in the disks containing sapdata directories, the data file was created on another disk, which was not intended to hold data files (after adding another disk you want to move this data file to the new disk)
- To replace a disk on Windows because of different drive letters used

> Caution:
> BRSPACE can move only those data files that were created according to SAP conventions. To move a data file not created in a sapdata directory, you must move the data file with Oracle tools.
>
> To create a data file on a non-standard sapdata disk, create the directory path
>
> <drive:>\oracle\<SID>\sapdata<n> on Windows and then create the new data file there.

To move or rename a data file, perform the following steps:

1. Start BRTOOLS or BRGUI and choose *Space management* → *Move data file*.
2. To enter the menu mode of BRSPACE, choose *Continue* on the next screen.

3. As an alternative, you can start `brspace -f dfmove` and choose *Continue*.
4. Choose the data file to move from the list of data files.

> Hint:
> You can select one or more data files, separated by a comma or a range of data files separated by a hyphen.

On the "Data file move menu" screen, enter the following required information:

- `destination`

  Enter the full path of the sapdata directory where you want to move the data file. This directory must already exist. Do not enter the subdirectory for the data file because BRSPACE creates the subdirectory.

- `parallel`

  When moving more than one data file, you can perform the copy process in parallel. Enter the number of copy processes here.

- `force`

  To move or rename a data file, the database must be shut down. BRSPACE will only shut down the database if SAP is not running. You can force BRSPACE to shut down the database, even when SAP is connected, by setting the force option.

Altering a Tablespace Using BR*Tools

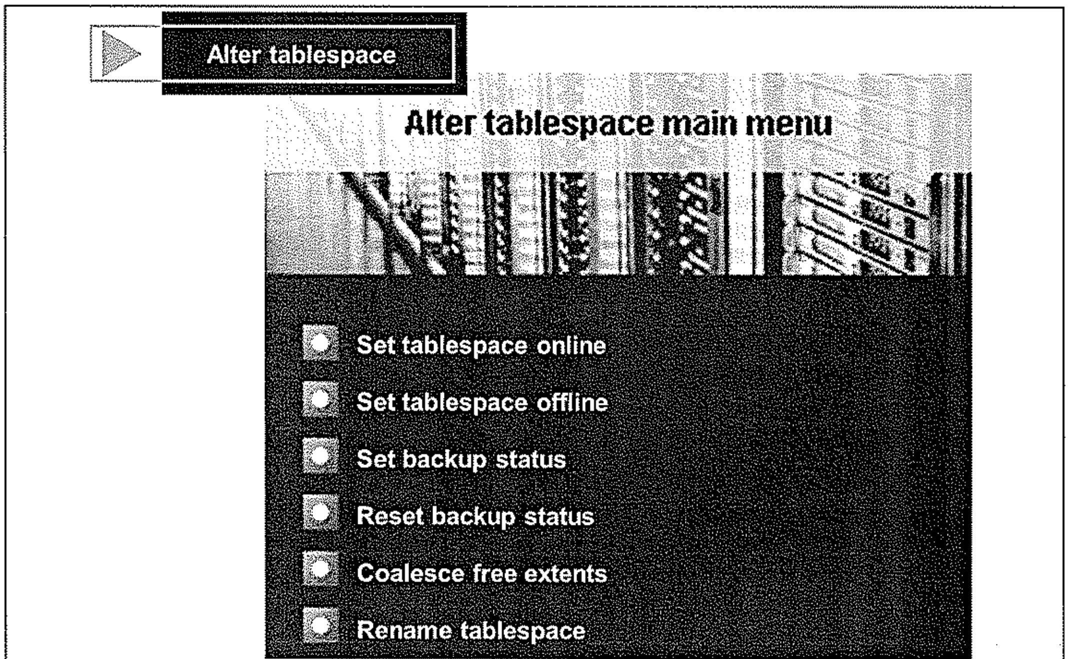


Figure 116: Altering a Tablespace with BR*Tools

To alter a tablespace, start BRTOOLS or BRGUI and choose *Space management → Alter tablespace*.

The Alter tablespace main menu displays the following actions:

- *Set tablespaces offline* and *Set tablespaces online*

  When a tablespace is set to offline, it cannot be accessed anymore. For SAP to work correctly, all tablespaces belonging to the corresponding database schema user SAP<SCHEMA-ID> must be online.

  There are situations when Oracle sets a tablespace to offline. This happens, for example, when Oracle receives an I/O error from the operating system when writing into a data file. To avoid corrupt blocks from being written, Oracle sets this tablespace to offline.

  > Caution:
  > Whenever a tablespace is offline and it is not actively set to offline by a database administrator, you must find the reason for this and solve the problem causing Oracle to set this tablespace to offline. After finding and resolving the problem, the tablespace can be set to online using this menu.

  In databases with MCOD, the tablespaces belonging to one schema user SAP<SCHEMA-ID> can be set offline to perform certain administrative actions without affecting other SAP systems using a different SAP<SCHEMA-ID> user. But even in this situation, the tablespaces *SYSTEM, PSAPROLL/PSAPUNDO*, and *PSAPTEMP* must not be set offline, because these tablespaces are used by all schema users.

- *Set backup status* or *Reset backup status*

  The backup status is regularly set by BRBACKUP when performing an online backup and reset after the backup. When BRBACKUP crashes during an online backup, one or more tablespaces remain in backup mode. This situation is checked by the database system check, which reports the following warning if a tablespace is in backup mode:

  ```
  BR0970W Database administration alert - level: WARNING, type:
  TABLESPACE_IN_BACKUP, object: PSAPT99.
  ```

  In this case, a normal shutdown would not execute. Also, if the database crashes, or if a shutdown abort is performed while a tablespace is in backup mode, a manual recovery is required before the database can be reopened.

  To avoid a manual recovery, reset the backup status when a tablespace is in backup mode when the following conditions are true:

  - You notice that an online backup has crashed.
  - There is currently no online backup running.

- *Coalesce free extents*

  Coalescing free extents means that several free extents that directly follow each other are combined to a single, larger free extent. The tool database system check will automatically coalesce free extents for all tablespaces. You must manually coalesce free extents using this menu when a high amount of data is deleted in the database, for example, after archiving data or the deletion of a client.

- *Rename Tablespace*

  As of Oracle 10g, the name of tablespaces can be changed. BRSPACE supports this feature using a new action *Rename tablespace* in the function *Alter tablespace*.

  The data files are also renamed. This action requires that the new tablespace is set to offline in the short term, which can affect the SAP system. Renaming the tablespace can also be set to run at a specific time when the system has a minimum load (see SAP Note 914174).

Additional Database Space Management Options

To show database information, start BRTOOLS or BRGUI and choose *Space management → Additional space functions*.

You can view information about tablespace, data files, redo log, and control files. In addition, you can display information about the disk usage of all directories with database data.

Unit 4
Exercise 11

Performing Tablespace Administration

Business Example

The database system check shows that a tablespace is nearly full. You want to enlarge the tablespace to provide additional space.

Enlarge tablespace *PSAP<DBSID>EX* to provide additional space and create a new locally-managed tablespace.

1. Extend tablespace *PSAP<DBSID>EX* by resizing its data file to 4 MB. Use the database system check to see whether enough free space is available after this action.

> Note:
> Tablespace *PSAP<DBSID>EX* was created in the previous exercise.

2. Set the data file of tablespace *PSAP<DBSID>EX* to autoextensible. The maximum size is 10 MB; the increment size is 2 MB. Check the size of the data file and run the `tsfill.bat` script. Check the size of the data file again; it should have grown by 2 MB.

3. Create a new locally-managed tablespace, PSAP<DBSID>NEW, with one autoextensible data file with a size of 4 MB, increment size 2 MB, and maximum size of 20 MB.

Unit 4
Solution 11

Performing Tablespace Administration

Business Example

The database system check shows that a tablespace is nearly full. You want to enlarge the tablespace to provide additional space.

Enlarge tablespace *PSAP<DBSID>EX* to provide additional space and create a new locally-managed tablespace.

1. Extend tablespace *PSAP<DBSID>EX* by resizing its data file to 4 MB. Use the database system check to see whether enough free space is available after this action.

> Note:
> Tablespace *PSAP<DBSID>EX* was created in the previous exercise.

a) Start BRGUI or BRTOOLS and choose *Space management → Alter data file*.

b) Choose *Continue*.

c) Choose *Resize data file*.

d) Choose *Continue* until the following list of data files displays:

```
BR0659I List menu 315 + please select one or more entries
-------------------------------------------------------------------
--------------
List of data files for alter

 Pos.  Tablespace        Status  Type      Size[KB] AuExt. File

   1 - PSAPT99           ONLINE  FILE         20480   YES    D:\ORACLE
\T99\SAPDATA3\T9
9_1\T99.DATA1
   2 - PSAPT99EX         ONLINE  FILE          2048   NO     D:\ORACLE
\T99\SAPDATA3\T9
9EX_1\T99EX.DATA1
   3 - PSAPT99USR        ONLINE  FILE         10240   NO     D:\ORACLE
\T99\SAPDATA3\T9
9USR_1\T99USR.DATA1
   4 - PSAPTEMP          ONLINE  FILE         20480   NO     D:\ORACLE
\T99\SAPDATA2\TE
MP_1\TEMP.DATA1
   5 - PSAPUNDO          ONLINE  FILE         20480   NO     D:\ORACLE
\T99\SAPDATA2\UN
DO_1\UNDO.DATA1
   6 - SYSAUX            ONLINE  FILE        204800   NO     D:\ORACLE
\T99\SAPDATA1\SY
SAUX_1\SYSAUX.DATA1
   7 - SYSTEM            SYSTEM  FILE        409600   NO     D:\ORACLE
```

```
\T99\SAPDATA1\SY
STEM_1\SYSTEM.DATA1

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
------------------------------------------------------------------
--------------
BR0662I Enter your selection:
```

e) Choose the data file of tablespace *PSAP<DBSID>EX*.

f) Choose *Continue*.

g) Enter 4 in the *New data file size in MB (size)* parameter. The database system check should not display any warning or error regarding free space in any tablespace.

h) Call the `tsfill.bat` command as follows:

```
D:\oracle\T99\scripts>tsfill.bat
Filling the tablespace PSAPT99EX

SQL*Plus: Release 11.2.0.1.0 Production on Mo Jun 6 16:33:22 2011

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

Connected.

1454 rows created.

Disconnected from Oracle Database 11g Enterprise Edition Release
11.2.0.1.0 - 64
bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options
```

When running the `tsfill.bat` script several times, the following message indicating a tablespace overflow appears:

```
D:\oracle\T99\scripts>tsfill.bat
...
...
...
D:\oracle\T99\scripts>tsfill.bat
Filling the tablespace PSAPT99EX

SQL*Plus: Release 11.2.0.1.0 Production on Mo Jun 6 16:33:36 2011

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

Connected.
INSERT INTO SKEL  SELECT * FROM DBA_INDEXES
*
ERROR at line 1:
ORA-01653: unable to extend table SAPT99.SKEL by 60 in
tablespace
PSAPT99EX


Disconnected from Oracle Database 11g Enterprise Edition Release
11.2.0.1.0 - 64
bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

D:\oracle\T99\scripts>tsfill.bat
```

2. Set the data file of tablespace *PSAP<DBSID>EX* to autoextensible. The maximum size is 10 MB; the increment size is 2 MB. Check the size of the data file and run the `tsfill.bat` script. Check the size of the data file again; it should have grown by 2 MB.

   a) Start BRGUI or BRTOOLS and choose *Space management → Alter data file*.

   b) Choose *Continue*.

   c) Choose *Turn on and maintain autoextend*.

   d) Choose *Continue* until the following list of data files displays:

```
BR0657I Input menu 316 - please enter/check input values
-------------------------------------------------------------------
--------------
Options for alter of data file D:\ORACLE
\T99\SAPDATA3\T99EX_1\T99EX.DATA1

 1 * Current data file status (status) ....... [FIXSIZE]
 2 * Current data file size in MB (currsize) . [4]
 3 * Alter data file action (action) ......... [autoext]
 4 - Maximum file size in MB (maxsize) ....... [10]
 5 - File increment size in MB (incrsize) .... [2]
 6 # New data file size in MB (size) ......... [4]
 7 # New data file name (name) ............... []
 8 # Force data file alter (force) ........... [no]
 9 - SQL command (command) ................... [alter database
datafile 'D:\ORAC
LE\T99\SAPDATA3\T99EX_1\T99EX.DATA1' autoextend on next 2M
maxsize 10M]

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-------------------------------------------------------------------
--------------
BR0662I Enter your choice:
```

   e) Choose the *PSAP<DBSID>EX* data file.

   f) Choose *Continue*.

   g) In the next menu, enter the following data:

Parameter	Value
Maximum file size in MB (size)	**10**
File increment size in MB (size)	**2**

   h) Check the size of the data file before and after running `tsfill.bat`. The following message displays the size of data file before running the `tsfill.bat` script:

```
D:\oracle\T99\scripts>dir D:\oracle
\T99\sapdata3\T99EX_1\T99EX.DATA1
 Volume in drive D is Database
 Volume Serial Number is 1CD3-C56E

 Directory of D:\oracle\T99\sapdata3\T99EX_1

06.06.2011  16:32          4.202.496 T99EX.DATA1
               1 File(s)      4.202.496 bytes
               0 Dir(s)  573.732.069.376 bytes free

D:\oracle\T99\scripts>
```

Run the script few times.

```
D:\oracle\T99\scripts>tsfill.bat
Filling the tablespace PSAPT99EX

SQL*Plus: Release 11.2.0.1.0 Production on Mo Jun 6 16:44:04 2011

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

Connected.

1454 rows created.

Disconnected from Oracle Database 11g Enterprise Edition Release
11.2.0.1.0 - 64
bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

D:\oracle\T99\scripts>
```

The following message displays the size of the data file after running the script:

```
D:\oracle\T99\scripts>dir D:\oracle
\T99\sapdata3\T99EX_1\T99EX.DATA1
 Volume in drive D is Database
 Volume Serial Number is 1CD3-C56E

 Directory of D:\oracle\T99\sapdata3\T99EX_1

06.06.2011  16:44         6.299.648 T99EX.DATA1
               1 File(s)      6.299.648 bytes
               0 Dir(s)  573.726.687.232 bytes free

D:\oracle\T99\scripts>
```

3. Create a new locally-managed tablespace, PSAP<DBSID>NEW, with one autoextensible data file with a size of 4 MB, increment size 2 MB, and maximum size of 20 MB.
   a) Start BRGUI or BRTOOLS and choose *Space management → Create tablespace*.
   b) In the *Main options for creation of tablespace in database <DBSID>* menu, enter PSAP<DBSID>NEW in the *Tablespace name (tablespace)* parameter.

   The following list of options displays:

```
BR0657I Input menu 305 - please enter/check input values
-----------------------------------------------------------------
--------------
Main options for creation of tablespace in database T99

 1 - Tablespace name (tablespace) ......... [PSAPT99NEW]
 2 - Tablespace contents (contents) ....... [data]
 3 - Segment space management (space) ..... [auto]
 4 # Database owner of tablespace (owner) . []
 5 # Table data class / tabart (class) .... []
 6 - Data type in tablespace (data) ....... [both]
 7 # Joined index/table tablespace (join) . []
 8 ~ Uniform size in MB (uniform) ......... []
 9 - Tablespace compression (compress) .... [no]
10 - Tablespace encryption (encrypt) ...... [no]
11 # Encryption algorithm (algorithm) ..... []

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
```

```
-------------------------------------------------------------------
--------------
BR0662I Enter your choice:
```

c) Choose *Continue* and define the file parameters, as shown below.

```
BR0657I Input menu 306 - please enter/check input values
-------------------------------------------------------------------
--------------
Space options for creation of tablespace PSAPT99NEW (1. file)

 1 - Tablespace file name (file) .......... [D:\oracle
\T99\sapdata3\t99new_1\t99
new.data1]
 2 # Raw disk / link target (rawlink) ..... []
 3 - File size in MB (size) .............. [4]
 4 - File autoextend mode (autoextend) .... [yes]
 5 - Maximum file size in MB (maxsize) .... [20]
 6 - File increment size in MB (incrsize) . [2]
 7 - SQL command (command) ................ [create tablespace
PSAPT99NEW extent
 management local autoallocate segment space management auto
datafile 'D:\oracle
\T99\sapdata3\t99new_1\t99new.data1' size 4M autoextend on next
2M maxsize 20M]

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-------------------------------------------------------------------
--------------
BR0662I Enter your choice:
```

d) Choose *Continue*.

After specifying the file parameters for the first the file, you can add an additional data file. To skip this action, choose *no* (BRTOOLS) or *skip* (BRGUI).

A new tablespace *PSAP<DBSID>NEW* with one autoextensible data file has been added.

LESSON SUMMARY

You should now be able to:

- Administer tablespaces

Unit 4
Lesson 2

Performing Reorganization of Tables

LESSON OVERVIEW

This lesson discusses how to perform an online reorganization of tables, indexes, and tablespaces.

Business Example

Your SAP system was installed years ago so you still have dictionary-managed tablespaces. Now, you have learned the advantages of locally-managed tablespaces and want to transform your dictionary-managed tablespaces into locally-managed tablespaces. For this reason, you require the following knowledge:

- An understanding of the situations in which a reorganization is necessary
- An understanding of how to perform a reorganization
- An understanding of how to transform dictionary-managed tablespaces into locally-managed tablespaces
- An understanding of how space can be reduced with the use of the advanced compression option

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Reorganize tables

Reorganization of Tables

Reorganization is the reconstruction, resize, or movement of segments such as tables, indexes, data files, or tablespaces in the database. It is generally performed to defragment database objects. By doing so, you can improve performance and regain space within tablespaces that were unused due to fragmentation.

A distinction is made between offline and online reorganization as follows:

- Offline reorganization

  It is always not possible to access the objects during an offline reorganization. In general, it is necessary to close the SAP system during the reorganization.

- Online reorganization

  You can always access the affected segments during the reorganization. In general, the SAP system can be run in parallel to the reorganization.

Classic method of carrying out an offline reorganization:

- Export the tables being reorganized (single tables or all tables of a tablespace).
- Drop the exported tables (including the corresponding tablespace if all tables were exported).
- Recreate the tablespace, if it was dropped before.
- Import the tables that were exported before.

This procedure (and other improved, but currently restricted procedures) created objects in the tablespaces contiguously and continuously. This improves performance and removes fragmentation.

Conversely, reorganization must be performed offline and takes a considerable amount of time when large tables or whole tablespaces must be reorganized.

Reasons for Reorganization (1)

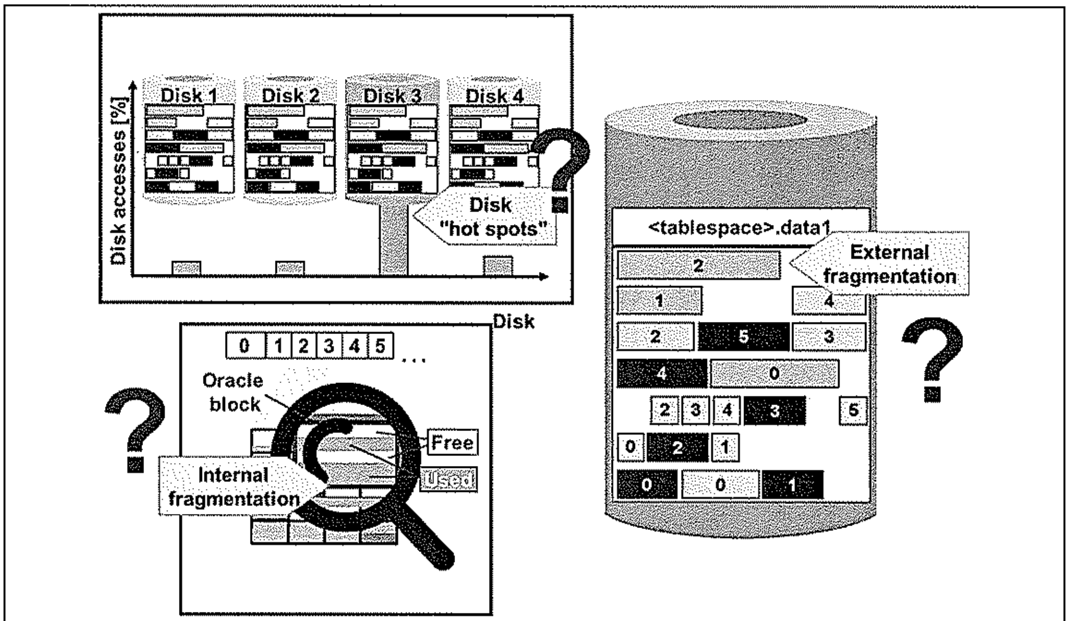


Figure 117: Reasons for Reorganization

Reorganizations can be useful in the following situations:

Situation	Uses
Tables	• Release free space permanently • Change properties like storage parameters, compression, and so on • Move to another tablespace • Eliminate corruptions • Sort a table

Situation	Uses
Tablespaces	• Prevent tablespace fragmentation • Change properties like locally-managed tablespace or automatic segment space management (ASSM)
Data files	• Unequal IO distribution • Data files with (by accident) to large free space
Indexes	• Index fragmentation • Change properties like storage parameters, compression, and so on • Move to another tablespace
Abstract	• Space reduction • Performance increase • Easier administration • Move certain large and heavily used tables into separate tablespaces.

Reasons for Reorganization (2)

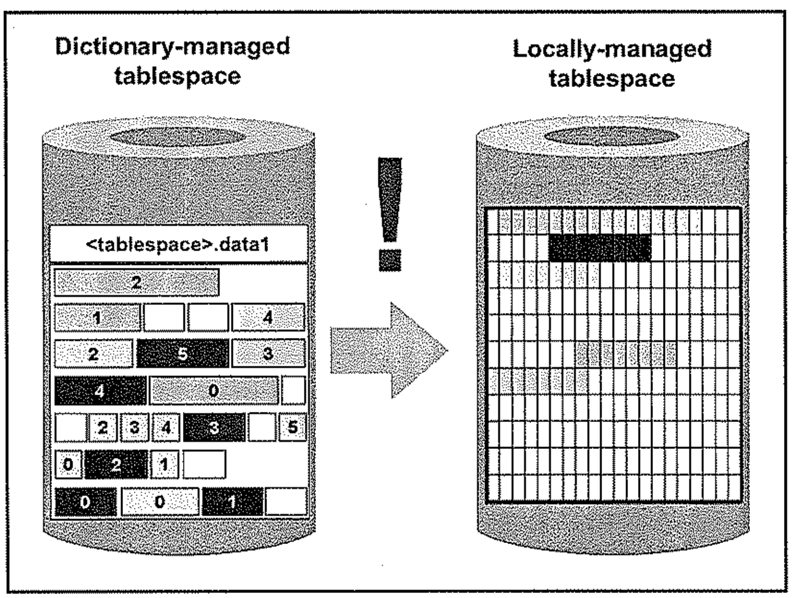


Figure 118: Reasons for Reorganization

Starting with Oracle 8 and continuing with Oracle 9i, and also because of evolving disk and redundant array of independent disks (RAID) technology, new features have been introduced to make a reorganization less necessary.

These new Oracle features are as follows:

- Efficiency within a tablespace has increased using locally-managed tablespaces and space allocation. There is no `MAXEXTENTS` parameter anymore to stop Oracle from allocating new extents. There is also no `NEXT` parameter, which can cause Oracle to create too many or too large extents if set incorrectly.
- There is a significant reduction in internal fragmentation within Oracle blocks and an improvement in the performance of parallel queries using automatic segment space allocation.
- There is a significant reduction in I/O hotspots because of large disks and RAID systems with large and secure memory buffers. Spreading database I/Os to different controllers and disks is performed by a good setup of the RAID system instead of manually distributing data files on different disks.

Methods to Perform a Reorganization

Oracle offers the following methods of performing a reorganization:

Method	Uses
Offline reorganization (Export or Import)	• Temporarily exports the data and metadata to OS level, drops and recreates the table, and loads the data • Creates the indexes
Online redefinition	• Creates a target table • Copies data from source to target table while the source table is still in use (online) • Keeps track of changes in the source table and reapplies those changes at the switch to the target table
Alter table ... move	• Creates a table with the same structure as the source table and copies the data in one command • Locks the table during data copy
Segment shrinking	• Fills the gaps in the first extents with the data of the last extents and return freed extents to the free space

All these methods to perform a reorganization are supported by BR*Tools (BRSPACE).

Comparison of Reorganization Methods

	Online Redefinition	Alter table ... move	Export/Import	Segment Shrinking
Typical use	Standard case	Reorganization of a mass of small tables	Oracle 9: Tables with long data type Tables that should be sorted	Simple reorganization
Online	Yes	No	No	Yes
Indexes	Recreated	Invalidated	Recreated	Parallel maintained
Speed	Fast	Fast	Slow (double move of data)	Slow (Single Row movement)
Additional Space	Target table and it's indexes	Target table and it's indexes	Target table and its indexes Export Dump on OS level	No
Complexity of steps	High	Medium	High	Low
In case of failure	Easy cleanup and Restart from beginning	Start from beginning	Repair and start from beginning Restore of Backup	Restart where terminated
Restrictions (extra)	Oracle 9: No LONG data type support Oracle 10: LONG -> LOB conversion Oracle 11: LONG -> SecureFile conversion	No LONG datatype support No structure change of table possible		No LONG datatype support Only ASSM Only to shrink space (no new segment are created)
BR*Tools support	Yes	Yes	Yes	Yes
Availability	>= Oracle 9	All releases	All releases	>= Oracle 10
Further information	SAP Online Documentation Reorganization with the Redefinition Package		SAP Online Documentation Export/Import	Note 910389
DB reorg with BR*Tools	brspace –f tbreorg –t <table>	brspace –f tbreorg –m offline –t <table>	brspace –f tbexport –t <table> brspace –f tbimport	brspace –f tbalter –a shrink –t <table>

Figure 119: Comparison of Reorganization Methods

The figure gives a comparison of reorganization methods.

Online redefinition is the standard method of reorganizing tables. Using Oracle online redefinition facilities, reorganization (or better online table redefinition) has become easier.

The advantages and features of online table redefinition are as follows:

- Can be performed online
- Can be parallelized for faster reorganization or not parallelized to prevent disturbing productive use of the database
- Can be used to move tables to a different tablespace in the same schema
- Can be used to recreate a table to reduce fragmentation
- Introduces less risk as checks are performed before the redefinition and objects being reorganized are only deleted after the new reorganized version of the object has been successfully created

Reorganization with BR*Tools

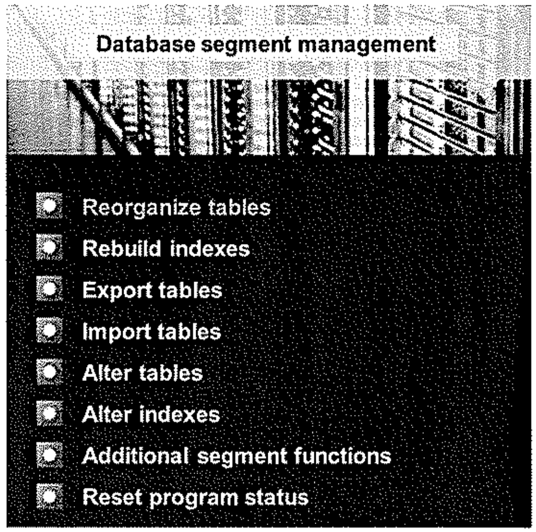


Figure 120: Reorganization with BR*Tools

To perform a reorganization, start BRTOOLS or BRGUI and choose *Segment management*. The list in the figure shows a short introduction of the different reorganization methods offered by the *Segment management* menu and for the kind of reorganization that they are used.

The reorganization methods offered by the Segment management menu are as follows:

- *Reorganize tables*

  Using *Reorganize tables*, BRSPACE performs an online redefinition of tables. Use this function to:

  - Reorganize tables due to internal or external fragmentation.
  - Transform dictionary-managed tablespaces to locally-managed tablespaces.
  - Transform tablespaces of traditional layout (different tablespaces for data and indexes) into tablespaces of Multiple Components in One Database (MCOD) layout and vice versa.
  - Move large tables to a separate tablespace.

- *Rebuild indexes*

  Use this function to rebuild fragmented or degenerated indexes. This normally improves the performance of the table's data access through indexes.

- *Export tables* and *Import tables*

  Use these functions to reorganize tables using the old reorganization method by exporting tables, dropping them, and importing them again.

  Normally, tables are reorganized using the BRSPACE function *Reorganize tables*, however, online redefinition is not possible in Oracle 9i for tables containing LONG fields (existing, for example, in SAP table clusters).

Hint:
Oracle has announced the discontinuation of support for LONG fields as of Oracle 11. As of Oracle 10g, online redefinition is possible for tables containing LONG fields; LONG fields are redefined as Large Object (LOB) fields. Reorganization tables containing LONG fields in Oracle 9i must be performed by export or import. SAP supports accessing LOB fields through LONG interface as of Oracle 10g. This makes it possible to redefine tables containing LONG fields into tables with LOB fields using BRSPACE.

SAP plans to support conversion from LONG to LOB fields. Then conversion can be done online with BRSPACE (Oracle 10g). After the conversion, all SAP tables can be reorganized online.

- *Alter tables* and *Alter indexes*

  The functions for altering tables or indexes do not actually reorganize a table or index, but turn certain features affecting performance on or off.

  In the *Switch on table monitoring* area, with table monitoring switched on, Oracle automatically collects statistics for these tables. This feature significantly improves the creation of statistics. When checking and updating the statistics, BRCONNECT first determines whether the number of rows in a table has changed since the last creation of statistics. For tables with monitoring switched off, the change in the number of rows in tables must be determined through collecting statistics on unique indexes. For tables with monitoring switched on, BRCONNECT selects information about table changes from *DBA_TAB_MODIFICATIONS* without updating index statistics.

Hint:
As of Oracle 9i, switch on table monitoring is available for all SAP tables. When turning on switch table monitoring, you can schedule a daily statistics update. As a result, the statistics are more up-to-date because the runtime takes only a fraction of the update time previously required.

Set parallel degree is possible for tables and indexes and can improve performance on INSERT statements.

Caution:
Only set parallel degree if you are told to do so by SAP support. On productive systems, no parallelization is used.

Reorganization of Tables

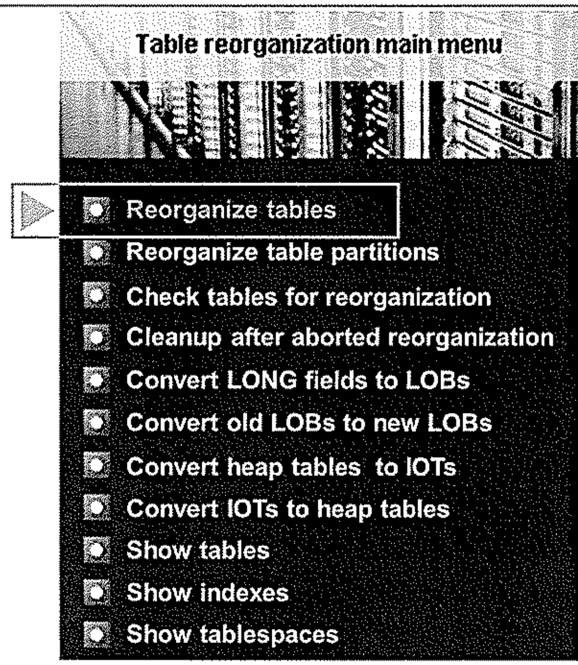


Figure 121: Reorganization of Tables Using BR*Tools

Before you start a reorganization using online table redefinition, check the following requirements:

- If reorganization is used to move tables into a new tablespace, this tablespace must already exist.
- If BRSPACE supports online table definition only in locally-managed tablespaces.
- When moving many tables or large tables, the data files of the target tablespace have enough free space or they should be automatically extensible and enough disk space should be provided to avoid a tablespace overflow.

To perform a reorganization using online table redefinition, start BRTOOLS or BRGUI and choose *Segment management → Reorganize tables*.

When you start BRTOOLS or BGUI, you can specify a list of tables or table names using wildcards and will go into quick mode of BRSPACE. Reorganization options are entered here. If you continue without entering any values (except the profile for BRSPACE), choose *Reorganize tables* menu from BRSPACE and then select tables from the list that displays.

> Hint:
> You can also specify a tablespace name instead of table names for reorganization. All tables from this tablespace (except tables with LONG fields) are assumed for reorganization.
>
> To reorganize a set of tables that cannot be matched by wildcards ([<owner.]<prefix>* or *), you can use parameter reorg_table in init<DBSID>.sap to specify a list of tables to be reorganized.

After selecting the tables to be reorganized, the following menu displays:

```
BR0657I Input menu 353 - please enter/check input values
-----------------------------------------------------------------------
--------
Options for reorganization of tables: SAPT99.SKEL (degree 1)

 1 * Reorganization action (action) ........... [reorg]
 2 - Reorganization mode (mode) ............... [online]
 3 - Create DDL statements (ddl) .............. [yes]
 4 ~ New destination tablespace (newts) ........ [PSAPT99NEW]
 5 ~ Separate index tablespace (indts) ........ []
 6 - Parallel threads (parallel) .............. [1]
 7 ~ Table/index parallel degree (degree) ...... []
 8 ~ Category of initial extent size (initial) . []
 9 ~ Sort by fields of index (sortind) ........ []
10 # Index for IOT conversion (iotind) ........ [FIRST]
11 - Compression action (compress) ............ [none]
12 # LOB compression degree (lobcompr) ........ [medium]
13 # Index compression method (indcompr) ....... [ora_proc]

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-----------------------------------------------------------------------
--------
BR0662I Enter your choice:
```

The parameters of the menu options are as follows:

- action

  The action parameter defines the reorganize table action. The reorg option represents the standard transaction of the online reorganization of tables.

- mode

  By default, the Oracle DBMS_REDEFINITION package is used for the reorganization MODE [online]. Instead of the Oracle DBMS_REDEFINITION package, it is possible to use the statement ALTER TABLE [PARTITION] MOVE [offline]. No Data Definition Language (DDL) statements are generated. Using the statement ALTER TABLE [PARTITION] MOVE [offline]can be quicker than an online reorganization, but locks the affected tables. The SAP system should, therefore, not be operational during the process.

  The ALTER TABLE [PARTITION] MOVE command does not support tables with LONG or LONG RAW fields; therefore, you cannot reorganize these tables offline.

- ddl

  To create the intermediate tables, BRSPACE creates an SQL script containing the DDL statements in $SAPDATA_HOME/sapreorg/<encoded_timestamp> called ddl.sql. This parameter influences the creation of the DDL script as follows:

  - If ddl.sql is set to NO, the DDL statements are created only internally and are not visible, which is not recommended.
  - The default setting is YES, which means that ddl.sql is created and, during reorganization, automatically used for the creation of intermediate tables.
  - FIRST means that ddl.sql is created before reorganization and BRSPACE stops before executing ddl.sql. This process enables you to view the statements and adapt them, if necessary. Choose *Continue* to perform the reorganization.

> Caution:
> Only an Oracle expert should perform changing the DDL statements before execution.

- `ONLY` will actually not perform a reorganization, but only create the DDL statements. In this case, #$ is not appended to the object names.

- `newts`

  Enter the tablespace into which the tables are to be reorganized. This tablespace must exist and be a locally-managed tablespace.

  If you do not specify a tablespace, the tables are reorganized within their original tablespace

- `indts`

  You only set this parameter if the tables are reorganized into another tablespace that only contains table segments. Specify the corresponding index tablespace here.

- `parallel`

  This parameter can be used to parallelize reorganization. A high parallelism speeds up reorganization, but has a greater impact on performance. With low parallelism, reorganization takes longer but does not affect performance as much.

  As good practice, `parallel` should be set to:

  - One (1) when reorganization is performed while SAP is active because reorganization has an impact on performance and must be performed in off-peak hours.

  - A value of one to two times the number of CPUs when reorganization is performed while SAP is down.

- `degree`

  This parameter is used to shorten a reorganization of large tables and parallelize the copy of data from the original table into the interim table. Parallel query is used by the *DBMS_REDEFINITION* package during the copy.

- `initial`

  This parameter provides the category of the INITIAL extend size.

- `sortind`

  This parameter is used to sort according to index fields. This sorting improves partially sequential accesses.

- `iotind`

  The conversion of normal (heap) tables (see SAP Note 641435) to index-organized tables (IOTs) and also the reverse action are supported by BRSPACE. This action is performed online using the DBMS_REDEFINITIONS package.

  A unique index on which the new index-organized table (IOT) is based must exist and be specified in the `iotind` option. (See SAP Note 1325304 – Conversions between heap organized tables and IOTs)

- `compress`

This parameter specifies the compression action (this can be a compression action or a decompression action).

- `lobcompr`

  This parameter defines the compression degree of SecureFile LOBs.

  You will find more details about LOB conversion and table compression in the SAP Note 1431296 - LOB conversion and table compression with BRSPACE 7.20.

- `indcompr`

  This parameter defines the index compression method. This is an index evaluation procedure that decides whether an index should be compressed and, if so, with how many key fields.

  See SAP Note 1464156 – Support for index compression in BRSPACE 7.20 for more details about index compression.

General Steps for Online Table Redefinition

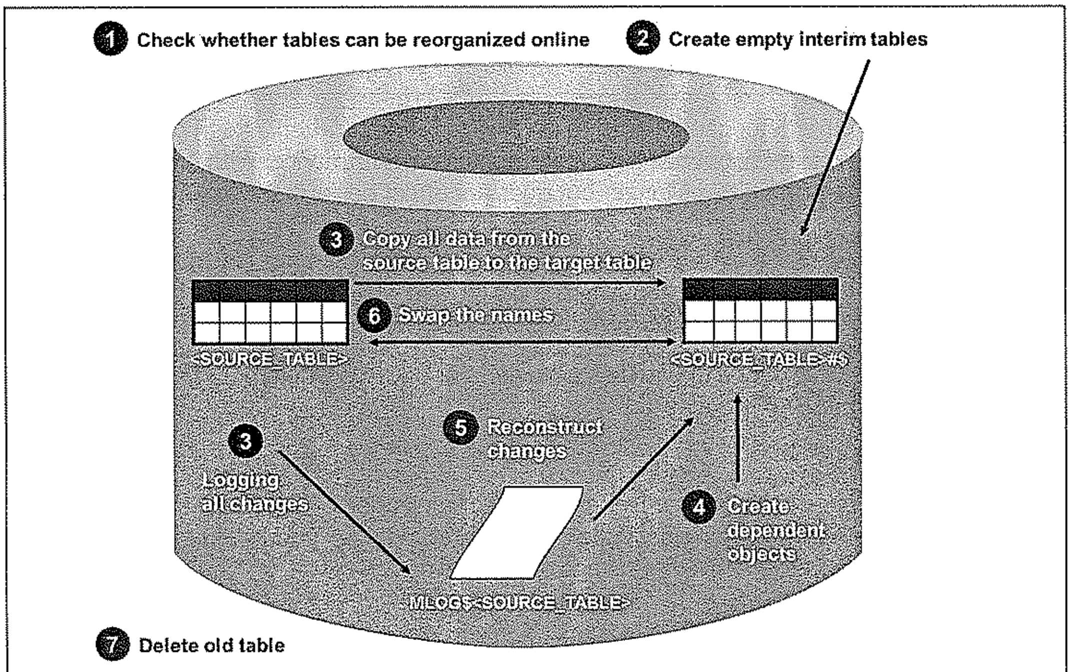


Figure 122: Online Redefinition Process

When performing an online table redefinition, BRSPACE uses the PL SQL package DBMS_REDEFINITION, provided by Oracle. The figure shows the steps in the redefinition process.

Steps for redefinition are as follows:

1. The system checks the tables see whether they can be redefined online. If the Oracle package returns an error on this check for one or more tables, BRSPACE excludes them from online redefinition (for example, tables having LONG fields).

2. The system creates an empty interim table. To create the DDL statement for this step, BRSPACE uses the PL/SQL package DBMS_METADATA. BRSPACE names the interim table *<SOURCE_TABLE>#$*.

3. The systems the redefinition process, which copies data from the original table into the interim table. You can run this as a parallel process with the `-e | -degree` option.

   Before calling `DBMS_REDEFINITION.START_REDEF_TABLE`, check the following:

   - Wait until all open changes have been committed to the table to be reorganized.
   - Copy all the data from the source table to the target table.
   - Log all changes to the source table in a Materialized View Log (MLOG $_<SOURCE_TABLE>).

4. The user creates all indexes, constraints, triggers, grants, and comments in the interim table. Referential constraints are disabled. BRSPACE names the triggers, indexes, and constraints *<ORIGINAL_NAME>#$*.
5. The system completes the redefinition using a procedure from the DBMS_REDEFINITION package. When this process ends, the original table is redefined and refers to all attributes, indexes, constraints, grants, and triggers in the interim table. Indexes, triggers, grants, constraints, and comments from the original table are transferred to the interim table. Referential constraints are enabled.
6. BRSPACE now renames the objects created in step 4 to their original name.
7. BRSPACE drops the interim table, causing indexes, constraints, triggers, grants, and comments defined on the original table to be dropped as well.

If any of the steps fail for a table, the original table is still there. BRSPACE then drops the interim table, causing any indexes, constraints, triggers, grants, and comments on the interim table to be dropped as well.

Transformation of Dictionary-Managed Tablespaces to Locally-Managed Tablespaces

To transform dictionary-managed into locally-managed tablespaces with Oracle 9i, perform a normal online reorganization of all tables within a tablespace into a new tablespace.

Perform the normal online reorganization of the tables using the following settings:

1. Before the reorganization, create one new locally-managed tablespace storing tables and indexes (contents: `both`). This tablespace must be large enough to store tables and indexes of both the old table and the old index tablespace (assuming you have a non-MCOD tablespace layout before reorganization). It is recommended to set the data files of the new tablespace to autoextensible.

> Hint:
> To have separate table and index tablespaces after reorganization, create two tablespaces, one with contents table and one with contents index.

2. Start BRGUI or BRTOOLS and choose *Segment management → Reorganize tables*. Specify the name of the tablespaces to be reorganized and enter the * character in the table field. No other options need to be changed.

> Hint:
> If you reorganize a tablespace of table that has an associated index tablespace, only enter the name of the table tablespace. All indexes of the reorganized tables will be automatically reorganized.

3. Continue and confirm the list of tables being reorganized.
4. In the *Options for reorganization of tables* menu, enter the name of the new tablespace. If you want to have separate table and index tablespaces after reorganization, specify the new index tablespace created under step 1. Enter the number of parallel threads and the option for creating DDL statements(see above). Then, choose *Continue*.

The online reorganization starts.

> Hint:
> Tables and their indexes that cannot be reorganized online (like tables with LONG fields) stay in the old tablespaces.
>
> To move these tables into the new locally-managed tablespace, you have the following options:
>
> - You must use the export or import functionality of BRSPACE, preferably after online reorganization.
> - As soon as offline conversion of tables containing LONG fields into LOB fields is available in BRSPACE (as of Oracle 10g), use this functionality before reorganization. After this procedure, these tables can be reorganized online.

See SAP Note 646681 for details, especially to find out which option that is offered at the moment and for details of the export or import if necessary.

Rebuilding of Indexes

To rebuild one or more indexes online, start BRTOOLS or BRGUI and choose *Segment management → Rebuild indexes*.

The menus displayed are similar to those displayed in table reorganization with the following exceptions:

- There is no DDL menu item because index reorganization is performed with the Oracle command ALTER INDEX REBUILD ONLINE and no objects have to be manually created.
- If you rebuild the indexes using BRTOOLS, the menu provides you the option of entering a list of tables. If you enter tables instead of indexes here, all indexes of the specified tables are rebuilt.
- You can enter a new tablespace name to move indexes into another tablespace during the rebuild.

Large Object Conversion and Table Compression

Among other features, Oracle 11g contains the following new features in the data storage area:

- New LOB types, SecureFile LOBs with the following advantages:

- Performance improvements
- Data compression
- Data encryption

- New and improved method of table compression, which promises significant memory space savings.

The activation of these new features is performed by an online reorganization.

Large Objects and Secure Files

Table: Webpages

Name	Url	Source
Page1	www.sap.com/page1	<Link 1>
Page2	www.sap.com/page2	<head></head><body></body>
Page3	www.sap.com/page3	<Link 2>
Copy Of 3	www.sap.com/page4	<Link 2>
...	...	...

LOB-Segment for Webpages-Source:
SYS_LOB0000003619C00002$$

Link	Data
Link1	<HTML Code>
Link2	<HTML Code>
...	...

Figure 123: Large Objects and Secure Files

Tables can store space-consuming values in one column, for example, ABAP programs and HTML pages. Store space-consuming values separately.

The LOB data types enable the storage of large blocks of unstructured data up to four gigabytes in size in a table column. The data stored in a LOB column can be stored within the table or in an extra object outside of the table, such as a segment of type LOBSEGMENT. To access the data efficiently, a segment of type LOBINDEX is created.

If the option `enable storage in row` is used for a LOB column during table creation, and the LOB column value is less than approximately 4000 bytes, the LOB column value is stored within the table. Larger values are stored in the extra LOB segment. The option `enable storage in row` is the default used by SAP. If the option `disable storage in row` is used during table creation, the data is stored in the extra lob segment.

Before the availability of LOBs, the LONG and LONG RAW data types were used. On tables with LONG or LONG RAW data type, some kinds of database operations, such as online redefinition, are not possible.

As of Oracle 9i, the data type LOB is available. In SAP systems, the data types LONG and LONG RAW are still used. These columns are not automatically converted to LOBs.

As of Oracle 10g, LONG or LONG RAW tables can be converted online to LOB columns.

Oracle 11g introduces a new approach, known as Secure Files, for internally storing and maintaining data from the data type LOB (see also SAP Note 1426979 – Oracle 11g: Secure Files– The new way to store LOB data). With the introduction of the term Secure Files as a new way of storing LOB data, Oracle also introduces the term Basic Files, which is the synonym for the way Oracle stored LOB data before Secure Files was developed.

Storage of Large Objects

Secure Files functionality enables a new storage method, which offers advantages in performance and scalability. Secure Files also offers encryption for security reasons and uses compression for space savings.

When creating new tables with LOB columns, the behavior of the database is controlled by the new initialization parameter `db_securefile`.

The default value is `db_securefile` = '`permitted`'. With this setting, Oracle uses BASICFILE when creating a table with a LOB column without specifying any storage option (BASICFILE or SECUREFILE) . No other values are allowed for SAP.

Advanced Compression

Rapid increase in data volumes have put enterprise IT infrastructures under severe pressure from a cost, performance, scalability, and manageability perspective. Efficient ways of storing and managing data are needed to meet the growing demands of large systems. Database compression is an important technique of reducing storage needs.

The Oracle advanced compression option contains the following features:

- OLTP table compression

  OLTP table compression compresses data by eliminating duplicate values in a database block. Compressed data stored in a database block is self-contained. That is, all the information needed to recreate the uncompressed data in a block is available within that block. Duplicate values in all the rows and columns in a block are stored once at the beginning of the block, which is called a symbol table for that block. All occurrences of such values are replaced with a short reference to the symbol table.

- SecureFile compression

  SecureFile compression can be used to compress in-line and out-of-line LOB data of SAP tables. To use this feature, existing LOB data needs to be stored as SecureFiles in the database.

- RMAN compression

  With RMAN compression, backup data is compressed before it is written to disk or tape and does not need to be uncompressed before recovery – providing up to five times reduction in storage costs.

- Data guard network compression

  Data guard network compression introduces the capability to compress redo data before it is sent over the network to the standby database(s). Through this compression, network bandwidth is maximized to increase network throughput. Redo transport can be up to two times faster with compression.

> Note:
> See SAP Note 1436352 – Oracle 11g Advanced Compression for SAP Systems.

OLTP Table Compression

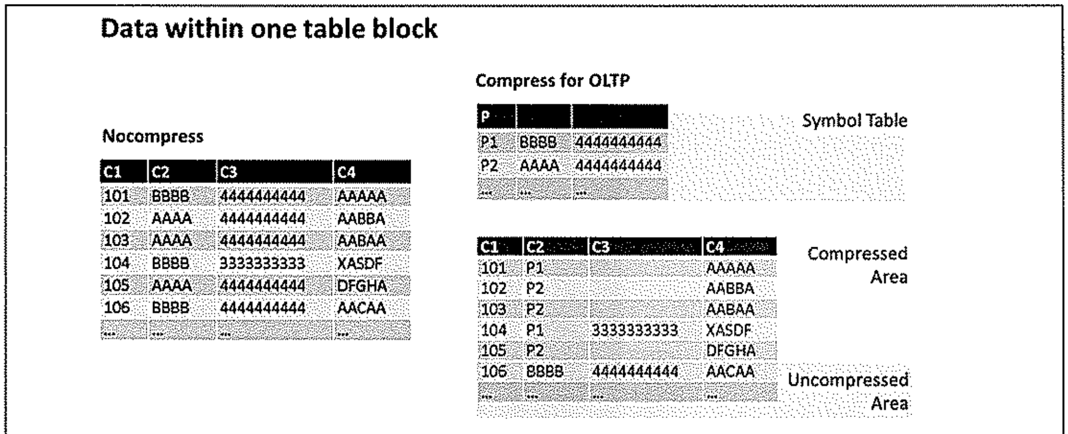


Figure 124: OLTP Table Compression

Table compression reduces the space occupied by a table. This can have a positive impact on performance, if the rows to be returned are in less blocks after compression than before.

The distinguishing characteristics of OLTP table compression are as follows:

- Works block wise
- Uses placeholders for redundant patterns in the block
- Stores references for the patterns in the rows to a symbol table that is also stored within the block
- Compresses only if specific thresholds are reached

  The block must be filled to find good patterns such as follows:

  - Some table blocks may be compressed.
  - Some rows within a block may be compressed.

> Hint:
> In heavily updated tables, issues with chained rows can occur. These can lead to a decrease in performance. Therefore, do not use extended table compression for such tables.

LOB Conversion and Table Compression Using BRSPACE

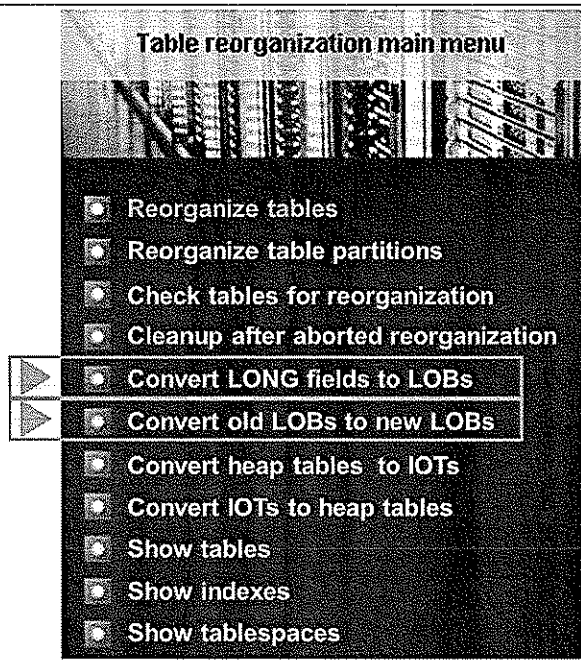


Figure 125: LOB Conversion Using BRSPACE

LOB conversion and table compression using BRSPACE provide new options for activating the new features, ACO and SecureFiles. The table reorganization menu offers the new options `lob2lob` (convert old LOBs to new LOBs) and `compress` (table compression).

LOB conversion and table compression of the entire database using BRSPACE 7.20 runs in a similar way to a migration to locally-managed tablespaces (both actions can be combined). Therefore, it is recommended that you move all the tables into newly created tablespaces that are locally-managed, and activate ASSM. When using BRSPACE to create the new tablespaces, you must ensure that the table data classes (tabart) are handled correctly. For more details, see SAP Note 646681. The newly created tablespaces do not have to activate the compression attribute at tablespace level.

The entire process runs online (online reorganization). That is, it can theoretically take place when the SAP system is running. However, because this requires a large amount of resources, the operation of the SAP system is restricted. Therefore, it is recommended that you perform this process when the system load is low, or when the SAP system is stopped (if possible).

Table Compression Using BRSPACE

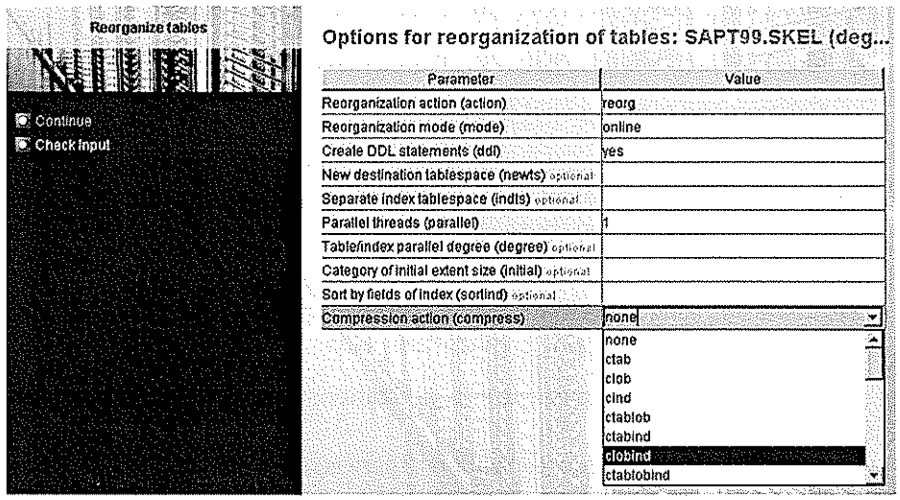


Figure 126: Table Compression Using BRSPACE

Details about LOB conversion and table compression are described in SAP Note 1431296 – LOB conversion and table compression with BRSPACE 7.20.

Segment Shrinking

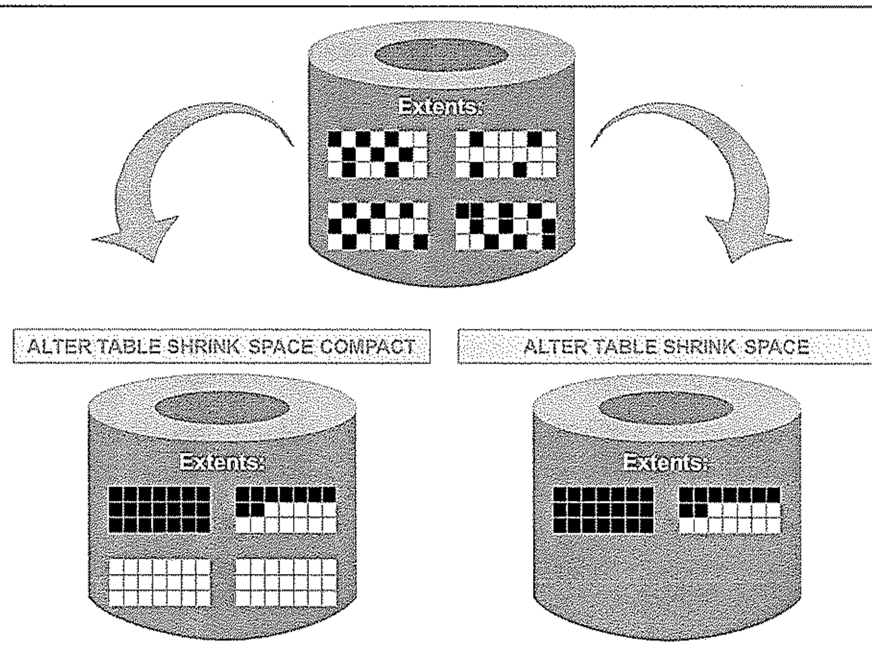


Figure 127: Segment Shrinking

Segment shrinking offers an alternative to reorganization for defragmenting a segment and gaining more free space. The free space occupied by a table by combining the segments is retrieved.

Segment shrinking is available as of Oracle Release 10g.

To use this new function, the following prerequisites must be met:

- The table to be shrunk must occur in an ASSM tablespace.
- Tables with LONG and LONG RAW fields cannot be shrunk.
- Mapping tables and overflow segments of IOTs cannot be shrunk.
- Compressed tables cannot be shrunk.
- ROW MOVEMENT must be activated for the table to allow ROWs to be shifted to a different location within a segment. If the ROWID is used when accessing the application, activating ROW MOVEMENT can cause issues. However, this is not the default setting in an SAP environment.

Segment shrinking gains free space by combining segments. It is not possible to store a table in another tablespace during segment shrinking.

Compared to offline and online reorganizations, segment shrinking offers the following advantages:

- Compared to offline reorganizations, there is no downtime.
- While an online reorganization temporarily requires twice the space, segment shrinking does not require any additional space.
- During an online reorganization, ALL table entries are accessed, while only a section of the entries are shifted during segment shrinking. This means that the amount of redo log information during segment shrinking is less than with an online reorganization.

Segment shrinking is particularly useful if a large amount of space is lost because of table fragmentation, causing performance issues. Segment shrinking allows you to gain space in the tablespace that can be reused for other segments, and to improve performance, depending on the situation.

When you shrink a segment, you can also specify the COMPACT and CASCADE options as follows:

- If you specify COMPACT, although the table is defragmented, the high water mark is not changed. This means that the excess space is not returned to the tablespace. The advantage, however, is that no locks are required.
- Without specifying CASCADE, the table is defragmented, the high water mark is adjusted and the unused space behind the high water mark is released to be reused by other segments in the tablespace. A temporary lock is required at the end of the runtime.
- By specifying CASCADE, not only the tables, but also dependent segments (such as indexes) are shrunk.

Caution:
The following issues can be triggered by segment shrinking:

- Fatal corruptions can occur in tables with LOB fields. With Oracle 10.2.0.2, make sure that the Oracle bugfix from SAP Note 1021454 has been implemented before you perform segment shrinking in tables with LOB columns.
- Even if segment shrinking is mainly an online operation, table locks (TM enqueues) may occur. The best solution is to run segment shrinking only at times when no long-running transactions are active, and change the underlying table.

Segment Shrinking with BR*Tools

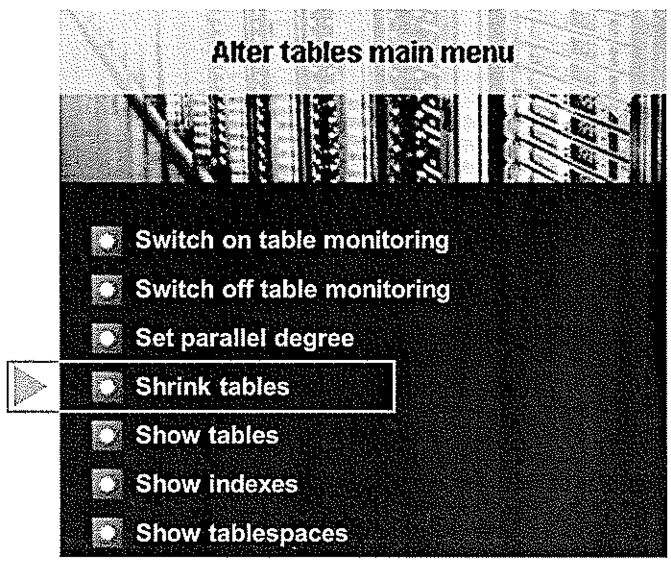


Figure 128: Segment Shrinking with BR*Tools

As of BR*Tools Release 7.00, segment shrinking can be performed using BRSPACE. The corresponding menu is shown in the figure.

Unit 4
Exercise 12

Work with Table Reorganization

Business Example

Now that you know the advantages of locally-managed tablespaces, you want to transform your dictionary-managed tablespaces to locally-managed tablespaces.

Convert dictionary-managed tablespaces to locally-managed tablespaces.

1. Reorganize all tables of tablespace *PSAP<DBSID>EX* into tablespace *PSAP<DBSID>NEW* (created in the last exercise).

Unit 4
Solution 12

Work with Table Reorganization

Business Example

Now that you know the advantages of locally-managed tablespaces, you want to transform your dictionary-managed tablespaces to locally-managed tablespaces.

Convert dictionary-managed tablespaces to locally-managed tablespaces.

1. Reorganize all tables of tablespace *PSAP<DBSID>EX* into tablespace *PSAP<DBSID>NEW* (created in the last exercise).
   a) Start BRGUI or BRTOOLS and choose *Segment management → Reorganize tables*.
   b) In *BRSPACE options for reorganization of tables*, enter the following data to reorganize all tables in this tablespace:

Parameter	Value
Tablespace names (tablespace)	**PSAP<DBSID>EX**
Table names (table)	*

   The following menu appears:

```
BR0657I Input menu 91 - please enter/check input values
-------------------------------------------------------------------
--------------
BRSPACE options for reorganization of tables

 1 - BRSPACE profile (profile) ...... [initT99.sap]
 2 - Database user/password (user) .. [/]
 3 ~ Reorganization action (action) . []
 4 ~ Tablespace names (tablespace) .. [PSAPT99EX]
 5 ~ Table owner (owner) ........... []
 6 ~ Table names (table) ........... []
 7 ~ Table partitions (tabpart) ..... []
 8 - Confirmation mode (confirm) .... [yes]
 9 - Extended output (output) ....... [no]
10 - Scrolling line count (scroll) .. [20]
11 - Message language (language) .... [E]
12 - BRSPACE command line (command) . [-p initT99.sap -s 20 -l E
-f tbreorg -s P
SAPT99EX]

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-------------------------------------------------------------------
--------------
BR0662I Enter your choice:
```

   c) Choose *Continue* two times.
   d) In *Table reorganization main menu*, choose *Reorganize tables*.

The following menu appears:

```
BR0656I Choice menu 351 - please make a selection
-------------------------------------------------------------------
--------------
Table reorganization main menu

 1 = Reorganize tables
 2 - Reorganize table partitions
 3 - Check tables for reorganization
 4 - Cleanup after aborted reorganization
 5 - Convert LONG fields to LOBs
 6 - Convert old LOBs to new LOBs
 7 - Convert heap tables to IOTs
 8 - Convert IOTs to heap tables
 9 - Show tables
10 - Show table partitions
11 - Show tablespaces
12 * Exit program
13 - Reset program status

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-------------------------------------------------------------------
--------------
BR0662I Enter your choice:
```

e) In the *Options for reorganization of tables: SAP<DBSID>.SKEL (degree 1)* menu, enter **PSAP<DBSID>NEW** in the *New destination tablespace (newts)* parameter.

The following menu appears:

```
BR0657I Input menu 353 - please enter/check input values
-------------------------------------------------------------------
--------------
Options for reorganization of tables: SAPT99.SKEL (degree 1)

 1 * Reorganization action (action) ........... [reorg]
 2 - Reorganization mode (mode) ............... [online]
 3 - Create DDL statements (ddl) .............. [yes]
 4 ~ New destination tablespace (newts) ........ [PSAPT99NEW]
 5 ~ Separate index tablespace (indts) ......... []
 6 - Parallel threads (parallel) .............. [1]
 7 ~ Table/index parallel degree (degree) ...... []
 8 ~ Category of initial extent size (initial) . []
 9 ~ Sort by fields of index (sortind) ......... []
10 # Index for IOT conversion (iotind) ......... [FIRST]
11 - Compression action (compress) ............ [none]
12 # LOB compression degree (lobcompr) ......... [medium]
13 # Index compression method (indcompr) ....... [ora_proc]

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
-------------------------------------------------------------------
--------------
BR0662I Enter your choice:
```

f) Choose *Continue* twice.

Then, online reorganization starts. The following message appears:

```
BR1101I Starting 'online' table reorganization...
BR0280I BRSPACE time stamp: 2011-06-07 10.56.38
BR1124I Starting 'online' reorganization of table SAPT99.SKEL ...
BR0280I BRSPACE time stamp: 2011-06-07 10.56.49
BR1105I Table SAPT99.SKEL reorganized successfully
```

```
BR0280I BRSPACE time stamp: 2011-06-07 10.56.49
BR1141I 1 of 1 table processed - 7288 of 7288 rows done
BR0204I Percentage done: 100.00%, estimated end time: 10:56
BR0001I **************************************************

BR0280I BRSPACE time stamp: 2011-06-07 10.56.49
BR1102I Number of tables reorganized successfully: 1

BR0280I BRSPACE time stamp: 2011-06-07 10.56.50
BR0256I Enter 'c[ont]' to continue, 's[top]' to cancel BRSPACE:
```


LESSON SUMMARY

You should now be able to:

- Reorganize tables

Unit 4
Lesson 3

Housekeeping and Troubleshooting

LESSON OVERVIEW

This lesson gives an overview of all regular activities that need to be performed on a database. The lesson explains recommended checks, how you can view the results, and how you should respond to warnings or errors. The lesson also discusses how to prevent and solve typical problems that might occur on the database.

Business Example

You have scheduled all recommended checks and backups. Nevertheless, an archiver stuck occurs. You need to resolve the archiver stuck as soon as possible, and you want to know how an archiver stuck can be prevented in the future. For this reason, you require the following knowledge:

- An understanding of the most common issues in Oracle databases
- An understanding of how to prevent an archiver stuck and solve it when it occurs
- An understanding of how to solve typical problems that occur on productive Oracle databases

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Perform housekeeping and troubleshooting

Housekeeping and Troubleshooting Concepts

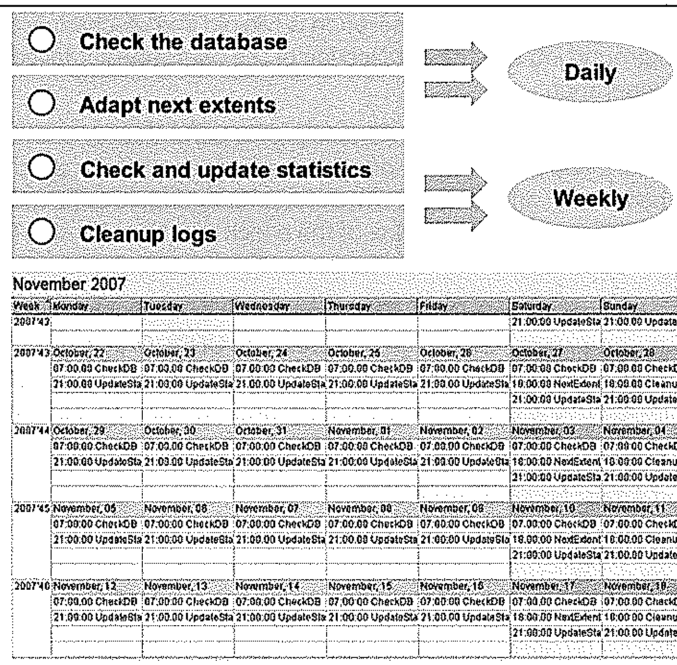


Figure 129: Regular Database Checks

In most cases, performing regular checks and viewing the results enables you to act before an error occurs, instead of waiting for the error to happen.

Database System Check

Tool

```
brconnect -u / -c -f check
```

Schedule

Schedule this job daily using the action *Check database* in the DBA Cockpit (transaction `DB13`).

Results

Check the action log daily with the DBA Cockpit (transaction `DB14`).

Check all yellow or red messages created after your last check of the results.

Actions

Resolve warnings and errors.

For open messages of type DBA (database administration), check the error or warning. There are two solutions to solve the problem. For most warnings or errors, you must check the reason for the message and then solve it. For some warnings, you may decide that there is no reason to react because you do not consider it as a warning. For example, the tablespace *PSAP<SCHEMA-ID>* has an overall size of 500 GB. During the database check, the system tells you that the tablespace is 95% full, and there is 24 GB free capacity. You decide after checking the database growth with transaction `DB02` that you want to be warned only if a tablespace has less than 5 GB free space. In this case you would change the check condition in table *DBCHECKORA* using transaction `DB17`.

For open messages of type DBO (database operation), check the scheduling of the corresponding action if the last operation is too old and reschedule the action. Check the log of the background job and the detailed log of the operation if it failed, solve the problem, and repeat the operation.

For open messages of type PROF (Oracle profile parameters), adapt the parameter in the Oracle profile. If you are sure that despite the warning, the current parameter is correct (for example, because of an SAP EarlyWatch recommendation, or because this value is listed in SAP Note 124361), change the value of the check condition in table *DBCHECKORA* using transaction `DB17`.

For open messages of type ORA, (Oracle error messages), check the Oracle documentation (command `oerr ora <message number>` on UNIX, Oracle error messages documentation on all operating systems), which provides the possible cause and solution of the error. If you are not sure about the cause of the problem and the solution, search the note in the SAP Service Marketplace for the error message.

Check and Adapt NEXT Extents

Tool

```
brconnect -u / -c -f next
```

Schedule

Schedule this job weekly using the action *Adapt next events* in the DBA Cockpit (transaction `DB13`).

Results

Check the action log weekly with the DBA Cockpit (transaction `DB14`). Check all yellow or red messages created after your last check of the results.

Actions

In rare cases, BRCONNECT adapts the `NEXT` extent size of certain tables that are too large, or you can set the `NEXT` extent size to a value defined by yourself. In this case, adapt the profile `init<DBSID>.sap` and set the parameter `next_exclude` to exclude tables from adapting the `NEXT` extent size. You could also set the parameter `next_special` to set a self-defined `NEXT` extent size.

> Hint:
> Do not schedule this action if all tablespaces in the database are locally managed.

Update optimizer statistics

Tool

```
brconnect -u / -c -f stats -t all
```

Schedule

Schedule this job daily with Oracle 10g using the action *Check and update optimizer statistics* in the DBA Cockpit (transaction `DB13`).

> Hint:
> For all other Oracle releases up to 9i, schedule a weekly run.

Results

Check the action log daily with the DBA Cockpit (transaction `DB14`). Check all yellow or red messages created after your last check of the results.

Actions

Sometimes, developers create new tables, which do not have statistics until the next planned run of `brconnect -f stats`. To create statistics of only tables that do not have statistics, run `brconnect -f stats -t missing`, or run transaction `DB20`, and choose *Global statistics → Create missing*.

Clean Up Old Logs and Traces

Tool

```
brconnect -u / -c -f cleanup
```

Schedule

Schedule this job weekly using the action *Cleanup logs* in the DBA Cockpit (transaction `DB13`).

Results

Check the action log weekly with the DBA Cockpit (transaction `DB14`). Check all yellow or red messages created after your last check of the results.

Actions

If no general hints are provided when log files are deleted too early or too late, check the parameter `cleanup_*` in profile `init<DBSID>.sap`.

Database backup

Tool

BRBACKUP and BRARCHIVE

Schedule

Schedule this job daily in the DBA Cockpit (transaction `DB13`). In most cases, planning the action *Whole database online + redo log backup* is sufficient, as here BRBACKUP and BRARCHIVE are executed in one run.

When choosing any other backup strategy, ensure the following points:

- For any incremental or partial backup, you have four full backups within your backup cycle.
- For any whole backup without redo logs, incremental, or partial backup, you have scheduled a daily redo log backup.

Results

Check the action log daily with the DBA Cockpit (transaction `DB14`).

Check all yellow or red messages created after your last check of the results.

Actions

If no general hints are provided, then check the log files of BRBACKUP and BRARCHIVE.

Database Support Without ABAP Schema

As of SAP Web Application Server Java 6.40 SR 1, the tables SDBAH and SDBAD are created in the Java database schema (SAP<Schema-ID>DB) as the standard schema. This completes the requirements for using BR*Tools for pure Java databases (without ABAP schema), which means you can save and restore Java databases with the BR*Tools.

The lack of the DBSTATC and DBCHECKORA tables in this environment, however, means that there are no Customizing options for updating the database statistics or for the database check. This restriction has now been resolved in BR*Tools 7.00. You can also use these tools with Oracle 9.2. Oracle Instant Client 10g must be installed on the database server (for more information, see SAP Note 849483).

In BR*Tools 7.00, two parameters are introduced in `init<DBSID>.sap`. These parameters assume the function of the tables DBSTATC and DBCHECKORA (see SAP Note 892294).

The parameters introduced in init<DBSID>.sap are as follows:

- `stats_special`

  For tables with specific statistics handling

- `check_cond`

  For Customizing the check conditions

> Hint:
> These new parameters and their options are described in detail in SAP Note 892294. This note also contains a copy template for the parameter `check_cond` according to the check conditions from the DBSTATC table.

Typical Problems

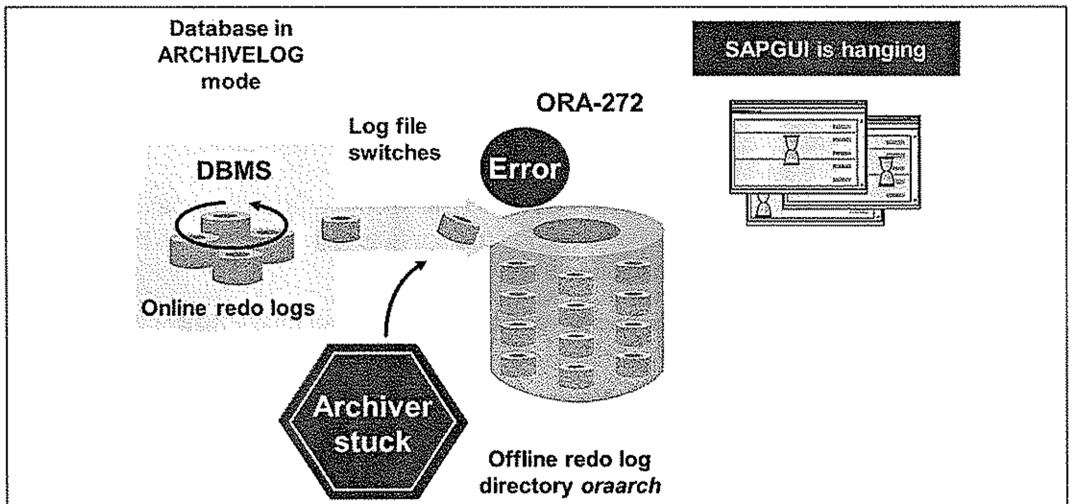


Figure 130: Archiver Stuck

Despite monitoring the database regularly and performing all the necessary administrative activities, sometimes issues occur.

One of the issues that you can face is archiver stuck. An archiver stuck is a situation in which the archiver instance process (ARCO) cannot copy an online redo log file to the archive directory (usually, `$SAPDATA_HOME/oraarch`) and the online redo log file to be copied needs to be overwritten by the log writer (LGWR).

When an archiver stuck occurs, the database instance is not shut down; however, the database instance does not allow any changes. In the SAP system, users see an hourglass as long as the archiver is stuck and Oracle writes an error message into its alert log file and create a trace file in `$ORACLE_HOME/ saptrace/background`. The reason for the archiver stuck error is that the disk holding the archiving directory is full.

The archiving directory can be filled because of the following reasons:

- There was high database activity (for example, to SAP data archiving); therefore, an exceptionally high number of archived redo logs was written.
- One or more runs of BRARCHIVE failed, for example, because there is a problem with a tape station that cannot be repaired in a short-time period. As BRARCHIVE normally deletes archived redo logs that have been saved successfully, the archive directory fills up more quickly than usual.

Reducing Occurrences of Archiver Stuck

You can never be certain that you never get an archiver stuck because you never have unlimited disk space. Therefore, you must be prepared for the archiver stuck error and know how to resolve it. There are some prerequisites that help an archiver stuck occur rarely or help solve an archiver stuck easily.

Prerequisites for reducing archiver stuck occurrence are as follows:

- Ensure that the archive directory has as much disk space as possible. The minimum recommended space is three times the space you need for the archived redo logs each day, because the default action of BRARCHIVE is to back up the archived redo logs twice before deleting them. You need some extra space to store the archived redo logs when you have higher database activity. To be able to perform a recovery over a long period of time, a minimum of 100 offline redo log files must fit into the archive directory.

> Hint:
> The larger the disk space in the archive directory, the smaller the chance of experiencing an archiver stuck.

- Create a dummy file of about 10 times the size of an online redo log file in the archive Directory, with the name `delete_me_on_archiver_stuck`. An archiver stuck cannot be avoided completely; when it occurs, you simply delete the dummy file, giving you some time to run BRARCHIVE to save and delete old archived redo logs.

> Caution:
> Never delete archived redo logs without having them backed up; otherwise, a full recovery from any backup created up to now is not possible.

- Do not keep the archive log (defined by the Oracle parameter `log_archive_dest`) on the same hard disk as the directory in which BRARCHIVE writes its log files (`$SAPARCH` or, if not set, $SAPDATA_HOME/saparch). If the archive log is full, BRARCHIVE may fail to save and may delete existing archived redo log files because BRARCHIVE cannot write to its own log file.

> Hint:
> Up to SAP R/3 4.6C, the SAP installation tools chose $SAPDATA_HOME/saparch to store archived log files and the BRARCHIVE log files. You should manually change it to oraarch in this situation.

To store archived log files on a disk different from the BRARCHIVE log files, perform the following steps:

1. Check the current log_archive_dest parameter and make a note of the complete path. To do so, start BRTOOLS or BRGUI and choose *Instance Management → Show database parameters*.
2. Run brarchive -cds two times to save and delete all existing archived redo log files.
3. Create a new directory, orarch, on a different disk, as follows:
   - In Windows, on the other disk, the complete directory tree <new disk>:\oracle\<DBSID>\oraarch must be created.
   - In UNIX, create the directory /oracle/.<DBSID>/oraarch, unmount /oracle/<DBSID>/saparch, change the mount definition in the file systems table (/etc/fstab, the file system, which was mounted to /oracle/<DBSID>/saparch should now be mounted to /oracle/<DBSID>/oraarch) and mount orarch with the command mount /oracle/<DBSID>/oraarch. Now copy the old log files back to saparch: cp /oracle/<DBSID>/oraarch/* /oracle/<SID>/saparch.
4. Change the parameter log_archive_dest . Only change the path; the last part of the original parameter log_archive_dest is not a directory, but the prefix for the names of the archived redo logs. Run the command brspace -c force -f dbparam -p log_archive_dest -v <new drive>:\oracle\<SID>\oraarch\<SID>arch on Windows and brspace -c force -f dbparam -p log_archive_dest -v /oracle/<SID>oraarch/<SID>arch on UNIX.
5. To check whether everything is fine, perform a manual log switch with brspace -c force -f dbalter -a switchlog and check whether a new archived redo log file is created in the new orarch directory.

- Delete the dummy file.

  When an archiver stuck occurs, perform the following steps:

  1. Delete the dummy file. After deleting the dummy file, the archiver stuck is immediately resolved and the database continues to work.
  2. Run brarchive -cds two times.

The second run of BRARCHIVE is necessary because the first run only deletes archived redo logs, which are already backed up the day before and, therefore, will not release much disk space.

- Change the archive directory.

  BRARCHIVE 7.00 is able to find and save offline redo log files after changing the archive directory, which offers the simple procedure for resolving an archiver stuck issue.

  Procedure for changing the archive directory:

  1. Use BRSPACE to change the value of the Oracle parameter `log_archive_dest[_1|_2]` so that it points to a new archive directory with enough free space.
  2. Start BRARCHIVE to save and delete offline redo log files. Make sure that the offline redo logs can be saved twice on different media. To do this, run `brarchive -cds` two times.
  3. Change the value of the parameter `log_archive_dest` back to the original value.

Online Backup Crashed

During an online backup, tablespaces are set into backup mode to enable proper recovery whenever a data file from an online backup needs to be restored. BRBACKUP ensures that after an online backup, the backup mode of all tablespaces backed up is turned off again.

If BRBACKUP crashes, one or more tablespaces remain in backup mode. A lock file of BRBACKUP can prevent further backups to start.

In this situation, the following conditions occur:

- The database cannot be shut down to a consistent state; only a shutdown abort is possible.

> Caution:
> In this case, do not shut down the database with Shutdown Abort. If you do so, the next start-up will not work and additional actions are required.

- Any other BRBACKUP can fail.
- A database crash or shutdown abort leaves the database in a state where a normal startup fails.

While a tablespace is in backup mode, `brconnect -f check` issues the warning – `WARNING, type: TABLESPACE_IN_BACKUP, object: PSAPT00EX`. To check whether a tablespace is currently in backup mode, you can use the faster and cheaper command `brspace -c force -f dbshow -t tslist`. Check the *Back* column to see whether any tablespace is in backup mode.

If any tablespace is in backup mode, but you expect no BRBACKUP to be running, perform the following steps to solve the problem:

1. Use the `ps` command on UNIX or Task Manager on Windows to check whether BRBACKUP is running. Continue only if no BRBACKUP is running.
2. Start BRTOOLS or BRGUI and choose *Space management* → *Alter tablespace* → *Reset backup status*.

3. Choose *Continue* two times to get a list of all tablespaces in backup mode.
4. Enter **0** to select all tablespaces and choose *Continue* to turn backup mode off.
5. Delete the lock file `$SAPDATA_HOME/sapbackup/.lock.brb`.
6. Run a test backup with BRBACKUP to check whether everything is fine.

Sometimes, the whole database or server crashes during an online backup, for example, when a reboot is executed while BRBACKUP is running. In this case, a normal startup is not possible.

When you start the database, the following error message appears:

```
...
SQL> ORACLE instance started.

Total System Global Area   48307140 bytes
Fixed Size                   453572 bytes
Variable Size              41943040 bytes
Database Buffers            5734400 bytes
Redo Buffers                 176128 bytes
Database mounted.
ORA-01113: file 5 needs media recovery
ORA-01110: data file 5: 'D:\ORACLE\T99\SAPDATA3\T99_1\T99.DATA1'
...
```

You can solve this issue by performing a recovery on the database using BRRECOVER. If no other problem has occurred, an unattended recovery using the command `brrecover -c force -t complete` solves the issue without further prompts. To perform the recovery using BR*Tools or BR*GUI, start BRRECOVER and choose *Restore and Recovery → Complete database recovery*.

> Hint:
> The issue can also be solved without performing a recovery, which can take a long time depending on the database activity and the duration of the tablespace being in backup mode. To avoid a long recovery, use SQL*Plus and startup the database into the MOUNT state. Then, execute the command `ALTER DATABASE END BACKUP` and open the database.

> Caution:
> The `command ALTER DATABASE END BACKUP` must only be used when no data file has been restored. Do not use this command after having restored any data file. Doing so can bring your database into an inconsistent and unusable state.

Unit 4
Exercise 13

Perform Housekeeping and Troubleshooting

Business Example

During an online backup, the server crashed and you must get the database back online.

Simulate a server crash during online backup and get the database running again.

1. Simulate a server crash during online backup by setting tablespace PSAP<DBSID> in backup mode and shut down the database with the *ABORT* option. Try to start the database and check the error message.

2. Run the database again.

Perform Housekeeping and Troubleshooting

Business Example

During an online backup, the server crashed and you must get the database back online.

Simulate a server crash during online backup and get the database running again.

1. Simulate a server crash during online backup by setting tablespace PSAP<DBSID> in backup mode and shut down the database with the *ABORT* option. Try to start the database and check the error message.
   a) Start BRGUI or BRTOOLS and choose *Space management* → *Alter tablespace* → *Set backup status*.
   b) Choose *PSAP<DBSID>* under the *Tablespace* column.
   c) Choose *Continue* to put it in backup mode.
   The following menu appears:

```
BR0657I Input menu 313 - please enter/check input values
----------------------------------------------------------------
--------------
Options for alter of tablespace PSAPT99

 1 * Current tablespace status (status) . [ONLINE]
 2 * Current backup status (backup) ..... [NO]
 3 * Alter tablespace action (action) ... [begback]
 4 # Set offline mode (mode) ............ [normal]
 5 # New tablespace name (name) ......... []
 6 # Force tablespace alter (force) ..... [no]
 7 - SQL command (command) .............. [alter tablespace
PSAPT99 begin backup
]

Standard keys: c - cont, b - back, s - stop, r - refr, h - help
----------------------------------------------------------------
--------------
BR0662I Enter your choice:

...
...
BR1016I SQL statement 'alter tablespace PSAPT99 begin backup'
executed successfu
lly
BR1073I Tablespace PSAPT99 altered successfully into status
'begback'
...
...
```

   d) To shut down the database with the abort option, start BRTOOLS or BRGUI and choose *Instance Management* → *Shut down database*.

e) Choose *Continue*.

f) In *Options for shutting down database instance <DBSID>*, enter **abort** in the *Database close mode (mode)* parameter and choose *Continue*.

g) Try to start the database again by starting BRGUI or BRTOOLS and choosing *Instance management → Start up database*. The following error message displays:

```
...
...
BR0304I Starting and opening database instance T99 ...
BR0278E Command output of 'D:\oracle\DEV\112\BIN\sqlplus /nolog
&lt; D:\ORACLE\T99\sapreorg\.segbbrkv.spi':

SQL*Plus: Release 11.2.0.1.0 Production on Di Jun 7 11:08:28 2011

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

SQL> SQL> SQL> Connected to an idle instance.
SQL>
SQL> ORA-32004: obsolete or deprecated parameter(s) specified
for
RDBMS instance

ORACLE instance started.

Total System Global Area  292319232 bytes
Fixed Size                  2174888 bytes
Variable Size             268435544 bytes
Database Buffers           16777216 bytes
Redo Buffers                4931584 bytes
Database mounted.
ORA-10873: file 4 needs to be either taken out of backup mode or
media recovered

ORA-01110: data file 4: 'D:\ORACLE\T99\SAPDATA3\T99_1\T99.DATA1'


SQL> Disconnected from Oracle Database 11g Enterprise Edition
Release
11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options
BR0280I BRSPACE time stamp: 2011-06-07 11.08.34
BR0279E Return code from 'D:\oracle\DEV\112\BIN\sqlplus /nolog
&lt; D:\ORACLE\T99\sapreorg\.segbbrkv.spi': 0
BR0302E SQLPLUS call for database instance T99 failed
BR0306E Start and open of database instance T99 failed
...
...
```

2. Run the database again.

a) The recommended way to get the database running again is to use BRRECOVER, using the scenario *Complete database recovery*. To recover the database, start BRGUI or BRTOOLS and choose *Restore and recovery → Complete database recovery*.

b) Check the actions performed by BRRECOVER. Choose *Continue* at any prompt to successfully complete the recovery without further input.

LESSON SUMMARY

You should now be able to:

- Perform housekeeping and troubleshooting

Learning Assessment

1. A tablespace is full. How can you extend the tablespace?

   Choose the correct answers.

   - [ ] A There is no way to extend a tablespace; therefore, a new tablespace has to be created.
   - [ ] B Another data file can be added to the tablespace.
   - [ ] C The properties of an existing data file can be changed to be autoextensible.
   - [ ] D An existing data file can be manually resized.

2. Which options are provided by the BR*Tools to change the properties of a tablespace?

   Choose the correct answers.

   - [ ] A Rename tablespace
   - [ ] B Set tablespace offline and online
   - [ ] C Shutdown tablespace
   - [ ] D Merge tablespaces

3. A distinction is made between offline and online reorganization. In which method of reorganization it is not always possible to access the objects?

   Choose the correct answer.

   - [ ] A Online
   - [ ] B Alter table
   - [ ] C Offline
   - [ ] D Segment shrinking

4. In which situations reorganizations can be useful?

   Choose the correct answers.

   - [ ] A Change the name of certain tables.
   - [ ] B Release free space permanently.
   - [ ] C Move certain large and heavily used tables into separate tablespaces.
   - [ ] D Change properties of a tablespace like locally-managed tablespaces.
   - [ ] E Change properties of a table like storage parameters or compression.

5. An archiver stuck is a situation in which the archiver instance process cannot copy an online redo log file to the archive directory and the online redo log file to be copied needs to be overwritten by the log writer.

   Determine whether this statement is true or false.

   - [ ] True
   - [ ] False

6. Which of the following features are parts of the Oracle advanced compression option?

   Choose the correct answers.

   - [ ] A OLTP table compression
   - [ ] B Statistics compression
   - [ ] C Data file compression
   - [ ] D RMAN compression

Unit 4

Learning Assessment - Answers

1. A tablespace is full. How can you extend the tablespace?

   Choose the correct answers.

   - [ ] A There is no way to extend a tablespace; therefore, a new tablespace has to be created.
   - [x] B Another data file can be added to the tablespace.
   - [x] C The properties of an existing data file can be changed to be autoextensible.
   - [x] D An existing data file can be manually resized.

2. Which options are provided by the BR*Tools to change the properties of a tablespace?

   Choose the correct answers.

   - [x] A Rename tablespace
   - [x] B Set tablespace offline and online
   - [ ] C Shutdown tablespace
   - [ ] D Merge tablespaces

3. A distinction is made between offline and online reorganization. In which method of reorganization it is not always possible to access the objects?

   Choose the correct answer.

   - [ ] A Online
   - [ ] B Alter table
   - [x] C Offline
   - [ ] D Segment shrinking

4. In which situations reorganizations can be useful?

   Choose the correct answers.

   - [ ] A Change the name of certain tables.
   - [x] B Release free space permanently.
   - [x] C Move certain large and heavily used tables into separate tablespaces.
   - [x] D Change properties of a tablespace like locally-managed tablespaces.
   - [x] E Change properties of a table like storage parameters or compression.

5. An archiver stuck is a situation in which the archiver instance process cannot copy an online redo log file to the archive directory and the online redo log file to be copied needs to be overwritten by the log writer.

   Determine whether this statement is true or false.

   - [x] True
   - [ ] False

6. Which of the following features are parts of the Oracle advanced compression option?

   Choose the correct answers.

   - [x] A OLTP table compression
   - [ ] B Statistics compression
   - [ ] C Data file compression
   - [x] D RMAN compression