

Database Administration (Oracle) with SAPDBA



HELP.BCDBORADBA

Release 4.6C



Copyright

© Copyright 2001 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft[®], WINDOWS[®], NT[®], EXCEL[®], Word[®], PowerPoint[®] and SQL Server[®] are registered trademarks of Microsoft Corporation.

IBM[®], DB2[®], OS/2[®], DB2/6000[®], Parallel Sysplex[®], MVS/ESA[®], RS/6000[®], AIX[®], S/390[®], AS/400[®], OS/390[®], and OS/400[®] are registered trademarks of IBM Corporation.

ORACLE[®] is a registered trademark of ORACLE Corporation.

INFORMIX[®]-OnLine for SAP and Informix[®] Dynamic Server[™] are registered trademarks of Informix Software Incorporated.

UNIX[®], X/Open[®], OSF/1[®], and Motif[®] are registered trademarks of the Open Group.







HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C[®], World Wide Web Consortium, Massachusetts Institute of Technology.

JAVA[®] is a registered trademark of Sun Microsystems, Inc.

JAVASCRIPT[®] is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, SAP Logo, R/2, RIVA, R/3, ABAP, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPHIRE, Management Cockpit, mySAP.com Logo and mySAP.com are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other products mentioned are trademarks or registered trademarks of their respective companies.

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax
	Tip

Contents

Database Administration (Oracle) with SAPDBA	15
New Features in SAP R/3 Release 4.0	16
New Features in SAP R/3 Release 4.5	18
SAPDBA.....	19
BRBACKUP, BRARCHIVE and BRRESTORE.....	20
New Features in SAP R/3 Release 4.6	21
Sources of Information for the Database Administrator	23
Computing Center Management System	24
R/3 Tables in the ABAP Dictionary	25
Database Utility.....	26
SAPNet.....	27
EarlyWatch Service	28
Setting Up the Database System	29
SAPDBA: Archive mode.....	31
ARCHIVELOG Mode	33
Archiving Parameters.....	34
Mirroring the Control File.....	36
Mirroring Online Redo Log Files	37
Organizing Disk Storage.....	38
Database Parameters	39
Tablespaces and SAP Naming Conventions	41
Using SAPDBA.....	43
Configuring SAPDBA	44
Environment Variables (UNIX)	45
Directory Structure (Unix).....	47
Configuring the Display Length	48
Configuring the UNIX Command at.....	49
Setting the Log Checks Option.....	50
Users and Roles	51
Database Users.....	54
UNIX User Configurations	56
SAPDBA: Expert Mode.....	60
Defining the Password	62
Restricted Session.....	63
SAPDBA and Autoextend.....	64
Starting SAPDBA	65
SAPDBA: Release Information.....	68
SAPDBA Command Options	69
-analyze	70
-check	72
-check_db_open.....	73
-checkopt.....	74
-cleanup	76
-delete.....	77

-export.....	78
-fill_tables	79
-next.....	80
-r	81
-shutdown(_abort)	82
-startup(_dba).....	83
-version.....	84
Return Codes for SAPDBA Command Options.....	85
SAPDBA: Initial Menu.....	86
Displaying and Managing Logs and Profiles	87
SAPDBA: Show/Cleanup.....	88
Deleting Log and Trace Files	89
Support for the Cost-Based Optimizer	91
SAPDBA: DB Optimizer Control.....	93
SAPDBA Internal Rules for Determining Statistics	95
Starting up and Shutting down the Database System	97
Status of the Database System.....	98
Starting Up the Database System	100
Shutting down the Database System	101
Backup Concepts	103
Logical Backup and Image Backup	104
Backup Methods and Backup Frequency	105
Complete Backups	107
Incremental Backup.....	108
RMAN Backup Strategies.....	110
Incremental Backups After Structural Changes	112
Restoring Incremental Backups with RMAN	113
Backup Media.....	114
Tape Devices with Hardware Compression	115
Software Compression.....	117
Backup to Multiple Disks	119
Backup to a Remote Disk.....	120
Backup to a Remote Tape Device	123
Two-Phase Backup.....	124
Online Mode and Offline Mode.....	127
Consistent Online Backup	128
Structure-Retaining Database Copy	130
Backing up Executable Programs and Other Components of the R/3 System.....	132
Backing Up Data Tablespaces	133
Duration of the Backup	134
Parallel Backup	135
BRBACKUP and BRARCHIVE Backups in One Run.....	136
Unattended Backup	139
Volume Management.....	142
Volume Initialization.....	143
Volume Label Check	145
Scratch Volumes	146
Volume Expiration Period.....	147

Used Volumes	149
Selecting Volumes Manually	151
Selecting Volumes with External Tools	152
Automatic Volume Management	153
Backup Verifications	155
Verification of RMAN Backups	157
Integration of the Recovery Manager	158
RMAN Backups with the SAP Backup Library	160
Backup with an External Backup Library	162
Incremental Backup Strategy Without a Backup Library	164
Saveset Grouping	167
Tape Layout	169
Profile Parameters and BRBACKUP Command Options	170
Backup of Large Oracle Databases	173
Hardware and Duration Example for a 100 GB Backup	174
Backup Devices for Large Databases	175
Backing Up Large Databases to Tape with BRBACKUP	177
External Backup Programs for Large Databases	178
Parallel Backup of Large Databases to Disk with BRBACKUP	179
Optimization of File Distribution	180
Optimization with a Logical Volume Manager	182
Partial Backups	184
Standby Database Scenario	185
Standby Database Configuration	186
Standby Database: Support by BRARCHIVE	188
Standby Database: Backup with BRBACKUP	190
Split Mirror Disks Backup	192
Split Mirror Online Backup	193
Split Mirror Offline Backup	194
Split Mirror Backup: Software Configuration	195
Backup with Jukebox/Autoloader	197
Mount / Dismount Command	198
Example: Backup with an Autoloader	200
Backing Up the Database with SAPDBA	202
SAPDBA: Backup Database	204
SAPDBA: Backup Database: Special Options	208
Archiving the Offline Redo Log Files	210
SAPDBA: Backup Offline Redo Logs	212
SAPDBA: Backup Offline Redo Logs: Special Options	215
Grouping the Offline Redo Log Files	217
SAP Backup Utilities	218
Using BRBACKUP and BRARCHIVE	219
Supported Backup Media	221
Automatic Tape Changing	222
Effects of the Command Options	223
cpio Continuation Tape	224
cpio Error	225
Canceling a Backup	226
BRBACKUP	227
Backing up Non-Database Files and Directories	228

Completion of BRBACKUP Backups.....	229
Backing up Database Files.....	231
Hardware Compression.....	232
BRARCHIVE.....	233
Hardware Compression.....	234
Logging.....	235
Using BRRESTORE.....	236
Restoring Files.....	237
Completion of BRRESTORE Runs.....	238
Examples.....	239
BRCONNECT and BRTOOLS.....	240
Using External Backup Programs.....	241
Advantages.....	243
Companies that Support the Interface to External Backup Programs.....	244
The SAP Utilities under Windows NT.....	245
SAP Conventions (Windows NT).....	246
Environment Variables (Windows NT).....	247
Directory Structure (Windows NT).....	249
Directory Tree (Windows NT).....	251
Naming Conventions for Files (Windows NT).....	252
Executables.....	253
Starting the SAP Utility Programs.....	254
Reorganization (Windows NT).....	256
Database Analysis.....	257
Restrictions on SAPDBA (Windows NT).....	258
Backup Strategy (Windows NT).....	259
NTBackup.....	260
BRBACKUP/BRARCHIVE.....	261
Other Backup Programs.....	262
Initialization Profile init<DBSID>.dba.....	263
backup_util_name.....	264
blkdev_dir.....	265
check_controlfile_mirror.....	266
check_initial_settings.....	267
check_redolog_mirror.....	268
chop_util_name.....	269
exe_dir.....	270
exireo_dumpdir.....	271
exireo_exp_imp_buf.....	272
expir_period_brarchive.....	273
exp_imp_degree.....	274
file_or_raw.....	275
help_<xy> and language.....	276
index_degree.....	277
max_file_size.....	278
message_buf.....	279
ops.....	280
orapwd_always_used.....	281

parallel_instance	282
rawdev_dir	283
recovery_degree	284
sqldba_path	285
trace_level	286
tspadd_tspname	287
Sample Initialization Profile	288
Overview of Script Types	289
Starting SQL Scripts Manually with SQLPLUS	290
SAPDBA Scripts and Files	291
Starting Individual SQL Scripts with SAPDBA	294
Starting Shell Scripts for ORACLE Export/Import	295
SAPDBA Logs	296
Log Files and Their Contents	297
Structure Log	299
Central Log reorg<DBSID>.log	300
RC.....	302
Logging of Parallel Export/Import Processes	304
Connection to CCMS	305
Initialization Profile init<DBSID>.sap	306
Using the Profile init<DBSID>.sap	307
Parameter Format	308
Description of Parameters	309
archive_copy_dir.....	311
archive_function.....	312
archive_stage_dir.....	313
backup_dev_type.....	314
backup_mode.....	316
backup_root_dir.....	318
backup_type.....	319
compress.....	320
compress_cmd.....	321
compress_dir.....	322
copy_in_cmd.....	323
copy_out_cmd.....	324
cpio_disk_flags.....	325
cpio_flags.....	326
cpio_in_flags.....	327
dd_flags.....	328
dd_in_flags.....	329
dismount_cmd.....	330
exec_parallel.....	331
expir_period.....	332
mount_cmd.....	333
mount_par_file.....	334
new_db_home.....	335
parallel_instances.....	336
primary_db.....	337
remote_host.....	338

remote_user	339
resync_cmd	340
rewind	341
rewind_offline	342
saveset_members	343
split_cmd	344
stage_copy_cmd	345
stage_root_dir	346
tape_address	347
tape_address_arch	348
tape_address_ctl	349
tape_address_ctl_arch	350
tape_address_rew	351
tape_address_rew_arch	352
tape_copy_cmd	353
tape_pos_cmd	355
tape_size	356
tape_size_arch	357
tape_use_count	358
uncompress_cmd	359
util_par_file	360
volume_archive	361
volume_backup	362
Command Options for BRBACKUP, BRARCHIVE, and BRRESTORE	363
Command Options for BRBACKUP	364
-a -archive	366
-b -backup	367
-bd -backup_delete	368
-c -confirm	369
-d -device	371
-e -execute	372
-f -fillup	373
-h -help	374
-i -initialize	375
-k -compress	376
-l -language	377
-m -mode	378
-n -number	380
-o -output	381
-p -profile	382
-q -query	383
-r -parfile	384
-s -saveset	385
-t -type	386
-u -user	387
-v -volume	388
-w -verify	389
Command Options for BRARCHIVE	390
-a -archive	392
-b -backup	393
-c -confirm	394
-d -device	395

-f -fill	396
-h -help	397
-i -initialize	398
-k -compress	399
-l -language	400
-m -modify	401
-n -number	402
-o -output	403
-p -profile	404
-q -query	405
-r -parfile	406
-s -sc -ds -dc -sd -scd -ss -ssd -cs -cbs	407
-sd -ds -dc -scd -cbs -ssd	409
-u -user	410
-v -volume	411
-w -verify	412
Command Options for BRRESTORE	413
-a -archive a1 archive1	415
-a2 -archive2	416
-b -backup	417
-b2 -backup2	418
-c -confirm	419
-d -device	420
-e -execute	421
-f -fillup	422
-h -help	423
-incr	424
-k -compress	425
-l -language	426
-m -mode	427
-n -number	428
-o -output	429
-p -profile	430
-q -query	431
-r -parfile	432
-w -verify	433
BRBACKUP, BRARCHIVE, and BRRESTORE Logs	434
Log Types	435
File system logs	436
Database Logs	437
BRBACKUP Logs	438
Names of the Detail Logs	439
Detail Log	440
Log Supplements	442
Summary Log	444
BRARCHIVE Logs	445
Names of the Detail Logs	446
Detail Log	447
Summary Log	449
BRRESTORE Logs	451
Names of the Detail Logs	452
Detail Log	453

Sample Logs	454
Volume Initialization.....	455
Detail Log of an Offline Backup.....	457
Supplements to a Detail Log	464
Log for an RMAN Preparation Run	467
Summary BRBACKUP Log	476
Detail BRARCHIVE Log	478
Summary BRARCHIVE Log	483
BRRESTORE Logs: Restoring a Complete Backup	487
Messages and Return Codes: BRBACKUP, BRARCHIVE, and BRRESTORE.....	493
Return Codes for BRBACKUP / BRRESTORE.....	494
Messages.....	495
Messages from BR002I to BR028E	496
Messages from BR029E to BR70W	501
Messages from BR071E to BR115I	506
Messages from BR116I to BR151E	511
Messages from BR152I to BR181E	516
Messages from BR182E to BR218E	521
Messages from BR219W to BR241I	527
Messages from BR242I to BR269W	532
Messages from BR270W to BR307I	537
Messages from BR308I to BR339W	542
Messages from BR340I to BR383E	547
Messages from BR384E to BR429W	552
Messages from BR430E to BR464W	557
Messages from BR465E to BR468I	561
Messages from BR501I to BR607E	564
Database Reorganization.....	575
Reorganization Concepts	576
Status of the Database.....	577
Limitations of the Database System.....	579
Reorganization Type	581
Reorganization of a Single Object.....	583
Reorganization of an Object List.....	584
Reorganization of a Tablespace	586
Reorganization of Special Tablespaces.....	587
Reorganization of a Tablespace with Data Files.....	588
Moving/Renaming the Data Files of a Tablespace	590
Resizing the Data Files of a Tablespace.....	591
Procedure for a Reorganization	592
Procedures in Detail.....	594
Backup Concepts in Reorganizations	595
Freespace Check	597
Freespace Check during Reorganization of a Single Object.....	598
Freespace Check during Reorganization of a Tablespace.....	599
Freespace Check during Reorganization of a Tablespace with Data Files	600
Freespace Check During CREATE TABLE AS SELECT, in Deferred Mode and During INDEX REBUILD	601
SAPDBA: Reorganization	602

Reorganization	603
Before a Reorganization.....	604
Checking Statistics	605
Extent Check Functions	606
Command Option -analyze	608
Number of Data Files	610
Analyzing the Statistics	611
Changing MAXEXTENTS and NEXT	612
Command Option -next.....	613
Carrying out a Reorganization	614
SAPDBA: Reorganization Menu.....	615
Changing the Extent Size.....	620
Moving a Table or Index.....	622
Performance Maximization during Reorganization.....	623
Speeding Up a Reorganization	624
Archiving Mode during the Reorganization	625
Reorganizing in NOARCHIVELOG Mode	626
Performing Processes in Parallel	627
Restarting a Reorganization	628
Preparing for the Restart	630
Restart Procedure	631
Restart File for Reorganizations	632
Resetting a Reorganization	634
Data Transport During a Reorganization	635
Reorganization with Create table ... as select.....	636
Exporting and Importing Database Objects.....	637
Data Consistency.....	638
SAPDBA: Export/import.....	639
Export Types	641
Parallel Export/Import.....	643
SAPDBA: Export Menu.....	644
Exporting Tables with Indexes and Data.....	648
Prerequisites for Export and Import.....	649
Export Options for Tables and Indexes with Data	650
Exporting Table and Index Definitions	652
Export Options for Tables and Indexes	653
Exporting Constraints, Synonyms, Sequences, Views, and Grants	655
Export Options for Constraints, Synonyms, Sequences, Views, and Grants.....	656
Importing Tables with Indexes and Data.....	658
Import Options for Tables with Data	659
Structure Import.....	660
Import Options for Object Structures	661
Restarting the Export/Import	662
Restart File for Export/Import	663
SAP unload/load, SQL*Loader	665
Changing Storage Parameters	668
Overview of the Storage Parameters.....	669
Options for Changing and Checking the Storage Parameters	671

Compress Extents	672
Default Value for NEXT	673
Checking Freespace.....	675
Reducing the Size of the Table or Index	676
Changing the Storage Parameters Manually	677
Tablespace Administration.....	678
Tablespace Administration: Overview	679
Backup Recommended	680
SAPDBA: Tablespace Administration	681
Checking Tablespaces	682
Freespace Statistics	683
Check Functions	684
Files/Raw Devices of a Tablespace	685
Extending a Tablespace.....	686
Specifying the File Size.....	689
Naming Convention.....	691
New Path.....	692
SAPDBA Troubleshooting	693
Creating a Tablespace.....	694
SAPDBA Database Recovery	697
SAPDBA: Check (and Repair) Database	699
Recovering the Database Automatically	700
Checking the Database Files	701
Finding the Backup Files	703
Restoring the Backup Files.....	705
Changing the Restore Parameters.....	706
Finding Incremental Backups	707
Restoring Incremental Backups	709
Finding the Archived Redo Logs	710
Restoring Archived Redo Logs.....	712
Recovering the Database	713
Full Restore and Recovery	714
Resetting the Database	721
Restoring One Tablespace	722
Restoring Individual Files	723
Recovery with SVRMGR	726
Recovery with SVRMGR: Overview	727
Types of Database Errors	728
Error Analysis	730
Recovery after User Errors	731
Recovery after Statement Errors	732
Recovery after Process Errors.....	733
Recovery after an Instance Error	734
Recovery after Media Errors.....	735
Recovery: One Control File Missing.....	737
Recovery: All Control Files Missing	738
Recovery: Current Online Redo Log Missing.....	742

Recovery: One Inactive Online Redo Log Missing	744
Recovery: User Tablespace Missing	746
Recovery: SYSTEM Tablespace Missing	749
Recovery: Index Tablespace Missing.....	751
Recovery: Errors During the Archiving of Online Redo Logs	752
Partial Recovery.....	754
Actions after a Partial Recovery	757
Automating the Recovery	759
Updating the Control File.....	760
Disaster Recovery	762
Oracle Databases on Raw Devices	766
SAPDBA Support for Raw Device Configuration	767
Raw Devices and BRBACKUP/ BRRESTORE	768
ORACLE Databases and Parallel Server Option	769
OPS with BRBACKUP, BRARCHIVE and BRRESTORE.....	770
Information in the Profile init<DBSID>.sap	771
Special Features of BRBACKUP	772
Special Features of BRARCHIVE	773
Special Features of BRRESTORE	774
Distributed Scenario.....	775
Central Scenario	778
Mixed Scenario	780

Database Administration (Oracle) with SAPDBA

Purpose

The administration tool SAPDBA for Oracle and its backup tools BRBACKUP, BRARCHIVE and BRRESTORE support the database administrator both in daily routine tasks and in less frequent, complex tasks, such as recovering or reorganizing the database. You can call up the many SAPDBA functions from an ASCII interface, or you can use a command line interface (CLI) to configure and execute the functions individually. SAPDBA also offers extensive certified interface solutions for external backup tools (BACKINT) and integrates the latest in database technology (including standby databases, EMC Timefinder).

Features

SAPDBA includes the following, high-performance, fully integrated components for administrating multiple Version 7.x and 8.x Oracle databases:

- Database backup, restore and recovery
- Database system check
- Database reorganization
- Cost-based optimization of access

R/3 Integration

- You can use the SAPDBA functions from the Computing Center Management System (CCMS).
- SAPDBA meets the R/3 System's application-specific requirements.
- SAPDBA is available independently of the R/3 System.
- SAPDBA is delivered as standard with the R/3 System.

Conditions of License



If you have acquired the SAP tools SAPDBA, BRBACKUP, BRARCHIVE and BRRESTORE with the R/3 System or with other SAP products, then you may use these tools only in accordance with your conditions of use of the R/3 System and other SAP products.

New Features in SAP R/3 Release 4.0

New Features in SAP R/3 Release 4.0

- [CCMS Support for the Cost-Based Optimizer: Oracle](#)
[\[Ext.\]](#) Use *DB Optimizer Control Panel* in the SAPDBA menu or the command option `sapdba -checkopt` to find missing or obsolete statistics for the CBO. You can then determine new statistics with the command option `-analyze` or with CCMS in the R/3 System.

See also: [SAPDBA Command Options \[Page 69\]](#), [Support of the Cost-Based Optimizer \[Page 91\]](#)
- [Database System Check](#)
[\[Ext.\]](#) The check function *DB System Check* (`sapdba -check`) has been extended. As well as checking space and fragmentation this function also checks the physical consistency of the database and alerts (errors) on the database. You can also make these checks with the CCMS in the R/3 System.
- [Consistent Online Backup \[Page 128\]](#)
You can use a consistent online backup to reset a database. This new type of backup (`backup_type = online_cons`) is logically consistent (data files and online redo log files) which allows you to recover the database completely.
- Support for a [two-phase backup \[Page 124\]](#) of a disk backup to tape
The backup has two phases, which you can run independently of each other; in the first phase back up the database to disk (*Backup database*), and in the second phase copy this disk backup to tape (*Backup from disk backup*).
- [Completion of BRBACKUP Backups \[Page 229\]](#) and [Completion of BRRESTORE Runs \[Page 238\]](#)
You can complete terminated or incomplete backups (partial backups). To do this, use the SAPDBA menu option: *Backup database – Restart backup or Make partial backups complete*. SAPDBA (dynamic menu) also supports you when you want to complete canceled restorations without having to repeat the whole restoration process (this does not apply to the restoration of offline redo log files, or to data files in a partial restore and full recovery).
- [Backup Verifications \[Page 155\]](#)
You can check the Oracle block structure of restored backups or data files (online) with ORACLE DB_VERIFY (*DB Check/ Verification – DB verification using DB VERIFY*). You can also verify backup tapes at a later point in time (*Backup database - backup function - Verify BRBACKUP tape*).
- [Data Transport During a Reorganization \[Page 635\]](#)
For the data transport in a reorganization you can speed up the process by using one of the following methods as an alternative to ORACLE export/import:
 - SAP unload/ ORACLE SQL*Loader
 - SAP unload/ SAP loader
 - CREATE TABLE AS SELECT
- *Object handling - Rebuild* in the [SAPDBA Reorganization Menu \[Page 615\]](#)
If you are just reorganizing the index you can use the old index to create a new index (*ALTER INDEX REBUILD*). The old index is only deleted after the reorganized index has been successfully restructured.

New Features in SAP R/3 Release 4.0

- *Object handling - HideTab* in the [SAPDBA Reorganization Menu \[Page 615\]](#)
You can temporarily rename the tables you are reorganizing. This stops these tables being accessed during the reorganization process.

New Features in SAP R/3 Release 4.5

New Features in SAP R/3 Release 4.5

[SAPDBA \[Page 19\]](#)

[BRBACKUP, BRARCHIVE, BRRESTORE \[Page 20\]](#)

SAPDBA

- [Reorganization \[Page 615\]](#)

When you reorganize individual tables (*single reorganization*) the table is no longer deleted directly after the import, instead it is renamed. The old data is deleted only after the reorganized new tables have been created and imported. This deferred mode increases data security.
- [CCMS Support for the Cost-Based Optimizer \[Ext.\]](#)

You can schedule all actions for checking and updating statistics in CCMS.
- [Checking the Database System \[Ext.\]](#)
- The database system check has been extended with new *DBA*, *ORA* and *PROF* parameters (such as *TABLES_NOT_IN_TABLE_TABLESPACE*).
- You can check the database system while the database is mounted.
- You can individually specify how often you want the check to be repeated for each parameter of the database system check.
- You can display the results of the database system check in different ways (for example, you can display the errors only).
- Database Recovery: Support for Incremental Backups with RMAN
The SAPDBA *restore/recovery* logic takes into account incremental backups made with BRBACKUP/Oracle Recovery Manager (RMAN).
- *AUTOEXTEND* and *RESIZE* (Oracle)
As of Release 4.5A, SAPDBA supports the Oracle options *AUTOEXTEND* and *RESIZE*. This means that backups of files whose sizes have changed due to *AUTOEXTEND* or *RESIZE* can be used for *Restore* and *Recovery* functions. For more information, see [SAPDBA and Autoextend \[Page 64\]](#).
- SAPDBA Help
You can call the HTML version of the SAP online documentation directly from the SAPDBA menu (*SAP Online Help – SAP Online Documentation*).
- New Command Option
Use the command option `-V | -VERSION` to obtain patch information.

BRBACKUP, BRARCHIVE and BRRESTORE

BRBACKUP, BRARCHIVE and BRRESTORE

- [Integration of the Recovery Managers into BRBACKUP/ BRRESTORE \[Page 158\] \[Page 108\]](#)
You can use the Oracle Recovery Manager (RMAN) to make an [incremental backup \[Page 108\]](#) of your database. In an incremental backup, only the changes that have been made since the last complete backup are saved. This strategy is particularly useful for large databases.
- **New Device Type: `backup_dev_type = stage | stage_copy | stage_standby`**
You can make backups to a remote disk, without NFS mount. The transmission protocols *ftp/ rcp* make this secure; you do not need to verify with `-verify`.
- **Hardware-Compressing File Systems Under Windows NT**
As well as hardware-compressing tape devices, hardware-compressing file systems under Windows NT (`compress = hardware`) are also supported. This applies to:
`backup_dev_type = disk | disk_copy | disk_standby`
- **Backup with Automatic Deletion**
In the second phase of the [two-phase backup \[Page 124\]](#) you can use BRBACKUP/BRARCHIVE to integrate automatic deletion of files into the backup. If you choose this, the second phase consists of the disk backup being backed up to tape, followed by the deletion of the successfully backed up files from the disk.
For BRBACKUP see [-bd|-backup_delete \[Page 368\]](#)
For BRARCHIVE see [-sd|-ds|-dc|-scd|-cds|-ssd \[Page 409\]](#)
- **New [Command Option for BRBACKUP, BRARCHIVE and BRRESTORE \[Page 363\]](#)**
Use the command option to obtain patch information.

New Features in SAP R/3 Release 4.6

SAPDBA

DB System Check

The R/3 database systems can now use new parameters for the message types DBA, ORA and PROF to make even more detailed checks. Critical situations in the database can be detected early, and downtime can be avoided.

- The parameter FS_FULL of the message type DBA checks, for example, to see whether free memory is under the specified threshold in all `SAPDATA` directories, or in any directories you want to configure.
- The parameters of the new message type DBO determine database operations (such as backups) that have caused errors since the last backup (return code > 1), or since a specified date (*last success to old*).

You can also set certain parameters for specific objects.

- To overwrite the definition of the parameter, you can set thresholds for individual objects.
- You can define negative lists for parameters to exclude individual objects from the check.

For more information, see [Database System Check \[Ext.\]](#).

Reorganizing Single Tables Online

With this new release you can reorganize single tables when the R/3 System is running (online). A shadow table registers all changes to the table (inserts, updates and deletes). After the reorganization, the table is locked briefly to copy the changes stored in the shadow table.

For more information, see [Database Reorganization \[Page 575\]](#).

Cost-Based Optimizer

- The integration of the serial processes *Check for old statistics* (`sapdba -checkopt`) and *Update old statistics* (`sapdba -analyze`) into one process simplifies and speeds up the creation of statistics for the cost-based optimizer on NT platforms.



The integration of the serial processes *Check for old statistics* and *Update old statistics* for Unix platforms is planned for later releases.

- The runtime when creating statistics for large tables has been reduced by extending the internal rules.
- Due to their special attributes, the tables of *SAP New Dimension Products* (such as *SAP BW*) are handled differently.

For more information, see [Support for the Cost-Based Optimizer \[Page 91\]](#).

New Patch Concept

Read the Notes:

- 126769 for corrections in SAPDBA and the BR programs (for all releases)

New Features in SAP R/3 Release 4.6

- 141999 for new SAPDBA features (for all releases)

Sources of Information for the Database Administrator

To avoid error situations or bottlenecks in the database, you need to know where to find certain database administration information that goes beyond the scope of this documentation.

- Read the Oracle documentation on all important database processes.
- Read the Release Notes that appear with each new R/3 Release. To do this, choose *Help* → *Release Notes* in the R/3 System. The introductory chapters of this database guide tell you about the [New Features in the R/3 Releases \[Page 18\]](#).
- Read the sections of the R/3 Library on the [Computing Center Management System \[Page 24\]](#) to learn about how you can monitor a database and schedule and make backups.
- Read the sections of the R/3 Library on the ABAP Dictionary to learn about the conditions for creating [tables in the ABAP Dictionary \[Page 25\]](#) and on the database.
- Read about the [Database Utility \[Page 26\]](#) options.
- Use the SAPNet notes database.
- Read about the services of the [EarlyWatch Service \[Page 28\]](#).

Computing Center Management System

Computing Center Management System

You can make backups of the database and of the offline redo log files from the R/3 system. To do this, use the Computing Center Management System (CCMS). You can trigger the following actions from CCMS and then display the resulting logs.

- *Database offl. full bkup+archive logs*: Offline backup of the entire database using BRBACKUP followed by archiving of those offline redo log files that have not yet been archived using BRARCHIVE.
- *Database offline full backup*: Offline backup of the entire database using BRBACKUP.
- *Database online full bkup+archive logs*: Online backup of the entire database using BRBACKUP followed by archiving of the offline redo log files using BRARCHIVE to keep the backup consistent.
- *Database online full backup*: Online backup of the entire database using BRBACKUP.
 - *Archive redo log files*: Archiving of all offline redo log files not yet backed up from the archive directory to tape using BRARCHIVE.
- *Partial offline database backup*: Offline backup of specified tablespaces using BRBACKUP.
- *Partial online database backup*: Online backup of specified tablespaces using BRBACKUP.
- *Analyze tables in tablespace(s)*: Table analysis of a specified tablespace using SAPDBA (`sapdba -analyze`).
 - *Analyze tables via control table*: Analysis of the table statistics generated by SAPDBA (`sapdba -analyze`).
 - *Compute and adapt next table extents*: Adjustment of the values for the *NEXT* extent of tables in a specified tablespace with SAPDBA (`sapdba -next`).

You can find detailed information about planning and displaying backups and database analyses in the sections of the R/3 Library on the [Computing Center Management System \[Ext.\]](#). CCMS also offers several monitoring functions that you can use to monitor how your database system operates. For detailed information, see [The Monitoring Architecture: Concept \[Ext.\]](#) or [R/3 System Monitoring: Overview \[Ext.\]](#).

R/3 Tables in the ABAP Dictionary

As soon as you create a table, you can influence its storage parameters by maintaining its technical configuration. This technical configuration is used to optimize the space requirements and access response of individual tables. Always maintain at least the parameters *Data class* and *Size category* in the technical configuration for each table.

- **Data class**
The data class logically defines the tablespace where your table is stored. When you select the correct logical table type, your table will automatically be assigned to the correct database area when it is created. The F1 help for the field *Data class* provides information on how to select the proper value for the table. The assignment data class - tablespace (for data) is done using the table TAORA; the assignment data class - tablespace (for indexes) using table IAORA.
- **Size category**
The size category specifies the estimated space requirements of the table on the database using the categories 0 to 4. When you create a table an *INITIAL* extent is reserved on the database. If you need more space later, storage space is added corresponding to the selected category (*NEXT* extent). The F4 help on field *Size category* displays the number of data records that will fit in the assigned storage area of the database for each of the categories. Note that a maximum of 300 extents should be enough for storing table contents (with a database block size of 8 KB).

The assignment size category - extent size is done using an entry in the table DD09L and an analysis of table TGORA (for indexes using table IGORA). See also [Default Value for NEXT \[Page 673\]](#). These values of the ABAP Dictionary only represent starting values. For further database operations, use the options provided by SAPDBA for automatically adapting the size of the NEXT extent for all tables (for example, see [Command Option -next \[Page 613\]](#)).

If the table is defined logically, you will have to create it in the database in a second step. To do this, use the [database utility \[Page 26\]](#) options.

For more information on maintaining the technical configuration, see the documentation on the ABAP Dictionary.

Database Utility

Database Utility

Use

The database utility represents the interface between the ABAP Dictionary and the relational database that underlies the R/3 System. You can use the database utility to create, convert and delete ABAP Dictionary objects, as well as database table, database views and other R/3 objects. You can do this online or in the background. In a conversion, the definition of a table in the database is adapted to its changed definition in the ABAP Dictionary. You can also implement various analysis options, for example to display table and index definitions, or check the consistency of objects.

For some SAP objects, particularly transparent tables, you can set user-specific database parameters (INITIAL, NEXT, MINEXTENTS, MAXEXTENTS, tablespace name, FREELIST GROUPS, FREELISTS, PCTFREE, PCTUSED). You can also set flags to determine which parameters you want to apply the next time a table is created (deleted and created, converted) and which ones you want to take effect immediately. Not all the values can be changed immediately, just MAXEXTENTS, NEXT, PCTFREE and PCTUSED (these values are valid if a new area of memory is requested for the object).

Activities

To call the database utility, select *Utilities* → *Database utility* from the ABAP Dictionary initial screen. Note that an authorization for authorization object S_DDIC_OBJ (such as S_DDIC_ALL) is required to use the database utility. Make sure that the authorizations in your system are set such that only the database administrator is authorized to configure the database parameters. Check the settings and change them if necessary.

For more information on the database utility, see the documentation on the ABAP Dictionary.

SAPNet

Use

If problem situations occur in the system, SAPNet provides fast, effective help. In SAPNet, you can enter your questions to SAP directly and immediately receive an initial response. SAPNet provides you with the same information that SAP itself uses for First Level Support.

When you save the message you entered, the terminology contained in the message is used for an automatic search in the notes database. The system displays a list of notes, which contain detailed descriptions of how to solve the problem.

First and Second Level Support workers use incoming customer messages to write Notes. R/3 developers also create Notes if they recognize any potential sources of problems or missing information.

Activities

There are different ways to start a search for notes. Several are listed below:

- Specify the number of the note
- Enter the application area (for example BC-DB)
- Enter values for the change date
- Free text search: Enter an Oracle or SAP utility keyword



```
sapdba, brbackup, brarchive, brconnect, brrestore, restore,  
maxextents, offline redo logs, control file, stuck,  
archivelog, tape_size, expir_period, reorganisation, restart,  
ora<error number>, init.ora, init.sap, reconnect, ops$
```

EarlyWatch Service

EarlyWatch Service

The EarlyWatch service is a service for preventative system maintenance during all phases of the R/3 cycle. With this service, the status of your R/3 System is examined at regular intervals, and a forecast of expected resource requirements is created based on the current and past system data. This avoids performance problems that could arise due to bottlenecks in file storage and other system resources, and also increases the availability of your R/3 System. As well as this, the EarlyWatch service also performs other tasks during the different installation and operation phases of the R/3 System:

- Planning and test run for master data transfer
- Transfer of any transaction data
- Preparation for production start and load tests of the system
- Analysis of response times and tuning of system components after production start (this includes the database system)

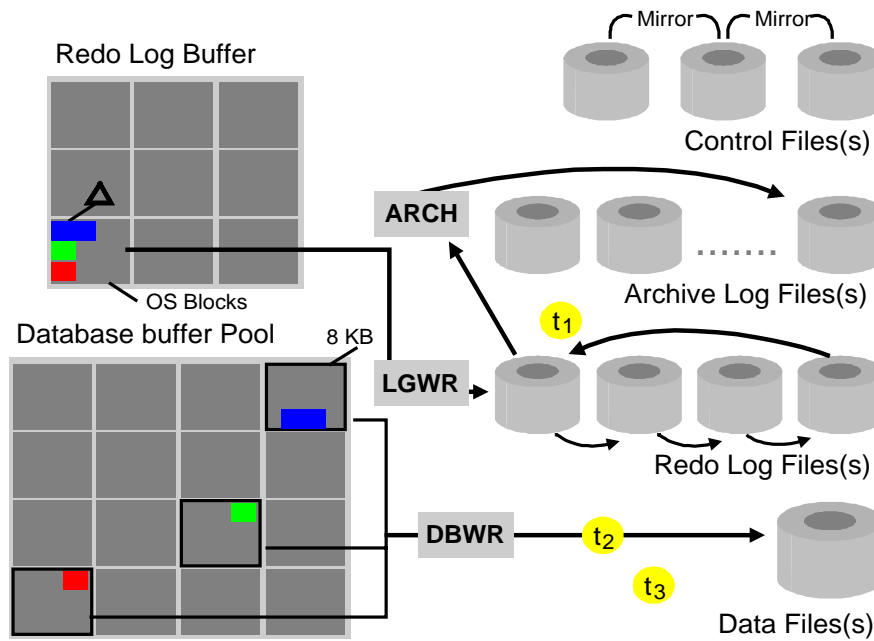
Setting Up the Database System

Prerequisites

1. Make sure that you have activated the archiving of the online redo log files (mode ARCHIVELOG) in all databases.
See [ARCHIVELOG Mode \[Page 33\]](#).
2. Make sure that the control files are mirrored on separate disks.
See [Mirroring the Control File \[Page 36\]](#).
3. Mirror the online redo log files on separate disks (at operating system level and/or using ORACLE resources). You can set this up, with Oracle support, when you install the database system.
See [Mirroring Online Redo Log Files \[Page 37\]](#).
4. Install online redo log files, offline redo logs files (online redo log files saved to the archive directory), and data files to separate disks.
See [Organizing Disk Storage \[Page 38\]](#).
5. Make sure that the block sizes (8K) of the database system and the operating system are the same. The AIX operating system is an exception, the block size of the database system must be 8K and the block size of the operating system must be 4K. Check the most important database parameters.
6. Familiarize yourself with the tablespace structure of the R/3 System. Monitor the critical tablespaces.
See [Tablespaces and SAP Naming Conventions \[Page 41\]](#).
7. Check the size of the database blocks (a block is the unit in which data transmission between main memory and disk storage takes place). For security and performance reasons, the blocks of the Oracle database and of the operating system must be the same size. For the R/3 System, the Oracle blocks have a size of 8K. The block size of the operating system is reformatted to 8K during the installation of the R/3 System. (The AIX operating system is an exception with a block size of 4K.)

Setting Up the Database System

Structure



For more information on the ORACLE Parallel Server (OPS configuration), see the appropriate Oracle documentation and the SAP documentation.

[Oracle Databases and Parallel Server Option \[Page 769\]](#)

SAPDBA: Archive mode

Use

If you work with an Oracle database with an OPS configuration (see [Oracle Databases and Parallel Server Option \(OPS\) \[Page 769\]](#)), you can use the *Select thread/instance* option to choose the parallel instance for which you want to perform one of the possible operations.

You can change both the database mode (*Toggle database log mode*) and the archiving mode (*Toggle automatic archival*).

Prerequisites

LOCAL INSTANCE: C11
 DATABASE LOG MODE: *archivelog*
 SELECTED INSTANCE: C11
 INSTANCE STATE: *opened*
 AUTOMATIC ARCHIVAL: *enabled*

a	<i>Toggle database log mode</i>
b	<i>Toggle automatic archival</i>
c	<i>Show all archive information</i>
d	<i>Show complete archive history</i>
e	<i>Select thread/instance</i>
f	<i>Refresh information</i>
q	<i>Return</i>

Features

Use the option *Show complete archive history* to obtain an overview of the redo entries and answer the following questions:

- Which online redo entries are affected?
- Which of the offline redo entries are still in the archiving directory on disk?
- Which of the files have already been archived to another volume?
- Which redo entries are needed to make the backup consistent?

For each database instance the following information is displayed:

- Last complete backup of the database (**LAST FULL BACKUP**) and the redo log files written during the backup (**HAS BEGUN AND ENDED AT LOG**: an interval of log sequence numbers is usually specified for an online backup).
- Log sequence number of the offline and online redo log files (the offline redo logs are numbered starting with **First_LOG**, the current online redo log file is **Curr_LOG**)
- Log sequence numbers of the offline redo log files that have already been archived (up to and including the file with the log sequence number **SAVED_LOGS**)

SAPDBA: Archive mode

If an error situation occurs in the database, making a recovery necessary, you should view this information before starting the recovery.

To display an overview of the status of the archiving process in your database, use the option *Show all archive information*. If the local and selected instance correspond (particularly in the case of a non-OPS configuration), the following information appears:

- LOCAL INSTANCE: Name of the local instance (e.g. C11)
LOCAL INSTANCE STATE: Status of the instance (for example, open, open for DBAs, mounted)
DATABASE LOG MODE: Database mode (archivelog/noarchivelog)
AUTOMATIC ARCHIVAL: Archiving mode (enabled/disabled)
- MAX LOGSIZE: Size of the individual online redo log files (SAP delivers the online redo log files with a size of 20 MB each).
- Archive directory details: LOG DIRECTORY (path), FREE SPACE (size of available freespace), LOGS ARCHIVED (number of offline redo log files).
- Details for online redo log files: ONLINE REDO LOGS (number of redo log files), ONLINE REDO GROUPS (redo log groups), OLDEST ONLINE LOG (log number of oldest online redo log file), NEXT TO ARCHIVE (log number of next online redo log file for archiving), CURRENT LOG (Number of the current online redo log file)
- LOG STATUS: O.K. or a warning (**danger of archiver stuck** or **archiver stuck**).

If the local and selected instance do not match, the following appears:

- Information on the local instance: LOCAL INSTANCE, LOCAL INSTANCE STATE, DATABASE LOG MODE
- Information on the selected parallel instance: INSTANCE (THREAD), AUTOMATIC ARCHIVAL, NEXT TO ARCHIVE, CURRENT LOG, LOG STATUS

ARCHIVELOG Mode

Use

After installation, the database runs in *ARCHIVELOG* mode. This configuration ensures that all written online redo log files are automatically archived after a redo log switch (using the background process ARCH). Only when the contents of the online redo log file have been saved successfully (as an offline redo log file in the archive directory), can the file be written to again.

Prerequisites

You must use *ARCHIVELOG* mode if you want to make backups online. If your database - for whatever reason - is in *NOARCHIVELOG* mode, you must restart the database and start *ARCHIVELOG* mode explicitly.

Activities

The database profiles provided with your R/3 System are configured to start the Oracle archiving process (ARCH) automatically.

See [Archiving Parameters \[Page 34\]](#)

You can configure *ARCHIVELOG* mode in the SAPDBA menu. To do this, select *Archive mode* → *Toggle database log mode*. SAPDBA shuts down the database, configures the database mode, and restarts the database.

See [SAPDBA: Archive Mode \[Page 31\]](#)

You can also set *ARCHIVELOG* mode with the Oracle program SVRMGR:

```
SVRMGR> connect internal
SVRMGR> startup mount
SVRMGR> alter database archivelog;
SVRMGR> alter database open;
SVRMGR> archive log list
```

When **archivelog** appears in the **database log mode** display, this means you have correctly activated the *ARCHIVELOG* mode. Use the SAP utility program BRBACKUP to perform an offline backup of the database.

See [Backing Up the Database with SAPDBA \[Page 202\]](#)

Archiving Parameters

Archiving Parameters

The following parameters in the `init.ora` file control the archiving process:

Parameter	Note
<code>log_archive_start = true</code>	Preset by SAP
<code>log_archive_dest = <directory>/<file prefix></code>	Part of the file name
<code>log_archive_format = ORACLE default</code>	Use the ORACLE default

The parameter `log_archive_start = true` causes the archive process to be started automatically when the database is started. The **automatic archival** mode is set to enabled.

The parameter `log_archive_dest` defines the archive directory of the online redo log files for archiving. It is delivered with the specification of the following SAP standard path (example for a single instance installation):

```
BS> <ORACLE_HOME>/saparch/<ORACLE_SID>arch
```

See [Environment Variables \[Page 45\]](#)

Prerequisites

- The corresponding archive directory (under UNIX: `saparch`) must exist.
- Make sure that the directory is not write-protected.
- Make sure that the directory has enough free space. Otherwise, the archiving process cannot archive any log files and no further actions are possible on the database (Archiver Stuck).

The Oracle database interprets the string `<ORACLE_SID>arch` as the prefix of a file name that, together with an appended sequential number (log sequence number), names the offline redo log files.

You can archive these offline redo log files (online redo log files saved in the archive directory) with the SAP utility BRARCHIVE.

See [Archiving the Offline Redo Log Files \[Page 210\]](#).

Configuration with SAPDBA

When automatic archiving of the online redo log files is not active, you can activate it using the SAPDBA menu. To do this, choose *Archive Mode* → *Toggle automatic archival*.

See [SAPDBA: Archive Mode \[Page 31\]](#).

Configuration with SVRMGR

After starting up an instance, you can also use the Oracle program SQLDBA to start and stop the archiving process.

Command for starting the ARCH process:

```
SVRMGR> connect internal
```

```
SVRMGR> alter system archive log start;
```

```
SVRMGR> archive log list
```

Archiving Parameters

When enabled appears in the **automatic archival** display, this means you have correctly activated the automatic archiving.

Mirroring the Control File

Mirroring the Control File

Use

If you only have one copy of the control file and lose this copy due to a disk error or other problems, it will likely be impossible to completely recover the database. This means data loss will be inevitable. To protect your database against the loss of the control file, create multiple copies. When the R/3 System is installed, the control file is mirrored to at least two additional disks (often three). You can also mirror the control file yourself; always make sure that all the control files are on different disks.

Prerequisites

The default database profile (**init<DBSID>.ora** profile) delivered with the system makes sure that the control file and its mirror copies are stored in directories that are mounted on different disks.

Procedure

You can change the standard mirroring of the control file by setting the **control_files** parameter.

Sample entry in profile **init<DBSID>.ora**:

```
control_files = ( ?/dbs/cntrl<SAPSID>.dbf ,  
                ?/sapdata1/cntrl/ctrl<SAPSID>.dbf , ?/sapdata2/cntrl/ctrl<SAPSID>.dbf )
```

The question mark ? is the official Oracle placeholder for the home directory of the database system (for example, **/oracle/C11**).

Mirroring Online Redo Log Files

Use

If you lose one or more online redo log files, you can no longer recover the database changes recorded in them. This means that you can only recover the database up to the first gap in the online redo log records. For this reason, SAP highly recommends that you mirror the online redo log files (at least once).

If an online redo log file of a group is lost, the database remains in operation. Oracle then uses the remaining member (or members) of this group to log the database changes. In such a case, you must recover the original mirroring of the online redo log files as quickly as possible.

[Recovery with SVRMGR: Overview \[Page 727\]](#)

[Recovery: One Inactive Redo Log Missing \[Page 744\]](#)

Prerequisites

When an R/3 System is installed using Oracle resources, an online redo log group normally consists of the original online redo log file and a mirror copy of this file (two members).

Procedure

You can check the mirroring with SAPDBA.

To do this, select *Archive mode* → *Show all archive information: ONLINE REDO LOGS*

See [SAPDBA: Archive Mode \[Page 31\]](#)

During SAPDBA startup, any missing online redo log files will be displayed if the profile parameters **check_initial_settings** or **check_redolog_mirror** are configured properly.

See [check_initial_settings \[Page 267\]](#)

When the database is open, you can also check the mirroring with the ORACLE program SVRMGR.

```
SVRMGR> connect internal
```

```
SVRMGR> select * from V$logfile;
```

You can also make additional mirror copies. For more information, see the appropriate Oracle documentation. In addition, many systems support hardware-based file mirroring. For more details, see your operating system documentation.

For more information on the security of offline redo log files, see [Archiving the Offline Redo Log Files \[Page 210\]](#).

Organizing Disk Storage

Organizing Disk Storage

SAP recommends that you store the files of the database system on different physical disks. In this example, the control file is mirrored twice:



Disk no.:	Directory	Contents of directory
Disk 1:	origlogA	Online redo log files from the first and third group (Set A)
Disk 2:	origlogB	Online redo log files from the second and fourth group (Set B)
Disk 3:	mirrorlogA	Mirrored online redo log files from the first and third group (Set A)
Disk 4:	mirrorlogB	Mirrored online redo log files from the second and fourth group (Set B)
Disk 5:	sapdata1	Database files, mirror of the control file
Disk 6:	sapdata2	Database files, mirror of the control file
Disk 7:	sapdata<n>	Other database files in sapdata3 up to sapdata<n>, each on separate disks when possible

- Make sure that the number and descriptions of the control file and its mirrors agree with the entry in the `init.ora` profile. The database files can be distributed among any number of disks.
- Make sure that the offline redo log files (online redo log files saved to the archive directory) are not kept on the same disk as the online redo log files. (The offline redo log files have not been taken into account in the above example.)

A disk assignment like the one described above is essential for reliable database operations. For performance reasons, distribute the online redo log groups to four disks.

Database Parameters

The R/3 System comes with a standard initialization profile for the database parameters:

```
OS> <ORACLE_HOME>/dbs/init<DBSID>.ora
```

for example /oracle/C11/dbs/initC11.ora

This profile contains the default parameter settings recommended for the R/3 System. Copy this standard profile so that you can access the original parameters, if necessary.

Operating System-Specific Parameters

All parameters that specify paths and file names are operating system-specific. These parameters have to be adjusted to agree with your operating system. When possible, however, SAP recommends that you retain the default path and file descriptions in the system. Making the appropriate changes to your operating system instead.

? is the ORACLE placeholder for \$ORACLE_HOME.

Parameter	SAP recommendation
control_files	See Mirroring the Control File [Page 36] .
log_archive_dest	OS> ?/saparch/<DBSID>arch, where ?/saparch can be a soft link to a separate disk or a separate mount point. See Archiving Parameters [Page 34] .
background_dump_dest	OS> ?/saptrace/background Directory for database trace and database ALERT files. If you have serious database problems, you must analyze the information in the ALERT and trace files.
user_dump_dest	OS> ?/saptrace/background Directory for user trace files.

Tuning Parameters

The following lists a selection of RAM-dependent tuning parameters. For more information, see the ORACLE documentation.

Parameter	Settings
ram (unit: MB) Portion of physical RAM that you want to reserve the system.	One system installed on one host: ram = 1.0*RAM Two systems installed on one host: ram = 0.5*RAM

Database Parameters

<code>rating</code> Proportion of ram taken up by the database and the R/3 System	Default: <code>rating=0.25</code> (25% of RAM reserved for the database, 75% for the R/3 System) Database-only server: <code>rating=0.8</code> (80% of RAM is reserved for the database) If <code>ram < RAM*0.8</code> : <code>rating=1.0</code> (100% of RAM is reserved for the database)
<code>db_block_size</code> <code>db_block_size</code> (unit: bytes) size of the Oracle database blocks	Operating system-specific, default: 8192

The parameters `ram`, `rating` and `db_block_size` are taken into account when calculating the following tuning parameters:

Parameter	Determination of its value
<code>shared_pool_size</code> <code>db_block_size</code> (unit: bytes) buffer size for parsed SQL statements in the Oracle SGA (System Global Area)	$shared_pool_size = ram * rating * 0.4 * 1024 * 1024$ The factor 0.4 means that <code>shared_pool_size</code> occupies 40% of the calculated SGA. $shared_pool_size = 20.971.520$ if the value calculated using the formula above is lower than 20971520 Bytes.
<code>db_block_buffers</code> <code>db_block_size</code> (unit: blocks) buffer size for data blocks and index blocks in the Oracle SGA	$db_block_buffers = ram * rating * 0.6 * 1024 * 1024 / db_block_size$ The factor 0.6 means that <code>db_block_buffers</code> occupies 60% of the calculated SGA.
<code>log_buffer</code> <code>db_block_size</code> (unit: bytes) buffer size for redo log entries in the Oracle SGA	Default: <code>log_buffer=327.680</code> Part of the SGA, which is very small when compared with <code>shared_pool_size</code> and <code>db_block_buffers</code> .
<code>row_cache_cursors</code> Number of cursors used for selecting lines from the ABAP Dictionary.	$row_cache_cursors =$ 300, if <code>ram >= 256</code> 200, if <code>ram < 256</code> 100, if <code>ram < 128</code>

Tablespaces and SAP Naming Conventions

Use

You can use SAPDBA to create new tablespaces in accordance with the SAP naming conventions, for example during the repository switch in a system upgrade. For administrative reasons, you may have to move a table to a separate tablespace. To help orientation, this tablespace should conform to the SAP naming convention.

See [Creating a Tablespace \[Page 694\]](#).

Prerequisites

The following syntax is used for naming tablespaces (TSP):

```

PSAP<name>D   for data tablespaces
PSAP<name>I   for index tablespaces
PSAP<TSP>I   (if it is not important to differentiate)
    
```

The following table contains an overview of all the SAP tablespaces and how they are used. The sizes are not absolute values since they can change according to how much the tablespace is used. They are there to help you compare the tablespaces.

Tablespace name	Use	Approximate size in MB
Oracle tablespaces	These tablespaces are required for operation of the Oracle RDBMS, and contain no SAP data.	
SYSTEM	Oracle DDIC	150
PSAPROLL	Rollback segments	300
PSAPTEMP	Sort processes	64
Basis tablespaces		
PSAPEL300D/I	Development environment loads	309/11
PSAPES300D/I	Development environment sources	1573/503
PSAPLOADD/I	Screen and report loads (ABAP)	65/37
PSAPSOURCED/I	Screen and report sources (ABAP)	58/14
PSAPDDICD/I	ABAP Dictionary	46/26
PSAPPROTD/I	Log-like tables (such as spool)	97/13
Application		
PSAPCLUD/I	Cluster tables	110/14
PSAPPOOLD/I	Pooled tables (such as ATAB)	392/254
PSAPSTABD/I	Master data, transparent tables	496/318
PSAPBTABD/I	Transaction data, transparent tables	209/191
PSAPDOCUD/I	Doc., Sapscript, Sapfind	70/13
Customers		
PSAPUSER1D/I	Customer tables	5/5

Activities

Monitor some tablespaces with special care, for the following reasons:

Tablespaces and SAP Naming Conventions

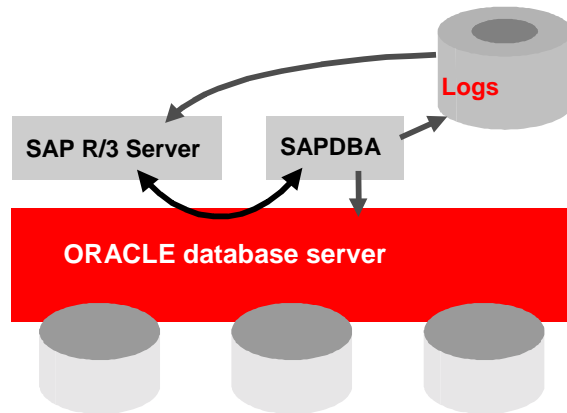
1. Some tablespaces have memory management problems (extent overflow, tablespace overflow, freespace problems) more often than others. Monitor the following tablespaces carefully, particularly during the transfer of data when you install your R/3 System:
 - PSAPSTABD/I
 - PSAPCLUD/I
 - PSAPBTABD/I
 - PSAPPOOLD/I
 - PSAPPROTD/I
2. Some tablespaces contain a large number of tables, which means that the number of extents for each tablespace is very high. You may experience problems with Oracle when you [reorganize special tablespaces \[Page 587\]](#).
 - PSAPSTABD/I
 - PSAPBTABD/I
 - PSAPPOOLD/I
3. The SAP software compresses a large portion of its tables in order to save disk space. It is, therefore, not useful to compress the tablespaces that contain these tables when you export data. SAP's compressed tables are held primarily in the following tablespaces:
 - PSAPCLUD/I
 - PSAPDOCUD/I
 - PSAPPOOLD/I
 - PSAPSOURCED/I or PSAPES300D/I
 - PSAPLOADD/I or PSAPEL300D/I

Using SAPDBA

To simplify the administration of the ORACLE database system, SAP provides the SAPDBA tool.

SAPDBA provides database management functions that you mostly cannot perform from within the R/3 System. SAPDBA runs outside the R/3 System, on the operating system of the database host.

If you administer your ORACLE database with the SAPDBA utility, all actions are logged. This means that you can check at any time which operations have been performed.



[Configuring SAPDBA \[Page 44\]](#)

[Starting SAPDBA \[Page 65\]](#)

[SAPDBA: Initial Menu \[Page 86\]](#)

[SAPDBA Command Mode \[Page 69\]](#)

[Displaying and Managing Logs and Profiles \[Page 87\]](#)

Configuring SAPDBA

Configuring SAPDBA

You can configure SAPDBA by setting the environment variables in a corresponding fashion, changing the default initialization profile or defining a new one.

Name of the default initialization profile:

```
OS> <ORACLE_HOME>/dbs/init<DBSID>.dba
```

An overview of profile parameters and their default values and uses is listed under [Initialization Profile init<DBSID>.dba \[Page 263\]](#) **Error! No bookmark name given..** Changes to profile parameters become active when you start SAPDBA.

[Environment Variables \[Page 45\]](#)

[Directory Structure \(Unix\) \[Page 47\]](#)

[Configuring the Display Length \[Page 48\]](#)

[Configuring the UNIX Command at \[Page 49\]](#)

[Setting the Log Checks Option \[Page 50\]](#)

[SAPDBA: Expert Mode \[Page 60\]](#)

[Users and Roles \[Page 51\]](#)

[Restricted Session \[Page 63\]](#)

Environment Variables (UNIX)

- **ORACLE_SID**: System ID of the database instance

Example: C11

SAPSID or **sapsid** refers to the name of the R/3 System (SAP System ID).

DBSID or **dbsid** refers to the name of the database instance (database instance system ID).

When a single instance is installed, **SAPSID** and **DBSID** are the same.

- **ORACLE_HOME**: HOME directory of the Oracle software.

Standard: /oracle/<DBSID>

- **SAPDATA_HOME**: Directory of the database files.

Standard: /oracle/<DBSID>



The variables **ORACLE_SID**, **ORACLE_HOME** and **SAPDATA_HOME** must always be set. There is no default.

The following environment variables must only be set if the corresponding paths deviate from the defaults specified here:

- **SAPARCH**: Directory for the BRARCHIVE logs.

Default value: \$SAPDATA_HOME/saparch

- **SAPBACKUP**: Directory for the BRBACKUP logs.

Default value: \$SAPDATA_HOME/sapbackup

- **SAPCHECK**: Directory for the sapdba -check/analyze logs.

Default value: \$SAPDATA_HOME/sapcheck

- **SAPREORG**: Directory for all other SAPDBA logs, as well as shell and SQL scripts. It is also the standard directory for export and unload dump files, if the parameter `exireo_dumpdir` in the profile `init<DBSID>.dba` is not set.

Default value: \$SAPDATA_HOME/sapreorg

- **SAPTRACE**: Directory for Oracle trace files and the alert file.

Default value: \$SAPDATA_HOME/saptrace

- **SAPDATA1**: Directory of the DB data files.

Default value: \$SAPDATA_HOME/sapdata1

(Analog for **SAPDATA<n>**, $n=1, \dots, 99$).

The environment variables **SAPDATA<n>** must only be defined if directories are used that deviate from the default.

Configuring SAPDBA

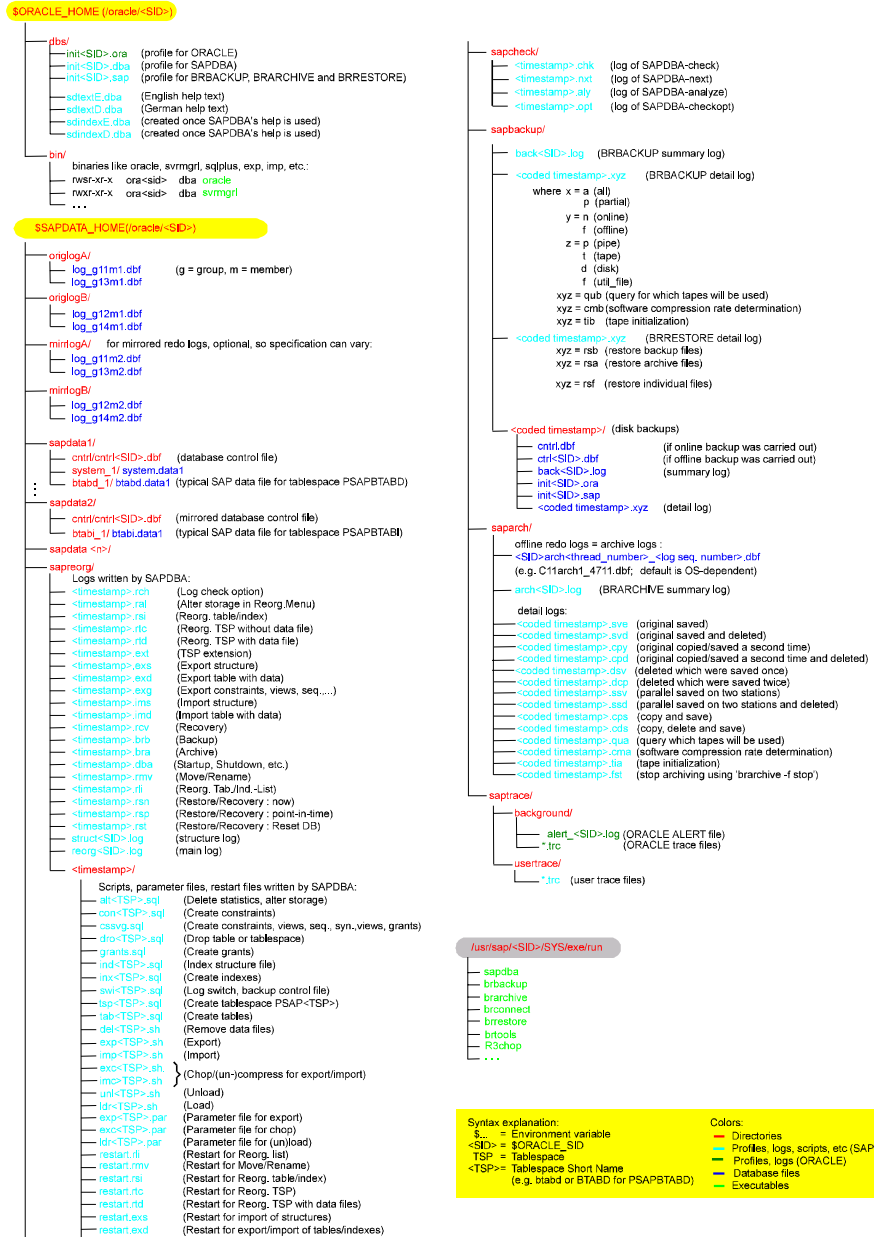
- **TWO_TASK**: Identification of a remote database system.
This environment variable must be left unset.

Other environment variables that can be set:

- **LINES**: Definition of the screen height.
Recommended height: ≥ 24 lines.
- **COLUMNS**: Definition of the screen width.
Recommended width: ≥ 80 columns
- **SAPDBA_DEBUG**: Setting the trace function for error analysis.
See Note 29324 for setting options.

Directory Structure (Unix)

Most important directories for SAP (>=4.0) & ORACLE



Syntax explanation:

- \$ = Environment variable
- <SID> = \$ORACLE_SID
- TSP = Tablespace
- <TSP> = Tablespace Short Name (e.g. btabd or BTABD for PSAPBTABD)

Colors:

- Directories
- Profiles, logs (ORACLE)
- Database files
- Executables

Configuring the Display Length

Configuring the Display Length

SAPDBA uses 24 lines and 80 columns for the screen display. You can, however, lengthen or shorten the display if required by your screen. If, for example, you choose to use fewer than 24 lines for the screen display then some of the SAPDBA will no longer be displayed completely. However you can display lists correctly.

To select a different screen display size, set the following environment variable (operating system specific) before you start SAPDBA:

```
LINES <Number of lines in the screen display>
```

```
COLUMNS <Number of columns in the screen display>
```

Also make sure that you set the variable `TERM` correctly for your terminal type.

Configuring the UNIX Command at

SAPDBA uses `/usr/bin/at` to schedule a reorganization

```
r-sr-xr-x root root at
```

To use `at`, you must make an entry in the file `/usr/lib/cron/at.allow`. Add `ora<sapsid>` to the list of authorized users.

The running `at` process has root authorization, and analyzes the jobs created by SAPDBA in file `/usr/spool/cron/atjobs`. To list all `at` jobs, enter the command `at -l`. These jobs are transparent files that you can display using UNIX commands such as `vi`, `view`, `more`, and so on.

Make sure the proper entries have been made, and then test your configuration (for example, by reorganizing a small table or tablespace such as `PSAPUSER1D`: See [Carrying out a Reorganization \[Page 614\]](#)). This will help you avoid processes crashing due to incorrect configuration of the `at` command.

Setting the Log Checks Option

Setting the Log Checks Option

Use

You can make SAPDBA log its database checks and analyses.

Procedure

Depending on your context, choose:

Tablespace administration → *Log checks*

[SAPDBA: Tablespace Administration \[Page 681\]](#)

Reorganization → *Check extents and fragmentation* → *Log checks*

[SAPDBA: Reorganization \[Page 602\]](#)

Leave the *Log checks* option set to *no* if you want to check the database, and the results do not have to be available at a later time. The displayed data is not recorded in the SAPDBA log files, to avoid unnecessary data. Select *yes* if you want to record the checks in the respective log files.



This log option is not global. It is only active in the menu in which it is set to yes.

Users and Roles

Operating System Users

In SAP R/3 Release 4.0 the roles of the users `ora<sid>` and `<sid>adm` on Unix, or `<SID>ADM` and `SAPSERVICE<SID>` on Windows NT, were separate. Only the user `ora<sid>`, or `<SID>ADM` on NT, has unrestricted authorization for DBA operations. This user belongs to the two operating system groups `oper` and `dba` (Unix), or `ORA<SID>DBA` and `ORA<SID>OPER` (Windows NT).

In contrast the authorizations of the operating system user `<sid>adm` (Unix) or `SAPSERVICE<SID>` (Windows NT) are restricted to operator activities, such as starting/shutting down the database, performing database backups and database checks. This user only belongs to the operating system group `oper` (Unix), or `ORA<SID>OPER` (Windows NT).

Database Roles

- **SYSDBA**
All authorizations
- **SYSOPER**
Operator activities, but *no* read or write authorizations.
- **SAPDBA**
Read and write authorizations in order to work with SAPDBA and BRBACKUP command options (and therefore CCMS DBA functions)

In order to be able to use the CCMS DBA functions or SAPDBA and BRBACKUP command options without restrictions the OPS\$ user must have both the SYSOPER role and the SAPDBA role.

Overview: Operating system users and groups, database users and roles

UNIX:

OS users	OS group	DB role	OS users
<code>ora<sid></code>	<code>dba</code> <code>oper</code>	SYSDBA SYSOPER	<code>INTERNAL (SYS)</code>
<code><sid>adm</code>	<code>oper</code>	SYSOPER SAPDBA	
			<code>OPS\$<SID>ADM</code>

Windows NT

OS users	OS group	DB role	OS users
<code><SID>ADM</code>	<code>ORA_<SID>_DBA</code> <code>ORA_<SID>_OPER</code>	SYSDBA SYSOPER	<code>INTERNAL (SYS)</code>
<code>SAPSERVIE<SID></code>	<code>ORA<SID>OPER</code>	SYSOPER SAPDBA	
			<code>OPS\$SAPSERVICE<SID></code>

Users and Roles



The OS group on Windows NT can also be specified globally (without instance name) (`ORA_DBA`, `ORA_OPER`).

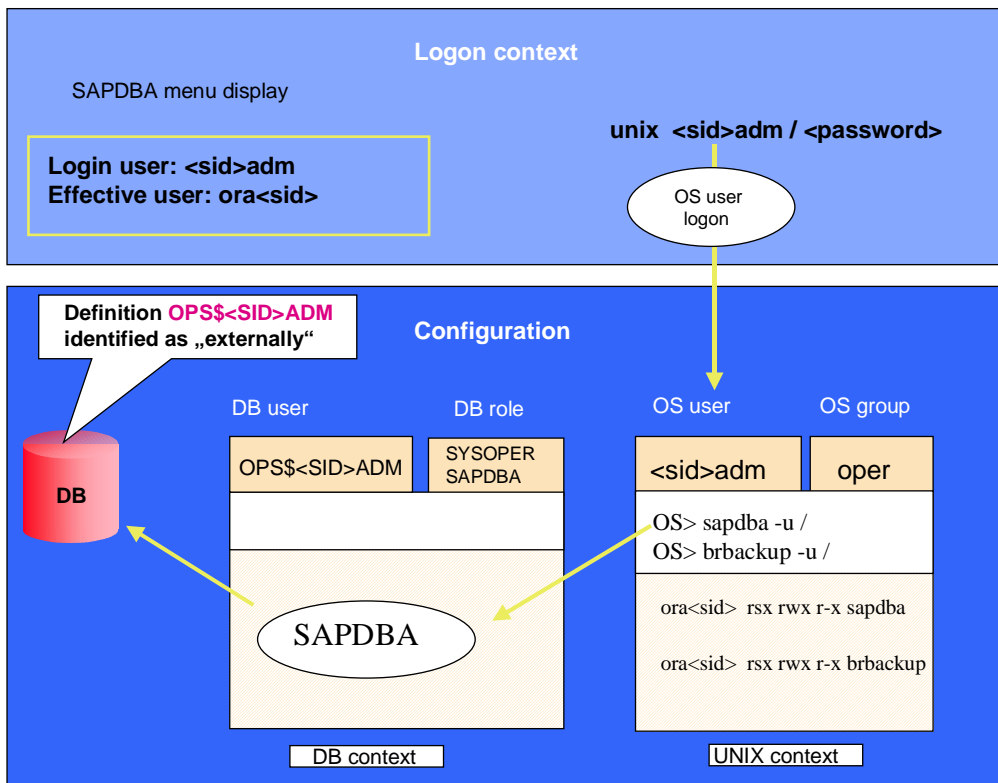
OPSS\$ Database User

The ORACLE OPSS\$ mechanism moves the entire DB security mechanism to the operating system level.

The prerequisite is that a DB user `OPSS$<OS_user>` corresponding to the OS user is defined on the database, and identified as externally.

Once you have logged on successfully with the OS user, you can connect to the database with `SVRMGR> connect /`, without having to enter another password. You are then working as `OPSS$<OS_user>`. In the same way you can start the program SAPDBA with `os> sapdba -u /`.

This **OPSS\$** mechanism is always used if you call SAPDBA or BRBACKUP from the CCMS menu in the R/3 System.



The OPSS\$ Mechanism (Unix)

SAPDBA Database User

The standard DB user is always SYSTEM. SYSTEM connects with the ORACLE option `AS SYSOPER` or `AS SYSDBA` for actions such as startup, shutdown, recover and so on, as well as selecting from V\$ tables when the database is not open.

Database Users

Database Users

You can display information about the current operating system and database users using the menu option *User and Security*.

As a security warning, the system informs you if the database user passwords are still set to the default values.

You can also display the roles of the default users `SYS`, `SYSTEM`, `OP$ORAC11` and `SAPR3`. The users `SAPR3` and `OP$ORAC11` must not have a DBA role while `SYS` and `SYSTEM` must have the DBA role.

Changing the Password of the Database User

Use the following SAPDBA command to change and encrypt the password of the database user `SAPR3`.

```
sapdba -sapr3 <sapr3_password>
```

The official SAP encryption routine is used to encrypt the password.

The option `-sapr3 <sapr3_password>` does the following:

1. The `SAPR3` password is changed as specified. If the database user `SAPR3` does not yet exist, it is created with this password in the database.
2. The new `SAPR3` password is entered in the table `OP$<SID>ADM.SAPUSER`.
 - If the previous entry in this table was not encrypted, then the new password is also entered without being encrypted.
 - If the previous entry was encrypted, then the new password is also encrypted when it is entered.
 - If the previous entry was both encrypted and not encrypted, then the new password is also entered in this way.
 - If no password exists, then the new password is encrypted when it is entered.

You can also change the password of the database user `SAPR3` or other database users interactively in SAPDBA. If you change the passwords of database users other than `SAPR3`, then the new password is not entered in the table `OP$<SID>ADM.SAPUSER`. To change the password of a database user interactively, proceed as follows:

1. Start SAPDBA using the command

```
sapdba [-u system/<system_password>]
```
2. In the main menu, choose `m(User and security) → p(Change password)`.
 - `a(Password entering mode: invisible/visible)`
 - `b(User [default user is SAPR3])`
 - `c(Change password)`

When you change the password interactively, you can enter the new password hidden, as long as it is no longer than eight characters (default setting).

UNIX User Configurations

UNIX User Configurations

Various user configurations at operating system level allow different security requirements to be met.

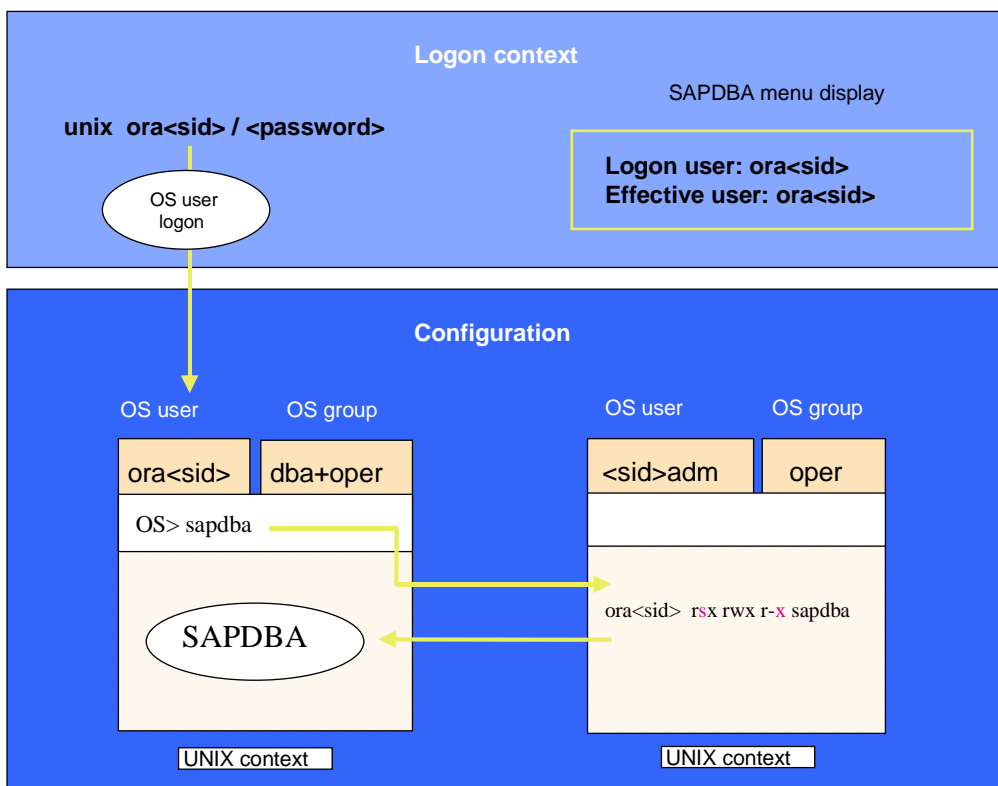
Configuration 1

Database administrator with all authorizations

This configuration matches the SAP standard installation.

One person is fully responsible for database administration using SAPDBA and other DB tools. As Unix user `ora<sid>`, this person can also perform all actions possible in this context.

In this case, there are no other security aspects that need to be taken into account and the following user configuration is appropriate:



User configuration 1

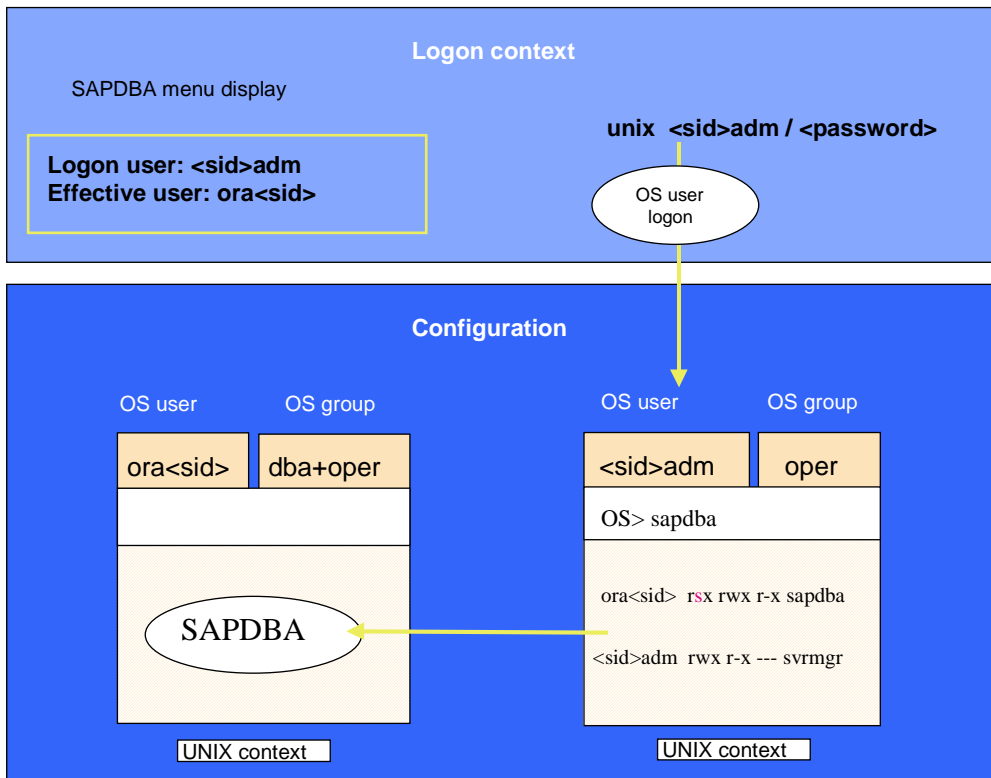
The database administrator knows the Unix password of the OS user `ora<sid>`. Logged on as such, the administrator belongs to the OS groups `dba` and `oper` and has very high privileges. `ora<sid>` can access the database directly and manipulate database objects. The administrator can also start the program SAPDBA at the operating system level.

Configuration 2

User with operator privileges

UNIX User Configurations

This operator is authorized to make database backups, and to call SAPDBA with certain command options, such as -analyze, -check, -checkopt. This operator can also start up and shut down the database, but only has limited authorization for reading or modifying data (that is, just data that is needed for the programs SAPDBA, BRBACKUP and BRARCHIVE, no application data). The administrator is the only user allowed to restore backups.



User configuration 2

The program SAPDBA belongs to ora<sid> but can be called by any user. SAPDBA runs with the authorization of the user ora<sid> due to the set s bit.

The operator logs on as user <sid>adm. This user belongs to the OS group oper. The user is authorized to start up and shut down the database.

The user <sid>adm has a corresponding OPS\$ user on the database (OPS\$<sid>adm) as standard. This OPS\$ user has the SAPDBA role on the database (granted). This allows the user to read the Oracle Dictionary tables and write to the SAPDBA log tables on the database.

The OPS\$ mechanism is activated automatically for the standard user <sid>adm during installation and upgrade. You can use the OPS\$ mechanism by calling SAPDBA with the option -u /.

Examples: **sapdba -u / -check**
brbackup u / -q

UNIX User Configurations



As user **<sid>adm**, the operator has complete administration authorization for the SAP System (not for the database). If this is not required, then a separate OS user with the aforementioned operator authorizations has to be set up. (See user configuration 3.)

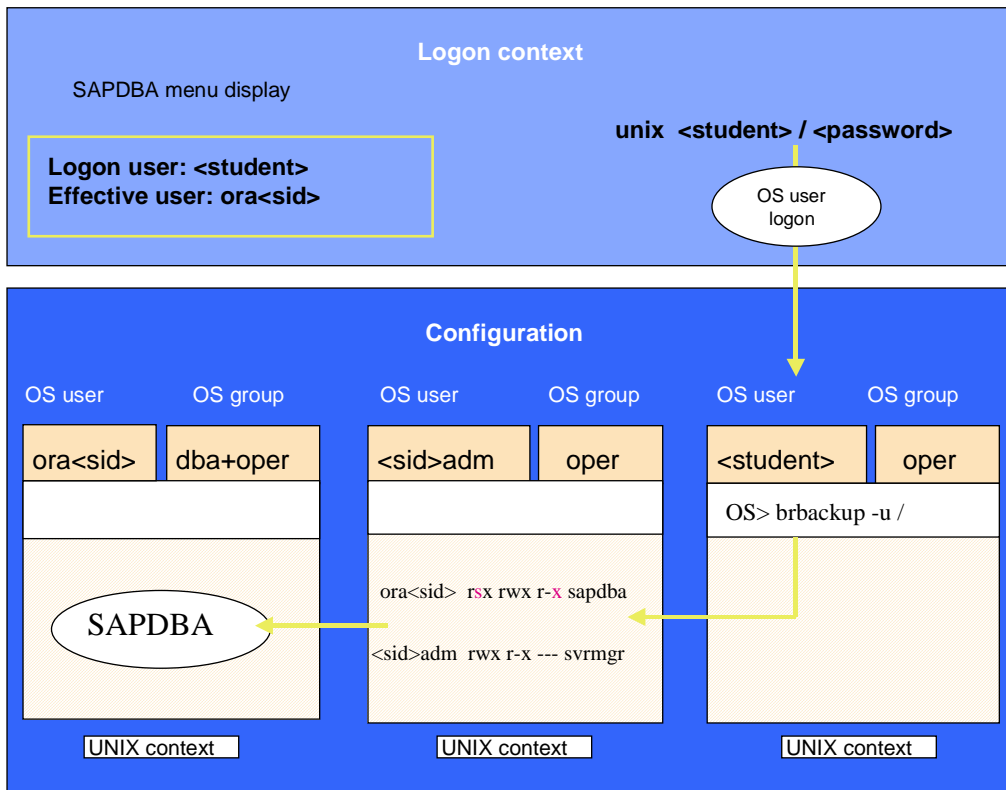


If the standard password has been changed from **system**, and the OPSS mechanism is not used, then SAPDBA, BRBACKUP and so on, have to be called with the option **-u**.

Configuration 3

Arbitrary user <student> who can perform selected operations only.

You need a security mechanism that allows an SAPDBA user to perform particular actions only (for example, monitoring) and does not provide any other privileges. This user must not know the ora<sid> or <sid>adm password, nor belong to the OS group **dba**. Depending on the actions needed, this user should be assigned to the OS group **oper** (required for DB backups, for example).



User configuration 3

The program SAPDBA belongs to ora<sid> but can be called by any user <student>. SAPDBA runs with the authorization of the user **ora<sid>** due to the set s bit.

UNIX User Configurations

The password of user **system**, which is used by SAPDBA to connect to the database, must in this case be stored in a password file in the directory **<orapwd_path>**. SAPDBA always requests this file when it is called by any user. The **system** password protection prevents an arbitrary user from logging on to the database with `SVRMGR> connect system/password`

SAPDBA supports the call **sapdba -u system/<password>** and the OPS\$ mechanism (**sapdba -u /**) in this configuration for the first time in Release 4.5A.

You can start BRBACKUP with **brbackup -u /** in releases before 4.5A. This means that you can work with the user **OPS\$<student>** to make backups. The prerequisite is that you have saved this user on the database, and assigned it the SAPDBA role (**granted**).

The expert mode is used to restrict the actions that the user **<student>** can perform.

For more information, refer to [SAPDBA: Expert mode \[Page 60\]](#).

SAPDBA: Expert Mode

SAPDBA: Expert Mode

You can set a so-called expert mode for the essential functions of SAPDBA. In this way, only users with the appropriate privileges (knowledge of the password for activating expert mode) can execute critical functions. Expert mode is set for the required function with the menu option **Expert mode**. If no changes were made, SAPDBA only requires that expert mode is activated when the function *Restore/Recovery*. You can also protect a number of other functions with expert mode. See “*Allocating Expert Mode*”.



This procedure depends on the operating system (for example it is not supported at all for Windows NT).

EXPERT MODE: OFF/ON

PRIVILEGES: ALL SAVED (NOTHING ALTERED)/<n> CHANGED but not SAVED

a	- Switch expert mode
(b)	- Set initial password for expert
c	- Change password for expert
d	- Show/Alter privileges for main options
e	- Save current privileges

If SAPDBA reports that expert mode is not active when calling a function, select the option *Expert mode*. This menu (*Expert's Password*) enables you to activate the expert mode once you enter an appropriate password.

- The password is known.
You can activate expert mode by selecting *Switch expert mode* and entering the password. The entry *EXPERT MODE: ON* appears in the *Expert's Password* menu after successful operation. You can now use the SAPDBA functions for which the expert mode is required.
- The password is not known.
See [Defining the Password \[Page 62\]](#).

Allocating Expert Mode

Using the menu option *Show/Alter privileges for main options* you can see which menu options were protected with expert mode. If you activated expert mode (*EXPERT MODE: ON*), you can allocate further functions to this security mechanism or remove them from it:

	SAPDBA menu option which is locked
START/STOP	<i>Startup/Shutdown instance</i>
TSP_ADMIN	<i>Tablespace administration → Alter tablespace Add Datafile</i> <i>Tablespace administration → Create tablespace</i>

REORG	<i>Reorganization → Reorganize single table or index</i> <i>Reorganization → Reorganize list of tables and indexes</i> <i>Reorganization → Reorganize tablespace</i> <i>Reorganization → Reorganize tablespace and data files</i> <i>Reorganization → Move/rename data files of a tablespace</i>
EXP/IMP	<i>Export/import</i>
ARCHIVE_MODE	<i>Archive mode → Toggle database log mode</i> <i>Archive mode → Toggle automatic archival</i>
EXEC_SQL_SCRIPT	<i>Additional functions → Execute SQL scripts</i>
BACKUP	<i>Backup database</i> <i>Backup archive logs</i>
RESTORE	<i>Restore/Recovery</i>
CLEANUP	<i>Show/Cleanup → Cleanup log files/directories</i>
ALTER	<i>Reorganization → Alter/show table or index storage parameters</i>

With the *PRIVILEGES: display* you can see how many functions you made changes to in expert mode within the current session. You can save these changes with the menu option *Save current privileges*. SAPDBA logs the functions which are protected by the expert mode in the database table SAPDBAPRIV.



If table SAPDBAPRIV does not exist, SAPDBA creates it. The owner of the table is `sapr3`. You can allocate SAPDBAPRIV to user `system` using the corresponding ORACLE commands if you want to protect the table against unauthorized access.

If the database is closed, SAPDBA cannot access SAPDBAPRIV and cannot store any changes in the expert mode. In this case, SAPDBA uses internal default values, that is only the functions *Check (and repair) database* and *Restore/Recovery* are protected by the expert mode.

Defining the Password

Defining the Password

Select *Expert mode* to define the password for expert mode. Several of the steps necessary for this can usually only be performed by a user with `root` authorization (the UNIX administrator often has this authorization) and should be selected immediately after installation. SAPDBA will report when other users are authorized to perform this operation (platform-specific).

Creating the password

If you are setting the password for the first time, proceed as follows:

- In the *Expert's Password* menu, select the function *Set initial password for expert* and enter the desired password twice in succession.
- SAPDBA stores the password in encoded form in directory `passwd`, file `passwd.dba`. If the directory (for example `/oracle/C11/passwd`) already exists, you will have to delete it first.

Changing the Password

To change the password, proceed as follows:

- Open a UNIX window parallel to the SAPDBA window. Log on as user `root`. Change to the ORACLE home directory (e.g. `/oracle/C11/`).
- Change to directory `passwd` and assign file `passwd.dba` to owner `ora<sapsid>` (e.g. `oracl1`).
- In the *Expert's Password* menu, select the function *Change password for expert*. Enter the old password and then the new one twice in succession.

If you have forgotten the old password, delete directory `passwd` and create a new one as described above (you must be logged on as user `root`).

SAPDBA stores the password in encoded form in directory `passwd`, file `passwd.dba`. SAPDBA assigns the directory to user `root`.



On some operating system platforms, database user `ora<sapsid>` is not able to assign directories or files to user `root`. As a result, the password is not protected against unauthorized deletion. In such cases, SAPDBA issues a warning message with instructions for the administrator to change the owner and/or the access privileges.

Restricted Session

You have the option of operating the database in RESTRICTED-SESSION mode. This means that only users with a specific access privilege can log onto the database (usually the database administrators).

See [Starting up the Database System \[Page 100\]](#).

Use the SAPDBA function *User and Security* → *Check RESTRICTED MODE* to check which mode you are currently using.

When you exit SAPDBA, the program checks the database mode. If you have been working in RESTRICTED-SESSION mode, SAPDBA asks if you want to change the mode to NORMAL, to enable all users to log onto the database.

SAPDBA and Autoextend

SAPDBA and Autoextend

Autoextend and *Resize* are two Oracle options for influencing the size of the data files of the database system:

- The option *Autoextend* extends the data files automatically by a specified unit
- The option *Resize* lets you increase the size of data files manually (up to the maximum file system size), or reduce their size (down to the largest used block ID of the data file).

These options influence the following SAPDBA functions, and have been adjusted accordingly:

Restore/Recovery

When making a recovery (importing the offline redo log files), the Oracle database system automatically takes into account the Autoextends made to the data files in a database backup. The consistency checks made on files sizes during the database backup, and after the backup, are deactivated.

Database System Check

Parameters for checking freespace in tablespaces that take into account the *Autoextend* option:

- *TSP_FULL* (absolute freespace)
- *CRITICAL_SEGS* (critical segments)

Reorganization

- The preventative freespace check includes the *Autoextend* option
- When you reorganize a tablespace with data files, you can set the *Autoextend* parameters *MAXSIZE* and *INCREMENT_BY* when you specify the data files. The parameter *MAXSIZE* takes into account the memory of the file system or the raw device.

Tablespace Administration

- You can create the data files of the tablespaces with the parameters *AUTOEXTEND ON (OFF)*, *MAXSIZE* and *INCREMENT_BY*. You can also change these parameters. The parameter *MAXSIZE* takes into account the memory of the file system or the raw device.
- You can increase or reduce the size of the data files of the tablespaces with the *Resize* command. For more information, see [Resizing the Data Files of a Tablespace \[Page 591\]](#).

Starting SAPDBA

1. Log on to the host where your database system is running.



You can only use SAPDBA to manage a database system which is running on the same host system.

You can log on as the operating system user who owns the data files of the database system. The standard Oracle user created during the installation of the R/3 System is:

```
ora<sapsid>
```

You can also work with SAPDBA if you are logged on as the R/3 System user `<sapsid>adm`. This requires the authorization for program SAPDBA to be set accordingly (for example, under UNIX: `rwsr_xr_x oracl1 dba sapdba`). The advantage of this procedure is that the administrator who works with SAPDBA does not have the authorizations of user `ora<sapsid>` (who can delete database files directly, for example, and perform other critical operations for the database).

SAPDBA establishes the connection to the database with a special database user who has authorization to create and delete tablespaces, to create data files, and so on (DBA privileges). The default user is `system`.

For more information about the user concept, see [UNIX User Configurations \[Page 56\]](#).

For information about the environment variables, see [Environment Variables \[Page 45\]](#).



If you start SAPDBA with a script, as a background job or from the command line, you may not use special characters (such as `$` or `#`) in the user name and password.

2. If you want SAPDBA to log on as user `system` in your Oracle database system, call it in the following manner:

```
OS> sapdba
```

If the default password is not used, the system will require a password.

You can call SAPDBA with the following options:

- OS> **sapdba -p <path/profile name>**

You can also specify an initialization profile that is different to the standard profile.

If SAPDBA does not find another profile, it uses the values set in the default profile `init<DBSID>.dba`. See [Configuring SAPDBA \[Page 44\]](#). SAPDBA works with default values set internally if no profile is found.

- OS> **sapdba -e <number of extents>**

Specification of a number of extents to perform SAPDBA checks with this default value. The extent number 2 is always proposed if no value is specified explicitly. You can find this value in the [Check Functions \[Page 684\]](#) of the menu *Tablespace administration*. These analyses are also performed if there is a check run with `sapdba -check` (see `sapdba -check` in [SAPDBA Command Mode \[Page 69\]](#)).

Starting SAPDBA

- OS> **sapdba -u <user>/<password>**

Specification of a DBA user different to the default user (`system/<default password>`).

`<user name>`: Database DBA user that you defined

`<password>`: Password of DBA.

If possible, avoid starting SAPDBA with the command option `-u` and the immediate entry of user name and password. In this case, the command line with the DBA user and its password can be displayed in the UNIX process list (for example, by using the `ps` command).

Enter the password interactively. When you use `sapdba -u <user>`, the system will prompt you to enter a valid password. In this case, the password will not appear on the screen as it is entered and will not be displayed in the process list. Depending on the operating system the password length is limited to a certain number of places (for example, 8 characters for HP-UX, 32 characters for AIX).

- OS> **sapdba -l <user name>/<password>**

Specification of a DBA user different to the default user (`system/<default password>`). In contrast to the `sapdba -u` command option, the password appears on the screen when it is entered interactively (it does not appear in the process list). The length of the password is not limited.

The option `-l` is required, for example, to start SAPDBA with CRON.

- OS> **sapdba -U <superuser>[/<password>]**

The superuser is always `internal`. You can specify the password either as part of the command line or interactively. If you enter the password interactively then it appears neither on the screen nor in the process list.

After starting SAPDBA with `sapdba`, `sapdba -p`, `sapdba -e`, `sapdba -u`, `sapdba -l` or `sapdba -U`, the [SAPDBA: Initial Menu \[Page 86\]](#) appears.

The following examples illustrate different procedures for using passwords:



OS> **sapdba -u system** - SAPDBA prompts you for the password. The password is not visible on the screen.

OS> **sapdba -l system** - SAPDBA prompts you for the password. The password appears on the screen.

OS> **sapdba -l system < <file name>** - The password is written to the file `<file name>`. Access to this file can be restricted with operating system privileges.

OS> **sapdba -u /** - Call for an OPS\$ user (also applies to background processing). To make sure that the password is not visible in the process list, you can create an OPS\$ user (see ORACLE documentation and information in SAPNet) in the database and assign the same DBA authorizations as those of the `system` user.

You can get an overview of all the command options by entering the following command:

```
OS> sapdba -h[elp]
```

For information about other command options that are not mentioned in this section, see [SAPDBA Command Options \[Page 69\]](#).

SAPDBA: Release Information

Use

When you register problems with the SAP DBA tools with the Hotline, you must give as much information as possible about the SAPDBA version you are using. You can display this information with the command option `-version` in the SAPDBA menu.

Features

You can display the following information in the SAPDBA initial menu, by choosing *Additional functions* → *SAPDBA Release Information*.

- Date and time of compilation
- Newest source file with version (SCCS no.)
- Date of the source release
- Oracle version used for the compilation

See also:

[SAPDBA Command Mode \[Page 69\]](#) `sapdba -version`

SAPDBA Command Options

If you call SAPDBA with one of the following command options, the SAPDBA initial menu does **not** appear. Instead you can perform operations that are controlled by CCMS or an external scheduler, and that do not require interaction with the end user.



The specification of the objects after a command option cannot be longer than 255 characters.

The command options are written separately:

[-analyze \[Page 70\]](#)

[-check \[Page 72\]](#)

[-check_db_open \[Page 73\]](#)

[-checkopt \[Page 74\]](#)

[-cleanup \[Page 76\]](#)

[-delete \[Page 77\]](#)

[-export \[Page 78\]](#)

[-fill_tables \[Page 79\]](#)

[-next \[Page 80\]](#)

[-r \[Page 81\]](#)

[-shutdown\(_abort\) \[Page 82\]](#)

[-startup\(_dba\) \[Page 83\]](#)

[-version \[Page 84\]](#)

-analyze

-analyze

Phase II updates or recreates statistics, taking into account the entries in the control table DBSTATC.

Standard call

sapdba -analyze DBSTATCO

The first call of `sapdba -analyze DBSTATCO` uses the analysis method `E` to analyze all tables without statistics that are not in the control table `DBSTATC`. The next `analyze` run creates the statistics of these tables with the correct analysis method.

The `sapdba -analyze DBSTATCO` call creates up-to-date statistics for all tables entered in the control table `DBSTATC` for which the `TODO` flag is set. `sapdba -analyze NOOPTSTAT` is called implicitly. If a table is already in the control table, the analysis method specified there is used; otherwise SAPDBA uses the analysis method adapted to the size of the table according to the SAPDBA internal rules.

Syntax

sapdba

-u system/<password>

-analyze <tablespace | tablespace list | table | <keyword>>

-method E | C | EI | CI

-option R<n (*1000)> | P<n>

The command option correspond to those in phase I.

Key Words

DBSTATCO	All tables for which the <code>TODO</code> flag is set in the control table <code>DBSTATC</code> , and all tables and indexes without statistics (<code>NOOPTSTAT</code>).
NOOPTSTAT	All tables and indexes without statistics (excluding pooled and cluster tables and tables in the control table <code>DBSTATC</code> with <code>active = N</code> or <code>R</code>).

`analyze` is used for both the optimizer and for space analysis.

The analysis method and option specified with the command option have no effect if the key word `DBSTATCO` is used and an analysis method and option are entered for the appropriate tables.

The method for the `analyze` command is defined using two characters; the first is assigned to the tables and the second to the indexes. Since the optimizer always analyzes the indexes as well as the table, the second character must only be set if you want to analyze the indexes only, for example `sapdba -analyze PSAP%I`.

The `-analyze` logs are in the directory `sapcheck/<timestamp>.aly`.

For more information, see:

- [Support for the Cost-Based Optimizer \[Page 91\]](#)
- [CCMS Support for the Cost-Based Optimizer \[Ext.\]](#)

-analyze

- [SAPDBA Internal Rules for Determining Statistics for the CBO \[Page 95\]](#)

-check**-check**

Makes a database system check, covering the following aspects:

- Space monitoring (fill and fragmentation)
- Monitoring of Oracle alert messages
- Check on the `init.ora` profile parameters
- Physical consistency (availability of data files, control files and redo log files)
- Check on certain problem situations, based on SAP's experience in administrating R/3 databases

You can view the results of the database check in the logs under **sapcheck/<timestamp>.chk**.

Use the option **sapdba -check -e <no._of_extents>** to adjust the default number of extents (2), for checking the tablespace overflow. However, you should make individual adjustments to the database system check in the table **DBCHECKORA** (transaction DB17) in R/3 CCMS.

Start the database system check function weekly, if possible as a background process. For more information, refer to [Using the DBA Planning Calendar: Oracle \[Ext.\]](#).



When you start SAPDBA with a command option (which does not invoke the SAPDBA initial menu), you can use the operating system to substitute the password by reading a data file. Then the password does not appear in the process list of the operating system.

Example: `sapdba -check -l system < <file_name>`

The file `<file_name>` contains the password.

For more information, see [Checking the Database System \[Ext.\]](#) and [Configuring the Database System Check \[Ext.\]](#).

`-check_db_open`

Checks whether the database system has already been started. You can display the return code by using the command for checking the final state of the last command performed (for example `echo $status` for the C shell).

-checkopt**-checkopt**

Phase I determines whether new statistics are needed and enters this as a TODO flag in the control table `DBSTATC`. Using SAPDBA internal rules, methods are entered in `DBSTATC` for creating or updating obsolete or non-existent statistics.

Standard call

```
sapdba -checkopt PSAP% -method E
```

The `sapdba -checkopt PSAP%` call uses the primary index to determine the size of all tables. `checkopt` only enters tables into `DBSTATC` that already have statistics.

Syntax

```
sapdba
-u system/<password>
-checkopt <tablespace | tablespace list | table | <keyword>>
-method E | C
-option R<n (*1000)> | P<n>
-time <d>
-T -<h>
-force
```

SAPDBA Command Options

<code>-u</code>	Database user / password
<code>-checkopt</code>	Database objects and key words for the operation
<code>-method</code>	Analysis method
<code>-option</code>	Option for the analysis method (only for the method <code>E</code>)
<code>-time <d></code>	Updates all statistics older than <code><d></code> days
<code>-T <h></code>	Restricts the operation to <code><h></code> hours
<code>-force</code>	Resets the TODO flags in the control table <code>DBSTATC</code> regardless of the current situation

Key Words

<code>DBSTATC_TAB</code>	All tables entered in the control table <code>DBSTATC</code>
--------------------------	--

SAPDBA chooses the correct analysis method and option and enters them into the control table `DBSTATC` if the customer flag is not set. If the table is small the optimizer starts with the method `C`; if the table grows it switches to the method `E`.

The `checkopt` logs are in the directory `sapcheck/<timestamp>.opt`.

For more information, see:

- [Support for the Cost-Based Optimizer \[Page 91\]](#)

-checkopt

- [CCMS Support for the Cost-Based Optimizer \[Ext.\]](#)
- [SAPDBA Internal Rules for Determining Statistics for the CBO \[Page 95\]](#)

-cleanup

-cleanup

Removes logs and directories according to the default values set in the SAPDBA profile.

For more information, refer to [Initialization Profile init<DBSID>.dba \[Page 89\]](#) and [Deleting Log and Trace Files \[Page 263\]](#).

-delete

You can use this option to delete statistics for one or more tablespaces or for a particular table.

Input syntax: `-delete <tablespace(s)>|<table>`

You can also delete statistics that are not required by the cost-based optimizer, or that are harmful (pooled and cluster tables, tables with `Active = n` or `R` in `DBSTATC`).

Input syntax: `-delete HARMFUL`

[-checkopt \[Page 74\]](#)

[Support for the Cost-Based Optimizer \[Page 91\]](#)

[SAPDBA: DB Optimizer Control \[Page 93\]](#)

[CCMS Support for the Cost-Based Optimizer \[Ext.\]](#)

-export

-export

Input syntax: `-export <tablespace(s)/table>`

Exports the specified objects. A list or a LIKE value can be defined for tablespaces; a table must always be specified singly by its name. If there are no further specifications, the export is made to the null device `/dev/null`. This procedure is important for testing tables and indexes for corrupted Oracle blocks. No export dump file is generated.

You can, however, run a normal export by specifying a target for the export dump file (directory or tape device, specification of tape size, if required). Only the regular SAPDBA export scripts and logs are generated.

[Exporting and Importing Database Objects \[Page 637\]](#)

[SAPDBA: Export/import \[Page 639\]](#)

-fill_tables

This option fills the tables *DBSTATC*, *DBCHECKORA* and *DBAFID* with standard data. This does not overwrite any existing entries.

-next

-next

<tablespace(s)>

Alters the `next` parameters when necessary.

Call options (examples):

- `sapdba -next PSAP%`
For all SAP tablespaces.
- `sapdba -next PSAP%I`
For all SAP index tablespaces.
- `sapdba -next PSAPBTAB%`
For the tablespaces PSAPBTABD and PSAPBTABI.
- `sapdba -next <tablespace_name>`
Single tablespaces that grow quickly, for example PSAPBTABD (schedule more often).
- `sapdba -next <TSP_name1>,<TSP_name2>,...`
A list of tablespaces is possible (no space between the tablespace names).



You may **not** change the `NEXT` parameter for the `SYSTEM` tablespace.

For more information, refer to [Command Option -next \[Page 613\]](#).

-r

Input syntax: `-r <complete path>/<timestamp>`

Starts reorganization. Prerequisite: the restart script and the corresponding SQL and shell scripts (in directory **<timestamp>**) must exist. The user and the password are transferred with the command option `-P`, which is only for this purpose. Normally this option is only used for scheduling SAPDBA background processes.

For more information, refer to [Procedure for a Reorganization \[Page 592\]](#).

`-shutdown(_abort)`

-shutdown(_abort)

`-shutdown` shuts down the database cleanly.

`-shutdown_abort` shuts down the database immediately without ending any running processes cleanly.

For more information, refer to [Shutting Down the Database System \[Page 101\]](#).

-startup(_dba)

-startup starts up the database normally.

-startup_dba starts up the database in *DBA* or *RESTRICTED SESSION* mode.

For more information, refer to [Starting Up the Database System \[Page 100\]](#).

-version

-version

Displays administration information on the current SAPDBA version, such as the release, date of compilation of the current version, SAPDBA sources, development system, date of the last release, as well as the Oracle version used for compilation.

Logs written by **-check**, **-checkopt**, **-analyze**, and/or **-next** are in the directory **<ORACLE_HOME>/sapcheck**: **<timestamp>.chk**, **<timestamp>.aly**, **<timestamp>.nxt**.

Logs written by the startup or shutdown options are in the directory **<ORACLE_HOME>/sapreorg/<timestamp>.dba**.

[Displaying and Managing Logs and Profiles \[Page 87\]](#)

[SAPDBA Logs \[Page 296\]](#)

[Return Codes when Using the SAPDBA Command Options \[Page 85\]](#)

Return Codes for SAPDBA Command Options

There are three basic return codes:

0	Operation completed successfully
1	Operation completely successfully; warning(s) issued
2	Operation terminated with error message(s) issued

SAPDBA: Initial Menu

SAPDBA: Initial Menu

ORACLE version:	8.0.3.2.0
ORACLE_SID:	C11
ORACLE_HOME:	/oracle/C11
DATABASE:	shut down
SAPR3:	not connected

a	Startup/Shutdown instance	h	Backup database
b	Instance information	i	Backup offline redo logs
c	Tablespace administration	j	Restore/Recovery
d	Reorganization	k	DB check/verification
e	Export/import	l	Show/Cleanup
f	Archive Mode	m	User and Security
g	Additional functions	n	SAP Online Help
q	Quit		

- *Startup/Shutdown instance, Instance information* - See [Starting up and Shutting down the Database System \[Page 97\]](#)).
- *Tablespace administration* - See [Tablespace Administration \[Page 678\]](#)
- *Reorganization* - See [Database Reorganization \[Page 575\]](#)
- *Export/import* - See [Exporting and Importing Database Objects \[Page 637\]](#)
- *Archive mode* - See [SAPDBA: Archive Mode \[Page 31\]](#)
- *Additional functions* - See [Starting Individual SQL Scripts with SAPDBA \[Page 294\]](#), [SAPDBA: DB Optimizer Control \[Page 93\]](#) and [SAPDBA: Release Information \[Page 68\]](#).
- *Backup database* - See [Backing Up the Database \[Page 202\]](#)
- *Backup offline redo logs* - See [Archiving the Offline Redo Logs \[Page 210\]](#).
- *Restore/Recovery* - See [SAPDBA: Restore/Recovery \[Page 714\]](#)
- *DB check/verification* - See [SAPDBA: Check \(and repair\) Database \[Page 699\]](#)
- *Show/Cleanup* - See [SAPDBA: Show/Cleanup \[Page 88\]](#)
- *User and Security* - See [SAPDBA: Expert Mode \[Page 60\]](#), [Users and Roles \[Page 54\]](#), [Restricted Session \[Page 63\]](#) and [Users and Security \[Page 56\]](#).
- *Help*

Displaying and Managing Logs and Profiles

Use [SAPDBA: Show/Cleanup \[Page 88\]](#) to perform the following actions:

- Display the following log files:
 - [SAPDBA Logs \[Page 296\]](#)
 - Logs generated by the functions **-check**, **-checkopt**, **-analyze** or **-next** (see [SAPDBA Command Mode \[Page 69\]](#))
 - [Main Log \[Page 300\]](#)
 - [Structure Log \[Page 299\]](#)
 - BRBACKUP / BRARCHIVE logs
- Display the following initialization profiles:
 - ORACLE profile (**init<DBSID>.ora**)
 - BRBACKUP / BRARCHIVE / BRRESTORE profiles ([Initialization Profile init<DBSID>.sap \[Page 306\]](#))
 - SAPDBA profile ([Initialization Profile init<DBSID>.dba \[Page 263\]](#))
- Delete the following files (see [Deleting Log and Trace Files \[Page 89\]](#)):
 - SAPDBA logs and directories
 - BRBACKUP / BRARCHIVE logs
 - ORACLE trace files, audit logs

For an overview of the logs written by the SAP utilities, see [SAPDBA Logs \[Page 296\]](#) and [BRBACKUP, BRARCHIVE and BRRESTORE Logs \[Page 434\]](#).

SAPDBA: Show/Cleanup

SAPDBA: Show/Cleanup**Display**

Select *Show/Cleanup* → *Show log files/profiles*, to enter the display of the logs or profiles.

Show log files / profiles

<i>a</i>	- SAPDBA log files
<i>b</i>	- SAPDBA daily check log files
<i>c</i>	- SAPDBA main log
<i>d</i>	- SAPDBA struct log
<i>e</i>	- BRBACKUP log files
<i>f</i>	- BRARCHIVE log files
<i>g</i>	- <i>init<SID>.ora</i> (ORACLE profile)
<i>h</i>	- <i>init<SID>.sap</i> (BRBACKUP, BRARCHIVE)
<i>i</i>	- <i>init<SID>.dba</i> (SAPDBA profile)
<i>q</i>	- Return

Delete

Select *Show/Cleanup* → *Cleanup log files/directories* to delete logs, directories, trace files, or audit files.

Cleanup

<i>a</i>	- SAPDBA log files and dump directories
<i>b</i>	- SAPDBA daily check log files
<i>c</i>	- BRBACKUP log files and directories
<i>d</i>	- BRARCHIVE log files
<i>f</i>	- ORACLE traces and audits
<i>q</i>	- Return

Deleting Log and Trace Files

It is important that you regularly delete old log and trace files, particularly those which were created by the database system.

If the log file directory for the database ALERT and trace files fills up, the database system issues error messages which can be hard to analyze. To free more space, you should also regularly delete the operating system audit trail.

SAPDBA enables you to delete the following types of log and trace files:

- Logs and any directories of SAPDBA, BRBACKUP, BRARCHIVE



SAPDBA logs and associated directories for which the action is marked as an error in the main log are not deleted from SAPDBA.



The deletion of BRBACKUP/BRARCHIVE logs means that the information for the corresponding backups in the Computing Center Management System of the R/3-system can no longer be displayed. Also note the following: If an backup for which the corresponding log has been deleted needs to be restored with SAPDBA, you must first restore the log.

Always ensure that there is a sufficiently large number of logs present (SAPDBA recommended value: only delete logs older than 30 days).

- ORACLE trace files: SAPDBA deletes the appropriate database and user trace files without backing them up (directory **<SAPDATA_HOME>/saptrace/background** or **<SAPDATA_HOME>/saptrace/usertrace**).
- ORACLE audit logs: SAPDBA deletes the appropriate files without backing them up (directory **<ORACLE_HOME>/rdbms/audit**). If more than 4095 files appear in the audit trail, SAPDBA can only delete 4095 audit logs at once. To delete additional logs, repeat the operation.

If you do not explicitly specify the number of days for which deletion of the logs/directories should be blocked, the default values generally set in the profile **init<DBSID>.dba** will be used (see [Sample Initialization Profile \[Page 288\]](#)).

ALERT File

The ALERT file (**alert_<DBSID>.log**) of the database system is stored in directory **<SAPDATA_HOME>/saptrace/background**. Database activities are logged continuously in this file. If database problems occur, you can use this file to find out what happened. This means that the ALERT file contains very important information. For this reason, SAPDBA does not provide an option for deleting this file. This is for your own data security.

Audit Logs

Audit logs contain information about the inspected database action, the user who performed the operation, and the date and time of the operation. Audit logs can be saved in the operating system audit trail (**<ORACLE_HOME>/rdbms/audit**). Even when no explicit auditing is active,

Deleting Log and Trace Files

ORACLE writes an auditing log for every logon to the database. As a result, the directory may grow very quickly. Regularly delete the old audit logs.

Command Line Options

You can also delete log, trace, and audit files with the command option **sapdba -cleanup** (also see [SAPDBA Command Mode \[Page 69\]](#)). Check whether the parameters in **init<DBSID>.dba** are configured correctly (see [Initialization Profile init<DBSID>.dba \[Page 263\]](#)). When this option is called, SAPDBA deletes all the above log files and directories that are older than the specified interval (in days).

Support for the Cost-Based Optimizer

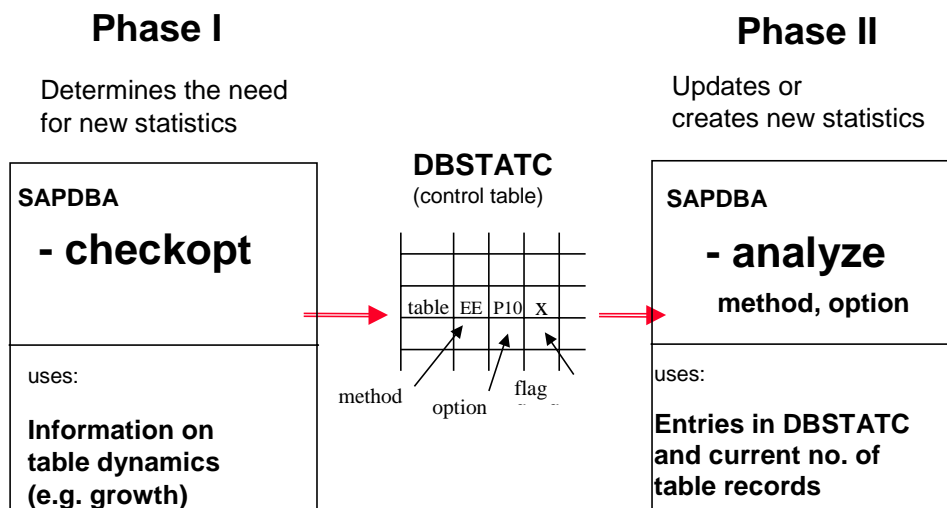
The cost-based optimizer (CBO) is an essential component for speeding up the performance of the Oracle database system. It uses statistics to examine how the database has been accessed and then define an optimum access strategy. The CBO replaced rule-based optimization in R/3 Release 4.0.

As a prerequisite for using the cost-based optimizer the parameter `OPTIMIZER_MODE` in the Oracle initialization profile `init<SID>.ora` must be set to `CHOOSE`.

Two-Phase Concept

SAP has introduced a two-phase concept to keep the CBO's statistics permanently up-to-date:

- Phase 1 (determine need)**
 SAPDBA determines the need for new statistics and enters this in the control table `DBSTATC`. Using SAPDBA internal rules, the analysis method and option for phase II are entered in the control table.
- Phase II**
 SAPDBA updates the old statistics or creates new statistics, taking into account the entries in the control table `DBSTATC`.



Run phases I and II regularly. You can make this process fully automatic by scheduling the commands `sapdba -checkopt` and `sapdba -analyze` in the DBA Planning Calendar in CCMS.

Support for the Cost-Based Optimizer

For more information, see:

- [Support for the Cost-Based Optimizer \[Page 91\]](#)
- [CCMS Support for the Cost-Based Optimizer \[Ext.\]](#)
- [SAPDBA Internal Rules for Determining Statistics for the CBO \[Page 95\]](#)

SAPDBA: DB Optimizer Control

Use

Use the CCMS for checking and updating statistics for the Oracle cost-based optimizer (CBO).

The SAPDBA menu option **DB Optimizer control** (SAPDBA menu *Additional functions*) is another way of displaying statistical information, checking the requirements for new statistics in online mode, deleting superfluous statistics, and creating missing statistics.

Features

Displaying Existing Statistics

Function: **Show all statistics information**

This display tells you how many tables or indexes of the user SAPR3 have no statistics, and how many tables entered in the control tables *DBSTATC* have obsolete statistics (TODO indicator in *DBSTATC*).

It also gives you information about the number of tables entered in control table *DBSTATC*, the number of tables explicitly excluded from the statistics (Active = N), the number of tables and indexes excluded from the CBO statistics (Active = R or N), and the number of tables that are marked with a customer indicator in the control table (*..user defined values*).

Checking Which Tables Need New Statistics

Function: **Check for new statistics**

a	Delta (percent) for new stat. indicator
b	Forced check (refresh marked tables)
c	Time limitation
d	SAPR3 Table (for option g)
e	Tablespace (for option h)
f	Start check for tables in control table <i>DBSTATC</i>
g	Start check for table
h	Start check for tables in tablespace

You can use this check either for a single table, for all the tables of a certain tablespace, or for all the tables of the control table. You can specify the threshold for the percentage change in table records from which the statistics are considered obsolete (menu option a). You can also restrict the check to a certain period of time (menu option c). However, you should not normally set a time limit.

A *Forced check* (menu option b) updates the entries in the control table, independently of the current status; this means that all existing TODO indicators are reset.

Deleting Unnecessary Statistics

Function: **Delete statistics excluded for R/3**

Deletion of tables not required for the R/3 System, for example pool and cluster tables or tables marked as such in the control table *DBSTATC* (Active = N or R).

Creating Missing Statistics

Function: **Create missing statistics for all tables and indexes**

SAPDBA: DB Optimizer Control

When, for example, you add one or more new tables or indexes, you can easily create statistics for them using this menu option.

For more information about the standard procedure for updating statistics, see [CCMS Support for the Cost-Based Optimizer \[Ext.\]](#).

SAPDBA Internal Rules for Determining Statistics

Phase I (-checkopt)

Phase I is based on the command `analyze index` which you can use to quickly display the current number of table records.

Step A

Determines the number of old table records from

- the Oracle Data Dictionary (table `DBA_TABLES`) or
- the table `DBSTATTORA` if there are no entries in the Oracle Data Dictionary

Step B

decides which analysis method to use for `analyze index`.

Number of old table records (X)	Analysis method	Option
X < 6000	C	
6000 < X < 60000	E	R6
60000 < X	E	P10
X unknown	E	P10

Step C

determines the number of current table records with the command `analyze index estimate/ compute statistics` and the analysis method from step B.

Step D

updates the control table `DBSTATC` (TODO flag and analysis method from step B). The TODO flag is set according to the number of current table records:

<200.000	and	Change > 10%		
>200.000	and	< Number of old table records	and	Change > 10%
>200.000	and	> Number of old table records	and	Change > 100%

Tables are entered in the control table `DBSTATC`, if

- statistics already exist and change > threshold
- the analysis method (from step B) has changed

Tables are deleted from the control table `DBSTATC`, if

- the date in the field `TODO chg.date` is older than 30 days, the type of usage is set to O, the customer flag is not set and the control flag is set to A.
- the table no longer exists in the database

SAPDBA Internal Rules for Determining Statistics

SAPDBA enters the appropriate analysis methods and options in the control table `DBSTATC` according to the rules from step B. SAPDBA uses the up-to-date number of table records from step C when doing this.



If the customer flag is set, SAPDBA does not change the settings in the control table `DBSTATC`.

Phase II (-analyze)

The analysis method and option specified in the control table are used `DBSTATC` if they create more precise statistics than those specified in the command option, or if the customer flag is set. If there is no entry in the control table for a table, then the analysis method and option from step B in phase I are used.

SAPDBA create statistics for all tables that require them:

- Implicitly, with the command **sapdba -analyze DBSTATC (DBSTATCO, DBSTATCA_TAB, DBSTATCAA_TAB)**
- Explicitly, with the command **sapdba -analyze NOOPTSTAT**

For each `checkopt` and `analyze` run, SAPDBA implicitly deletes the statistics of

- tables with the `R` or `N` in the control flag (*Active*)
- cluster tables and pooled tables

SAPDBA explicitly deletes statistics for each `delete` run.

These statistics are also used by applications that need space information (such as the R/3 Application Monitor). When statistics are created, all tables with the control flag `R` are analyzed; the statistics are then deleted. The statistics results are stored in the tables `DBSTATTORA` or `DBSTATIORA`.

For all tables without statistics you can create them with the keyword `NOOPTSTAT` (`sapdba -analyze NOOPTSTAT`).

Starting up and Shutting down the Database System

You can use SAPDBA to start up and shut down the database system. You should use the commands STARTSAP and STOPSAP for normal database and SAP system startups and shutdowns.

You can also use the SAPDBA command options `-startup`, `-startup_dba`, `-shutdown`, `-shutdown_abort` to start or stop the database system ([SAPDBA Command Mode \[Page 69\]](#)).

SAPDBA also supports the start up of ORACLE databases with OPS configuration (see also [ORACLE Databases and Parallel Server Option \[Page 769\]](#)).

[Status of the Database System \[Page 98\]](#)

[Starting up the Database System \[Page 100\]](#)

[Shutting down the Database System \[Page 101\]](#)

Status of the Database System

Status of the Database System

SAPDBA provides you with information about the status of your database system at various locations. Some of these possibilities are listed here:

- **Status displays:** The initial SAPDBA screen already shows you the status of your database system.
 - ORACLE_SID, ORACLE_HOME of the local instance are displayed
 - INSTANCE: The status of the local database instance is displayed (e.g. *Instance shut down, open*)
 - SAPR3: There is a check whether a `sapr3` user is connected to the database (*connected, not connected*).
- **Instance information:** Further information about the database instances can be obtained using the menu option *Instance information*.

<i>SELECTED INSTANCE</i>	Name of the instance (e.g. <i>C11</i>)
<i>INSTANCE STATUS</i>	Status of the instance (e.g. <i>shutdown, mounted, opened</i>)
<i>INSTANCE HOME</i>	Local home directory (e.g. <i>/oracle/C11</i>)
<i>CONNECTED PROGRAMS</i>	Number of processes connected with the database (e.g. <i>1</i>)
<i>SAPR3</i>	Logon status of user <code>sapr3</code> (<i>connected, not connected</i>)

<i>a</i>	- <i>Select remote instance</i>
<i>b</i>	- <i>Select local instance</i>
<i>c</i>	- <i>Show summary of all instances</i>
<i>d</i>	- <i>Refresh</i>
<i>q</i>	- <i>Return</i>

This menu option should help you to obtain information about all the instances involved if your ORACLE database has an OPS configuration. You can obtain the specified status information for both the local instance (default value or *Select local instance*) and the other parallel instances (*Select remote instance*). With *Show instance summary* you can obtain an overview of all the parallel instances.

If you have a non-OPS installation, the instance information does not provide you with any new information. In this case, the information in the status display of the initial screen should be sufficient.

See [ORACLE Databases and Parallel Server Option \[Page 769\]](#).

Status of the Database System

- You can use the menu option *Startup/Shutdown instance* to start/stop the database instances. In this case, too, the status information for the local instance, the parallel instances, the number of processes logged and user `saprc3` is displayed.

Starting Up the Database System

Starting Up the Database System

You can start ORACLE databases with one instance (non-OPS configuration) and databases with several parallel instances (OPS configuration) using SAPDBA.

Non-OPS Configuration

From the initial SAPDBA menu, choose **Startup/Shutdown instance** → **Startup...** You can start the database system in one of the following modes:

- Startup normal: Select *Startup normal* to start up the database instance for regular, productive operation. SAPDBA will attempt to open the database instance. If you need a different status of your database system (**nomount, mount**) for your DBA tasks, use the ORACLE tool SQLDBA to start up the system.
- Startup RESTRICTED: Select *Startup RESTRICTED* to start the database instance in RESTRICTED-SESSION mode (generally exclusive for the database administrator) (see [Restricted Session \[Page 63\]](#)).

Start the database system in this mode, for example, if you want to reorganize the database. Only the user with RESTRICTED-SESSION privilege (generally the DBA user) has access to the database instance during this operation. If the database instance is already in production operation, you must shut it down and restart it in DBA mode.

The database system issues messages about the progress of the startup in both modes.

OPS Configuration

From the initial SAPDBA menu, choose **Startup/Shutdown instance** → **Startup...** Have a look at the status information. You can start all the instances (*Startup all instances*) or instances selected with *Select remote/local instance (Startup selected instance ...)*. You can also choose between the aforementioned options *Startup normal/RESTRICTED*.

Shutting down the Database System

You can shut down ORACLE databases with one instance (non-OPS configuration) and databases with several parallel instances (OPS configuration) with SAPDBA.

Non-OPS Configuration

From the initial SAPDBA screen, choose **Startup/Shutdown instance → Shutdown....** You can stop the database system in one of the following modes:

- **shutdown normal:** Select *Shutdown normal* to shut down the database system normally. Make sure only one process is connected to the database instance (status: *Connected Programs: 1* - that is, SAPDBA is the only user process) if you want to use this option. You should shut down the database instance normally whenever possible.
- **shutdown immediate:** When more than one process is logged on to the database instance, you can use *Shutdown immediate* to close the database instance.

SAPDBA will not carry out a `shutdown` as long as the following conditions apply:

- User `sapr3` is still linked to the database system (the SAP System is still running).
- Database contents are currently being backed up (the SAP program BRBACKUP is currently running).

If the database system is currently archiving online redo log files to the archiving directory (process ARCH is just backing up this online redo log file), the shutdown process may take several minutes.

If you use `shutdown immediate` to close the database instance, the links to all database users will be terminated as soon as their current commands have been processed. The ORACLE database system makes sure all open transactions are terminated and rolled back.

When the database instance has been shut down (with `shutdown normal` or `shutdown immediate`), the state of the database instance is consistent. As a result, when you continue using the database system at a later time, no recovery procedures are required.

Shutting down the SAP System before the database system shuts down the background processes that are part of the SAP System. To find out whether background processes from other database instance users exist, enter the following command (depending on the operating system; here is a UNIX example):

```
OS> ps -eaf | grep ora
```

When the database system has been shut down, no “ora*” processes should appear. If background processes are still active, they should be shut down in order to avoid problems during reorganization and other operations (for example, using the UNIX command `kill <process number>`).

End backup: During the shutdown procedure, SAPDBA carries out an ALTER TABLESPACE END BACKUP command for all the tablespaces involved. This command guarantees that every tablespace is closed in a consistent status. This can be necessary if the tablespace was prepared for saving with ALTER TABLESPACE BEGIN BACKUP but was not released with END BACKUP. Otherwise, the database system would require you to restore the tablespace in question during the next system startup.

Shutting down the Database System

- **shutdown abort:** Select *Shutdown abort* in the following cases:
 - You could not close the database instance with `shutdown normal` or `shutdown immediate`.
 - You know that a problem has occurred and that you must therefore close the database instance with `shutdown abort`.

If you are not quite sure, always attempt to shut down the database first with `shutdown immediate` before you use `shutdown abort`. In general, `shutdown abort` is only used for serious problems, such as data loss in connection with a volume or other volume-related problems.

When `shutdown abort` is used, the database instance is stopped immediately, with no regard for active transactions. The stopped database system therefore will have an inconsistent status.

Do not use `shutdown abort` to attempt to “kill” the database instance quickly if you can use `shutdown immediate` instead. After `shutdown abort`, there will be an immediate automatic recovery of the ORACLE instance (instance recovery) when starting again. Data changed after successful completion of a transaction is recovered automatically, but data changed by terminated transactions will be lost. The automatic instance recovery can significantly slow down the next startup operation. The overall time taken to stop and start the database instance is then significantly greater than with `shutdown immediate` and database instance startup.

OPS Configuration

From the initial SAPDBA screen, choose **Startup/Shutdown instance → Shutdown...** Have a look at the status information. You can stop all the instances (*Shutdown all instances (normal/immediate/abort)*) or an instance selected with *Select remote/local instance (Shutdown selected instance...)*. You can choose between the options *Shutdown normal/immediate/abort* described above in this case as well.

Backup Concepts

Before you make your first backups, you must consider carefully what backup methods are most suitable for your needs and the conditions in your system. Carefully read the various [sources of information available to the database administrator \[Page 23\]](#).

Consider the following in detail:

- Which processes you want to start **manually** and which you want to run **automatically**:
Read about the options under:
[Computing Center Management System \[Page 24\]](#)
[Unattended Backups \[Page 139\]](#)
[Scratch Volumes \[Page 146\]](#)
[Automatic Volume Management \[Page 153\]](#).
- In which **mode** you want to make the backup and how much **time** you want to schedule for it: Read the following:
[Online Mode and Offline Mode \[Page 127\]](#)
[Duration of the Backup \[Page 134\]](#)
and time-saving methods such as [Parallel Backup \[Page 135\]](#).
- **What** you want to back up and **how often**: Read about the following options
[Complete Backups \[Page 107\]](#)
[Incremental Backups \[Page 108\]](#)
[Backup Strategies with RMAN \[Page 110\]](#)
[Backing Up Executable Programs and Other Components of the R/3 System \[Page 132\]](#)
[Backing Up Data Tablespaces \[Page 133\]](#).
- When you want your database to be **in which status**: Read about the following
[Consistent Online Backup \[Page 128\]](#)
[Structure-Retaining Database Copy \[Page 130\]](#).
- **Which media** you want to use: Read about:
[Backup Media \[Page 114\]](#)
[Volume Management \[Page 142\]](#)
[Tape Devices with Hardware Compression \[Page 115\]](#).
- How much **data security** you want to have: Read about the following configurations:
[Standby Databases \[Page 185\]](#)
[Split Mirror Disks \[Page 192\]](#)
[Backup Verifications \[Page 155\]](#).

These are just guidelines for you to think about when you organize your database backup and administration concept. Individual circumstances mean that you may have to adapt your concept at times; this requires detailed knowledge. Remember that in emergencies you do not have the time to read about these concepts in depth; make sure that you have learnt the relevant information in advance.

Logical Backup and Image Backup

Logical Backup and Image Backup

Oracle supports two types of database backup:

- **Logical backup**

- **SAP-Programs for Logical Backups: SAPDBA**

The ORACLE export/import programs enable you to back up specific objects in the database. These backups are not a suitable replacement for image backups of the database, however, because the database can then only be recovered to the condition at the time of the backup. Database changes made after the backup cannot be recovered.

R/3 tables are usually used by multiple users; this means that it is a good idea to make user-related backups with Export/Import. For example, you do not want a user to restore the table ATAB in order to retrieve lost entries in a particular R/3 table.

Logical backups of R/3 objects can be performed using the R/3 utility *R3trans*. *R3trans* exports R/3 System objects (among others) from the database into operating system files. If a user then inadvertently deletes an object, that object can be imported from the exported backup file. For more information, see to the documentation on *Transport Tools*.

- **Image backup**

- **SAP Programs for Image Backups: BRBACKUP**

Image backups are complete block-for-block copies of the database (online or offline). Image backups are required in order to recover the database in a consistent and current state.

You can perform complete image backups or image backups of specific tablespaces. Tablespaces that are used intensively should be backed up separately between full backups - this can speed up any necessary recovery.

Backup Methods and Backup Frequency

Backing Up the Complete Database (Production System)

- The frequency of image backups of the entire database should depend on the degree of activity in your database system. High activity in the database increases the number of redo log files written between complete backups. This increases the time required for any necessary recovery.
- The safety level of your database operations is increased the more often you run backups. If a redo log file is lost, complete recovery of the database after an error is often not possible. Instead, you will only be able to recover up to the gap in the redo log file sequence.
- Performing frequent complete backups reduces the number of redo log files which must exist in order to make a complete recovery. The data loss should one of these files be lost can thus be kept to a minimum.
- To perform an online backup, the database must be in *ARCHIVELOG* mode; production operations can then be continued without restrictions, unlike an offline backup.
- SAP recommends keeping several generations of complete backups and the corresponding redo log files. This ensures that you can still recover the database, even if the last complete backup is lost.
- To enable fast, simple recovery of the database, back up at least the changed tablespaces and the control file after every structure change (new, changed, or deleted tablespaces; new data files). After a reorganization with data files, always back up the affected tablespace if you want to use the recovery functions of SAPDBA. Follow the instructions for the tablespace backup below.
- Use the SAP utility BRBACKUP to back up the database and BRARCHIVE to archive the offline redo log files (see [Using BRBACKUP and BRARCHIVE \[Page 219\]](#)).

Backing Up a Tablespace

Backing up tablespaces that are changed frequently can reduce the time required for any necessary recovery. When a more recent backup of an intensively used tablespace is available, fewer redo log entries will have to be processed in order to update the tablespace. If you are able to back up the entire database on a daily basis, tablespace backups are not necessary.

Note that tablespace backups are no replacement for frequent backups of the entire database.

- If you only perform tablespace backups for a long period of time, this increases your dependence on the archived redo log files, and therefore the risk of data loss if one of the redo log files is lost.
- If tablespace backups are used, the database administrator decides what has to be backed up. The SAP utility BRBACKUP supports the backup operation itself, but does not help you decide which tablespaces to back up (exceptions: operations such as a tablespace extension, or an SAPDBA reorganization; in these situations, SAPDBA recommends that you backup the tablespace immediately).

Backup Methods and Backup Frequency

Backing Up the Control File

Another type of partial backup is to back up the control file. The control file records the physical file structure of the corresponding database. You should therefore back up the control file after every structure change.

Mirrored control files protect you against the loss of a single control file. If data files are damaged, an older control file that mirrors the corresponding structure of the database may be necessary for recovery. For this reason, mirroring the control files is by no means a replacement for backing up the control file after every change in the structure of the database.

When the SAP utility BRBACKUP is used to back up the database files, the control file is always saved along with them. The control file is saved before and after the operation for various administration measures with SAPDBA (for example, tablespace extension, reorganization of a tablespace).

Backing Up a Test System

The data of a test database might not have to be backed up as often, depending on how your test system is used. If you accept the restriction that you will only be able to recover the database from the last offline backup, you can operate the database in *NOARCHIVELOG* mode. If you do not back up the database at all, you will have to reinstall the database in a recovery situation.

Complete Backups

Use

As of Release 4.5, SAPDBA is integrated with the Oracle Recovery Manager (RMAN). You can use RMAN to make [incremental backups \[Page 108\]](#). To decide which database backups are the best reference for incremental backups you need to know what different types of backup involve and be able to tell them apart.

[RMAN Backup Strategies \[Page 110\]](#)

Definitions

Complete backup is a **generic term**. A complete backup can be one of the following:

1. A **whole backup** `-m all`:
Backs up all database objects. This backup is not cataloged which means that you cannot use this backup as a reference backup for an incremental backup with RMAN.
Syntax: `backup_mode = all`
2. A **full backup** `-m full`:
Backs up all database objects. You can make a full backup with or without RMAN. If you do not use RMAN, then RMAN is called separately to catalog the backup. This means that you can use this backup as a reference backup for an incremental backup with RMAN.
Syntax: `backup_mode = full`
In the context of the Recovery Manager this backup is a level 0 backup.

Incremental Backup

Incremental Backup

Use

As of Release 4.5A, BRBACKUP supports the incremental backup of the database with the Oracle Recovery Manager (RMAN). In an incremental backup, only the changes that have been made since the last complete backup are saved.

The incremental backup strategy optimizes the performance of the data backup; it saves both time and memory. If you do not save as much time as you expect, then this is because each block has to be checked to see if it needs to be backed up, whether it has been changed or not.

The incremental backup strategy is of special advantage for large databases which have few changes on a daily basis.

Prerequisites

You can only make incremental backups with the Oracle Recovery Manager (RMAN). The [integration of RMAN into the SAPDBA environment \[Page 158\]](#) offers you more functions.

[RMAN Backup Strategies \[Page 110\]](#)

Activities

To be able to make an incremental backup, you must first make a reference backup, a so-called complete backup (level 0). The complete backup (level 0) of the database backs up all Oracle database blocks that have already been used.

You can then make incremental backups: The incremental backup (level 1, cumulative) of the database backs up all Oracle database blocks that have changed since the last complete backup (level 0).

The following describes a possible backup scenario:

- Sundays: Full backup (level 0) of the database
- Monday to Saturday: Incremental backup (level 1, cumulative) of the database



The SAP incremental backup has the advantage that a complete backup (level 0) of the new files is made after each structure change and extension of the database. This means that you do not need to make a complete backup of the entire database.

Incremental Backups After Structural Changes

If changes are made to the structure of the database during a backup cycle, for example, if a data file is added, then RMAN terminates in the next incremental backup. However, the integration of RMAN into the SAP backup program BRBACKUP (either with the SAP backup library or an external backup library) lets you make incremental backups even if the structure of the database has been changed.

If, for example, a new file is created in the period between two full backups, all subsequent incremental backups until the next full backup are distributed internally across two savesets.

1. The changes in the old files are backed up incrementally, as usual.

Incremental Backups After Structural Changes

2. BRBACKUP also creates a new saveset, in which it stores the new files. Each of these new files is backed up in full in each incremental backup (level 0).

RMAN Backup Strategies

RMAN Backup Strategies

Use

The decision whether to use the Recovery Manager or not, and which backup library to use with it, depends largely on the strategy that you use to back up your data. The best strategy depends on the size of the database, the amount of data added or changed each day, the backup media you use and your security requirements.

Normal Backups

Normal backups, that is complete or partial backups of the database (`backup_mode = all | <object_list>`), were previously made with BRBACKUP or with an external backup tool and the BACKINT interface. As of Release 4.5A you can also make these backups with the Recovery Manager. The advantages of the RMAN include automatic recognition of database block errors, a lower backup volume and fewer redo logs.

You can make RMAN backups without a backup library to a local disk only. The Oracle server process takes care of the entire backup to disk. The data on the disk is a one-to-one copy of the database files. You cannot use RMAN to make a normal backup to a remote disk. If you want to make RMAN backups to tape devices, you need a backup library for tape management purposes.

Incremental Backups with RMAN.

As of Release 4.5A, BRBACKUP supports incremental backups of the database (RMAN). The Oracle Recovery Manager offers different types of incremental backups (different levels, cumulative or non-cumulative). The integration of RMAN into BRBACKUP is limited to cumulative incremental backups of level 1. This means that in the event of a recovery you only need to import one incremental backup. In an incremental backup, only the changes that have been made since the last complete backup are saved. This significantly reduces the amount of data that has to be backed up. This strategy is particularly useful for large databases.

An incremental backup requires an earlier reference backup, a complete backup of the database (level 0). RMAN backs up all Oracle database blocks that have already been used. The subsequent incremental backup (level 1) of the database backs up all Oracle database blocks that have changed since the last complete backup (level 0). This takes into account all changes that have been made to the database.

You can only make incremental backups (level 1) with RMAN. When you make the complete backup (level 0) you also have the option of using BRBACKUP or an external backup tool and BACKINT. BRBACKUP automatically logs the backup as a level 0 backup.



An RMAN backup has to read the database to see if a block has been used or changed just as many times as a backup without RMAN. This means that an incremental backup may not reduce backup time. A significant reduction in backup time is achieved only if the speed of the tape devices is the reason for slow backups.

For security when using an incremental backup strategy, choose a sufficiently long expiration period for the tapes (at least 28 days), so that you always have several complete backups available. Make sure the offline redo log files cover the period up to the oldest complete backup, and that they are backed up daily.

Features

Full backup (full, level 0)

- Full backup (level 0) with RMAN
 - to a tape device with the SAP backup library
 - to a local disk without backup library
 - with an external backup library to the appropriate media
- BRBACKUP backup to local or remote tape devices, local or remote disks followed by cataloging
- Backup with external tool and BACKINT followed by cataloging

Incremental backup (Incr, level 1)

- RMAN
 - to tape device or remote disk with the SAP backup library
 - to a local disk without backup library
 - with an external backup library to the appropriate media

Activities

The following describes a possible incremental backup scenario:

- Sundays: Full backup (level 0) of the database
- Monday to Saturday: Incremental backup (level 1) of the database

The tools used to implement this backup strategy depend on the specific requirements of the end user regarding backup media, volume of data and so on.

For more information on incremental backups, refer to the relevant Oracle8 documentation.

Incremental Backups After Structural Changes

Incremental Backups After Structural Changes

What action should you take when the structure of the database has been changed during a backup cycle, for example when a data file has been added?

This would lead to the Oracle Recovery Manager failing to make the next incremental backup. However, the integration of RMAN into the SAP backup program BRBACKUP (either with the SAP backup library or an external backup library) lets you make incremental backups even if the structure of the database has been changed.

For more information, see [Incremental Backup Strategy Without a Backup Library \[Page 164\]](#).

If you use an SBT backup library, savesets will always be created. Savesets are created on the backup medium and contain one or more files (see [Saveset Grouping \[Page 167\]](#)). During incremental backups the changes are normally saved in a single saveset. If the structure of the database has been changed during a backup cycle, a second saveset is created to which the new files are backed up. In each subsequent incremental backup in the backup cycle, the changes in the files that existed at the time of the last complete backup are written to **one** saveset. In addition, the new files are backed up in their entirety (not just the changes) to another saveset in every incremental backup that is made until the next complete backup. This has the advantage that you can specify exactly what you want to restore when you restore an incremental backup.

Restoring Incremental Backups with RMAN

Use

Any backups made with RMAN must also be restored with RMAN. This does not apply to backups made to disk. As of Release 4.5A SAPDBA supports a database reset to an incremental backup. As of Release 4.5B, SAPDBA also supports a partial restoration followed by a recovery using incremental backups. You can continue to recover the database by importing the offline redo log files if you want.

Prerequisites

In contrast to the BRRESTORE restoration process (without RMAN) where the database has to be closed, a restoration with RMAN needs the database to be mounted. A restoration for the incremental backup strategies consists of several steps.

Features

The procedure for resetting the entire database to an incremental backup, or to a point in time before the database failure (database reset, point-in-time recovery, disaster recovery) is as follows:

- 1.) Prerequisite: The database is closed.
Restoration of the control file and, if needed, the online redo log files with BRRESTORE (without RMAN) from the last incremental backup.
- 2.) Prerequisite: The database is mounted.
Restoration of the full backup (level 0) (with RMAN).
Restoration of the last incremental backup (with RMAN), possible rest point.
- 3.) Prerequisite: The database is mounted.
Import the offline redo log files into the database to reset the data to the required point in time.

If only some of the files are corrupt (partial restore and full recovery), perhaps due to a media error, then you only need to restore those files from the full backup and then restore the last incremental backup. RMAN then makes the changes to the files automatically, restoring them to the last incremental backup. The current database is recovered by restoring the redo log files.



If the database has been restored since the last full backup (level 0), then you cannot use it as a reference for the next incremental backup.

[Incremental Backup \[Page 108\]](#)

[Incremental Backups After Structural Changes \[Page 112\]](#)

[RMAN Backup Strategies \[Page 110\]](#)

Backup Media

Backup Media

Standard backup media are backup devices that are locally connected to the database server.

- Only back up a production database to a remote host if the database is not too large, and the network is stable.
- You should be able to back up a test database to a remote host without any major problems. You may want to back up the test database to backup devices that are connected to the host on which the production database is running.

Of course, you can also back up the database to disk, when you have enough storage space available.

Tape Devices with Hardware Compression

Purpose

Always use this option if your tape devices support hardware compression. This reduces back up time because more data can be written to a single volume. Tape units with hardware compression are very common. The compression method used is normally based on the Lempel-Ziv algorithm.

How Useful is Compression?

The amount of data that can actually be written to a tape depends on the compression rate. The average compression rate is between 3 and 5, however this can vary for several reasons:

It is lower if

- the data is mostly already compressed. the compression rate does not improve if the files are compressed again.

It is higher if

- New or relatively empty database files are compressed.

When Is it Good to Compress?

After a large data transfer or a reorganization of a tablespace, the affected tablespaces must be compressed again. If a database file has no essential changes in two consecutive compression runs, this compression rate can also be seen as a constant. A further check of the compression rate must only be carried out after a longer period (for example, after a year).

BRBACKUP can optimize a backup on tape units with hardware compression if the current compression rates are known before starting the backup. To do this, there is a BRBACKUP option which approximates compression rates: **brbackup -k only**. Repeat this activity at least once a month to update the compression rates.

Compression can take a very long time when large databases are involved (around 100 hours for 300-400 GB) and do not result in much change in compression rates. You can exclude these files from regular compression since the compression rate stays constant. To reduce compression time for large databases you can reduce the amount of data by compressing database files individually, or excluding them from compression. You can also run multiple compressions in parallel. Parallel compressions to determine the compression rates (without starting a backup) are used.

Syntax

Set the `init<DBSID>.sap` parameter `compress` to `hardware`. Enter the correct address for tape devices with hardware compression in the parameters `tape_address` and `tape_address_rew` (for example, a lower-case `c` is important).



- Device type:
`backup_dev_type = tape|pipe`

Tape Devices with Hardware Compression

- Addresses for the tape device:
tape_address = (/dev/rmt/0hnc)
tape_address_rew = (/dev/rmt/0hc)
- Compression parameters:
compress = hardware
- Tape size:
tape_size = 1600M

Software Compression

Use software compression only if you have no tape devices with hardware compression. Using hardware and software compression at the same time does **not** improve compression rates.

- **Advantage** of software compression:
If you make remote backups over a network, using software compression will significantly reduce the network load.
- **Disadvantage** of software compression:
 - high CPU utilization
 - lengthy backups due to compression process



- Device type:
`backup_dev_type = tape|pipe`
- Tape device addresses:
`tape_address = (/dev/rmt/0mn)`
`tape_address_rew = (/dev/rmt/0m)`
- Compression parameters:
`compress = yes`
- Tape size:
`tape_size = 1800M`

Software Compression and Remote Backup to Parallel Tape Devices

You can use multiple tape devices when you use software compression or make a remote backup to a remote host. This means that unattended backup of large databases is possible even if you do not have a tape device with hardware compression or want to perform remote backups.

If several tapes are required for a backup with software compression or for a remote backup, the existing tape devices are used in parallel if the number of parallel copy processes was not reduced by changing the parameter [exec_parallel \[Page 331\]](#). The tape devices must be defined in parameters [tape_address \[Page 347\]](#) and [tape_address_rew \[Page 351\]](#).

Size of the Compression Directory.

When software compression is used (`compress = yes`, backup not on disk), BRBACKUP uses the compression rates to determine the space required in the compression directory. The free space must be at least as large as the largest compressed file. The calculated compression rates are stored in a detail log and in the database table SDBAD (see [BRBACKUP, BRARCHIVE and BRRESTORE Logs \[Page 434\]](#)).



When BRBACKUP is started for the first time, compression rates are not available. In this case BRBACKUP uses internal default values that are usually smaller than the actual compression rates. Successful compression is then ensured only if the compression directory has at least as much free space as the largest database file needs before the compression.

Software Compression

If you use `compress = only` (determine the compression rates), no disk space is required in the compression directory. The sizes are evaluated by reading the compressed files directly (using redirection). As a prerequisite for this, the redirection character ">" must be used in the parameter `compress_cmd` (as set already by default).

Backup to Multiple Disks

You can use BRBACKUP for backup to multiple disks if the space available on one disk or logical volume is not sufficient.

To do this, you have to specify the directories (on the different disks) to which you want to save your database files in the `init<DBSID>.sap` profile parameter `backup_root_dir`.



- Device type:
`backup_dev_type = disk`
- Backup directories:
`backup_root_dir = (/backup/dir1, /backup/dir2)`
- Compression parameters:
`compress = no|yes`

BRBACKUP normally uses all the directories specified in `backup_root_dir` in parallel (the number of copy processes corresponds to the number of disks) in order to make a backup of the database files. Since BRBACKUP attempts to optimize the speed of the backup, all the hard disks specified in `backup_root_dir` are written to (except if the number of files you want to back up is smaller than the number of disks). This setting can be changed with the `init<DBSID>.sap` parameter [exec_parallel \[Page 331\]](#) or the command option `-e|-execute`.

Operating system utilities can then be used to copy a disk backup to tape.

Backup to a Remote Disk.

Backup to a Remote Disk.

Use

You can make a backup to a remote disk directly with BRBACKUP/BRARCHIVE, or together with an incremental backup with the Oracle Recovery Manager (RMAN).



Backup to the disk of a remote host only appropriate when the network is very fast and stable. This means that this procedure cannot be recommended for production systems in most cases.

Prerequisites

The following prerequisites must be satisfied for a backup to remote disk:

- The target directory for the backup defined in the parameter [archive_stage_dir \[Page 313\]](#) must have been created on the remote host.
- The related subdirectory `<archive_stage_dir>/<SID>` must have been created on the remote host. This is the target directory for backing up the profiles and logs.
- The name of the remote host and the relevant user must be specified in the initialization profile with the parameters [remote_host \[Page 338\]](#) and [remote_user \[Page 339\]](#).
- If you use FTP as the transmission program, a password is needed for the remote host. You have two options here:
 - 1.) Specification of the password in the parameter [remote_user \[Page 339\]](#).


```
remote_user="<user_name> <password>"
```
 - 2.) Password is not specified explicitly: In this case, the password of the BRBACKUP database user is used automatically. You must make sure that the password of the operating system user is the same.
- No password is needed if you use *RCP* for the remote disk backup. A prerequisite for a successful remote SHELL call is for example the following entry in the file `.rhosts`, which is in the HOME directory of the remote UNIX user on the remote host:

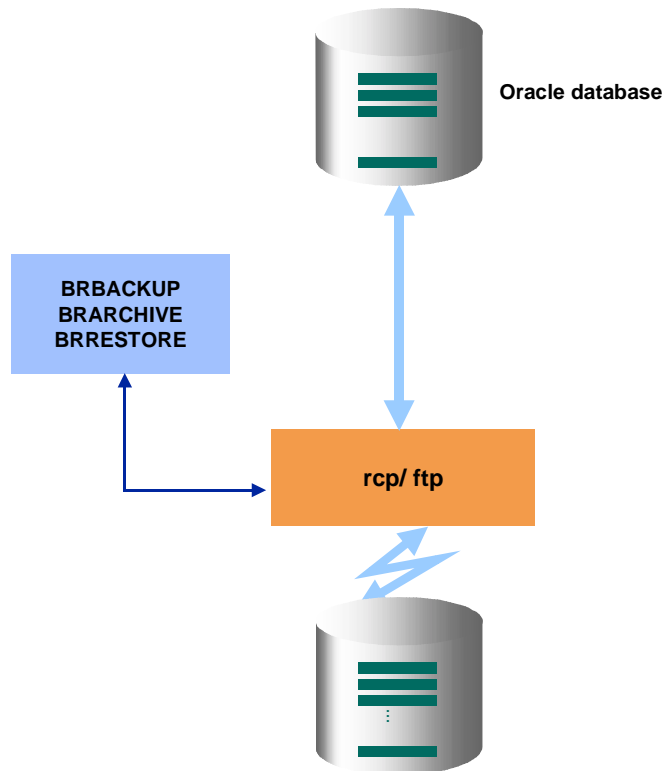

```
copy_in_cmd = "dd bs=16k if=$"
copy_out_cmd = "dd bs=16k of=$"
```

Remote Disk Backup with BRBACKUP

The remote backup to disk removes the need for a NFS disk mount, if the backup is to made through the network. The NFS protocol is relatively insecure. This means that the backup must be verified. In contrast, *RCP* and *FTP* are relatively secure ways of transferring data over the network. This means that the backup does not have to be verified.

The following entries have to be made in the initialization profile `init<DBSID>.sap`:

```
backup_dev_type = stage|stage_copy|stage_standby
stage_copy_cmd = rcp|ftp
```

A remote backup to disk is particularly useful in connection with the standby database configuration, and support for the AMASS software (Hierarchical Storage Management System of the company EMASS).

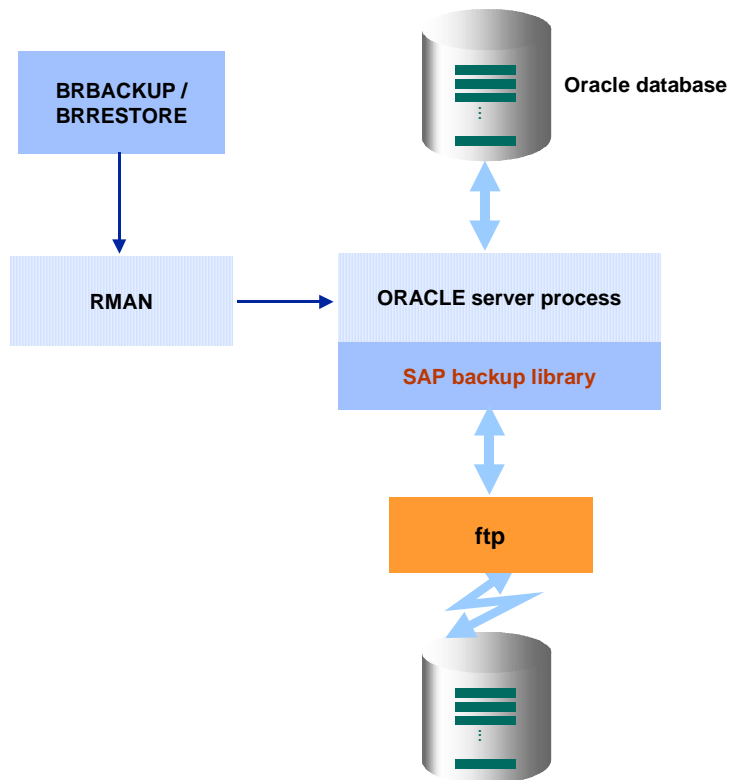
Incremental Remote Backup with RMAN

You can use RMAN to make an incremental backup to the disk of a remote host. As well as satisfying the aforementioned prerequisites, you also have to make the following entries in the initialization profile `init<DBSID>.sap`:

```
backup_dev_type = stage
backup_mode = incr
```

FTP is always used automatically as the transmission program.

Backup to a Remote Disk.

**See also:**

[Backups with the SAP Backup Library \[Page 160\]](#)

[Backups to Disk with the Recovery Manager \[Page 170\]](#)

Backup to a Remote Tape Device

BRBACKUP/BRARCHIVE can back up files to a remote tape device that is connected to a UNIX host in the network.

The UNIX versions of the local and remote hosts do not need to be identical. If, for example, the database runs on an HP-UX host, the backup can be performed on an AIX host.

The individual database files are transferred to the remote host via a remote shell call (via "pipe"). The remote host is defined in the `init<DBSID>.sap` parameter `remote_host`, and the appropriate user is defined in the parameter `remote_user`.

Make sure that no additional messages (that is, not belonging to command output) are issued on remote login (for example, from `.login` or `.cshrc`). Test the command `remsh <host_name> date`: only one line with the output of the `date` command should be displayed.

A prerequisite for a successful remote SHELL call is for example the following entry in the file `.rhosts`, which is in the HOME directory of the remote operating system user on the remote host:

```
<local_host_name> <local_user_name>
```

`<local_host_name>` is the host where the database runs.

`<local_user_name>` is the operating system user who starts the backup.

On the remote host, the files are written to tape using the UNIX `dd` command. The appropriate `dd` command is defined in the profile `init<DBSID>.sap`.



```
copy_in_cmd = "dd bs=16k if=$"
copy_out_cmd = "dd bs=16k of=$"
```

You can use several tape devices on the remote host for backup. These are used in parallel.



Backup on a remote tape device is only appropriate when the network is very fast and stable. Therefore, this procedure is usually not recommended for production systems. Test databases, on the other hand, are entirely suitable for remote backups, for example, backing up the test database remotely using backup devices connected to the host where the production database runs.

The number of parallel copy processes normally corresponds to the number of backup devices available. This setting can be changed with the `init<DBSID>.sap` parameter [exec_parallel](#) [Page 331] or the command option `-e|-execute`.

[Backup to a Remote Disk \[Page 120\]](#)

Two-Phase Backup

Two-Phase Backup

As an alternative to a direct backup to tape you can make a two-phase backup with the complete support of BRBACKUP. This backup strategy enables you to easily make a disk backup, as well as having several copies of this and previous backups available. For a recovery, SAPDBA can, in this case, access the disk backup directly. If a backup from tape is required then the files are directly copied to the original directories by BRRESTORE.



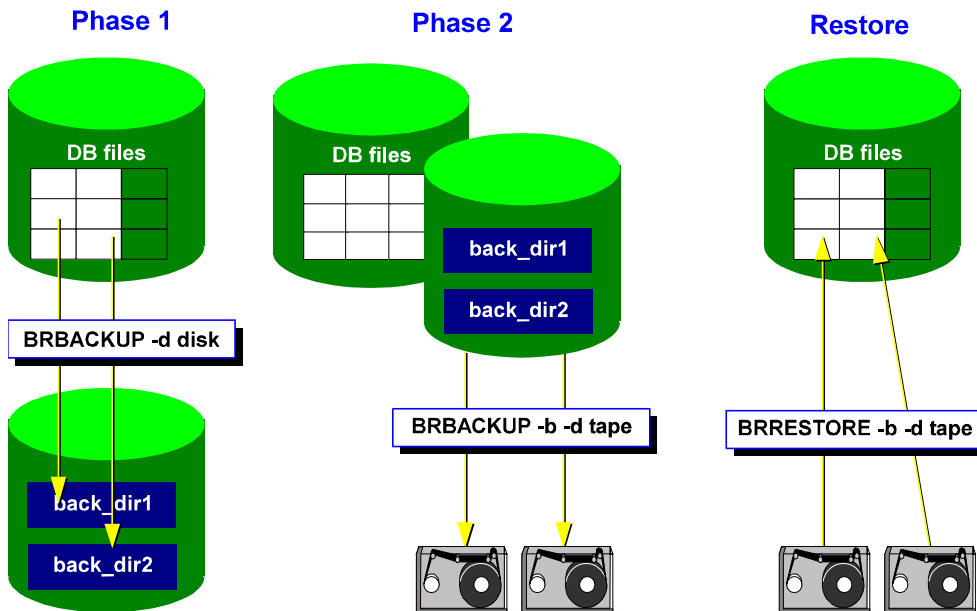
You can also carry out the second phase of the two-phase backup with external means (backup programs, operating system utilities, and so on). The user then has full responsibility for the complete and correct execution of this phase.



Backups to raw devices cannot be backed up to tape in a second phase with the BACKINT interface. In this case save directly to tape.

BRBACKUP: Two-Phase Backup

- **Phase 1**
BRBACKUP backup to disk.
- **Phase 2**
BRBACKUP backup from disk to tape
- **Restore phase:**
BRRESTORE restoration from tape directly to the original directories or BRRESTORE from the disk to the original directories.



To make the backup to tape (phase 2) start BRBACKUP with the relevant command option (see [-b backup \[Page 367\]](#)).

Advantages

- The first phase can be much shorter than a direct backup to tapes.
- In the case of a recovery the restoration phase is shortened, if the backup is directly available on the disk.
- When restoring from tape, BBRESTORE can write the backed up files directly to the original directories.
- Volume management and all other automatic actions of BRBACKUP can be fully utilized in the second phase.

Disadvantages

- The hardware requirements (disk storage and storage tapes) must be fulfilled. Additional disk storage space is required, in comparison to a direct backup to tape.
- BRBACKUP has to be started twice for a two-phase backup.



The backup type (offline, online) and the extent of the backup (full, partial) must be identical in the first and second phases. In an offline backup the database remains open in the second phase.

BRARCHIVE: Two-Phase Backup

Two-phase backup of the offline redo log files using BRARCHIVE runs analog to the BRBACKUP backup. You cannot, however, make any number of copies to tape (maximum of two). We thoroughly recommend a backup of the offline redo log files to tape, in addition to a disk backup.

- **Phase 1**
BRARCHIVE backup to disk
- **Phase 2**
BRARCHIVE backup from disk to tape
- **Restore phase:**
BBRESTORE restoration from tape directly into the original directories or direct import of offline redo log files in an SAPDBA recovery from disk (no restoration).

To make the BRARCHIVE backup of the offline redo log files to a volume start BRARCHIVE with the appropriate command option (see [-a|-archive \[Page 233\]](#)).

Advantages

- The first phase can be much shorter than direct backup to the volume, therefore the SAPARCH directory is emptied more quickly.
- In a recovery situation, the restoration phase can be much shorter, if the backup of the offline redo log files is directly available on disk. In this case SAPDBA uses the offline redo log files directly from the disk. There is no restoration.
- When restoring from tape BBRESTORE can write the backed up offline redo log files directly to the SAPARCH directories.
- The volume management of BRARCHIVE can be fully utilized in the second phase of the backup.

Two-Phase Backup

Disadvantages

- A maximum of two copies of the offline redo log files can be backed up to tape, regardless of the way in which the offline redo log files are backed up: directly to tape or with a disk backup.
- Additional storage space and storage tapes are required.
- BRARCHIVE must be started twice for a two-phase backup.

Online Mode and Offline Mode

Online Mode

Purpose

You can perform an online backup with the database running - that is, the users can continue to work normally. The management of database changes by the corresponding Oracle background processes is not affected either.

Tablespace online backups are inconsistent. The database can only be made consistent in a recovery if redo log entries from the period between *BEGIN BACKUP* and *END BACKUP* exist and can be imported.

Prerequisites

An online backup is made using operating system utilities (for example, under the control of the SAP utility BRBACKUP). Because these utilities are not part of the database system, you must be informed about the starting point of a backup. In this way, a unique restart point is defined from which the recreation of all the files of a tablespace can be carried out in case of an error.

Procedure

- The starting point of the backup is set using the command *ALTER TABLESPACE <tablespace name> BEGIN BACKUP*. The header of the tablespace files holds information on the checkpoint and redo log files. When the next redo log file switch or checkpoint occurs (normal database operations continue), the header information remains unchanged.
- Based on this mechanism, all the files of a tablespace are copied with uniform header information specifying when the backup was started.
- Once the backup of the tablespace is complete, the command *ALTER TABLESPACE <tablespace name> END BACKUP* ensures that the header information of the files is updated.

Offline Mode

Purpose

After an offline backup of the complete database, you have a backup of the database that is consistent. If you work with the database after the backup, the backup is consistent, but not up-to-date. In this case, you have to recover the database after you restore the backup.

Prerequisites

The database is closed during an offline backup; because the database cannot be accessed you cannot work with the R/3 System during an offline backup. However, the R/3 System does not have to be shut down for an offline backup. If the *RECONNECT* mechanism is set in the SAP start profiles, the connection to the R/3 System is remade after the database is restarted; this means that the information in the buffers of the R/3 System is not lost.

Consistent Online Backup

Consistent Online Backup

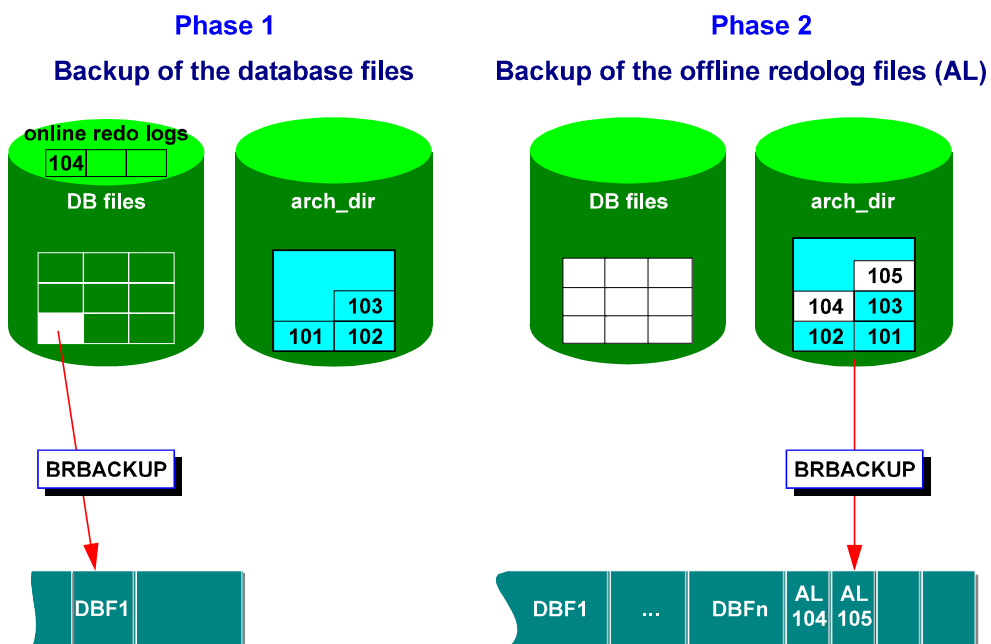
Definition

A consistent online backup (`backup_type = online_cons`) is an alternative to an offline backup when you cannot close the database. This backup has logically consistent data because the offline redo log files created by BRBACKUP during a backup are backed up with the database files on the same volume. The backup of the offline redo log files using BRBACKUP runs completely independently from the BRARCHIVE backups and does not influence them in any way. A consistent online backup differs from an offline backup in that a recovery of the database is always necessary in order to guarantee consistent data.

Use

Any recovery of the database is made easier since the offline redo log files of the same backup can be imported. In the same way the database can be reset to an earlier status (*Reset database*) (only when using a BRBACKUP online backup).

Structure



Activities

To make a consistent online backup, use the BRBACKUP option `-t|-type` [\[Page 386\]](#) `online_cons` or the relevant `init<DBSID>.sap` profile parameter `backup_type` [\[Page 319\]](#) = `online_cons`.

To restore the offline redo log files from the BRBACKUP backups, use the BRRESTORE option `-m|-mode` [\[Page 427\]](#) `archive_logs`. Restore a complete BRBACKUP backup using the BRRESTORE option `-m full`.

[Command Options for BRBACKUP, BRARCHIVE, and BRRESTORE \[Page 363\]](#)

Structure-Retaining Database Copy

Structure-Retaining Database Copy

Use

With BRBACKUP you can make a copy of the database files which has exactly the same directory structure. You can use this type of database to

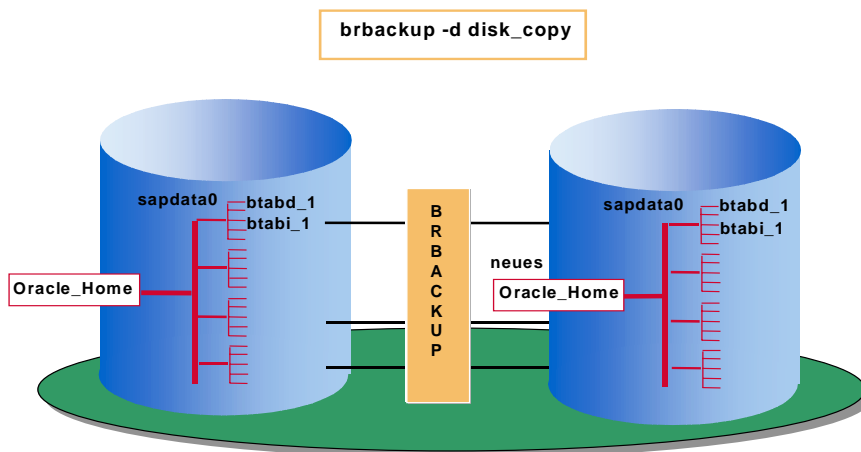
- generate a test system from a production system
- Set up a [Standby Database Scenario \[Page 185\]](#).
- have a database backup available which saves you the restoration process during a recovery. In this case the Oracle Home directory will be renamed as the new Oracle Home directory of the database copy (set link). The copied files are then the current files and the offline redo log files can be imported directly.
- change the location of the database files (file system or raw device). Database copy is the only way of moving database files from one file system to raw devices (or vice versa) by means of BRBACKUP.

Prerequisites

The following directories must be created on the target database:

- sapdata directories
- sapbackup directory
- origlogA, origlogB, mirrlogA, mirrlogB directories of the online redo log files

The corresponding subdirectories are created automatically during copying.



Structure-Retaining Database Copy



```
/oracle/c11/sapdata2/stabd_1/stabd.data1 is copied to  
/oracle/c12/sapdata2/stabd_1/stabd.data1
```



Since this type of copy is a 1:1 copy no software compression may take place.

To make the copy of the database you have to define the name of the new Database_Home directory (of the database copy) in the `init<SID>.sap` profile parameter [new_db_home \[Page 335\]](#). In addition to this set the parameter [backup_dev_type \[Page 314\]](#) to `disk_copy` or call up BRBACKUP with the relevant command option `brbackup -d|-device disk_copy`.



Under Windows NT, the `sapdata` directories can be distributed across several drives. When you make the copy, you can retain this distribution by specifying the appropriate target drives (see `brbackup m|-mode [Page 378]`).

Backing up Executable Programs and Other Components of the R/3 System

Backing up Executable Programs and Other Components of the R/3 System

In addition to the database files and offline redo log files, the following non-database files should also be backed up:

- **Constant Files**
You can find these files, for example, in the subdirectories `/usr/sap/<SAPSID>/SYS` and `<ORACLE_HOME>/dbs`. They include executable programs and profiles of the R/3 System and of the Oracle database system. Back up the SAP directories after an R/3 upgrade and the Oracle directories after a database migration.
- **Temporary Files**
You can find these files, for example, in the subdirectory `/usr/sap/<SAPSID>/<INSTANCE>`. The loss of these files is not that critical, and will not cause data inconsistency. SAP provides tools that can reset the references to these files in the database, when required.

You can use the BRBACKUP program to back up non-database files as well (see [Backing Up Non-Database Files and Directories \[Page 228\]](#)). However, you should only use it for exceptions. Backing up non-database files using BRBACKUP is not a replacement for backing up the file system using operating system means.

Backing Up Data Tablespaces

Especially with large databases it can be important to keep the volume of data to be backed up or restored as small as possible. Index definitions are stored in the SYSTEM tablespace as a matter of course. Therefore indexes can always be recreated. You do not have to back up pure index tablespaces during a database backup.

To make a backup of all tablespaces except for pure index tablespaces start BRBACKUP with the command option `-m all_data` ([m\]-mode \[Page 378\]](#)) or set the relevant `init<DBSID>sap` profile parameter [backup_mode \[Page 316\]](#) = `all_data`.

During the backup BRBACKUP indicates all pure index tablespaces, so that BRRESTORE recognizes these and the restore process can also be carried out from a complete backup (`-m all`) excluding the pure index tablespaces: `brrestore -m all_data` ([-m\]-mode \[Page 427\]](#)).



After restoring the backed up data tablespaces both the index tablespaces and the indexes must be newly created with the definitions in the SYSTEM tablespace. This action is automatically carried out by SAPDBA during the recovery.

[Backup of Large Oracle Databases \[Page 173\]](#)

[SAPDBA Database Recovery \[Page 697\]](#)

Duration of the Backup

Duration of the Backup

Backups often take a great deal of time. For example, if you use DAT tape devices as a backup medium, you can back up 500-1000 MB per hour (without hardware compression) or 1000-2000 MB per hour (with hardware compression). This means that backing up a 10 GB database on one tape device of this type would take approximately 10 hours.

Parallel Backup

You can solve this problem by performing data backups in parallel. If you have several backup devices, you can make a parallel backup with BRBACKUP by assigning several tape devices to the `init<DBSID>.sap` parameters `tape_address` and `tape_address_rewind` or several disks to parameter `backup_root_dir`. You can then make a backup of, for example, a 20 GByte database in parallel on 5 tape devices in 5 hours.

[Parallel Backup \[Page 135\]](#)

Logical Volumes

If you are using the logical volume manager, a logical volume is considered to be one disk. Therefore, you should not distribute a logical volume over several physical disks when this is not absolutely essential. This increases the effectiveness of the BRBACKUP minimization of read/write head movement on the disk.

Backup on disk

One interesting way to reduce the backup time is to back up to disk (`backup_dev_type = disk`). These backups can then be copied from disk to a volume (tape, for example), using UNIX utilities (`tar`, `cpio`). This later copy operation could be treated with lower priority

[Two-Phase Backup \[Page 124\]](#)

[tape_copy_cmd \[Page 353\]](#).

Parallel Backup

You can back up your database to several backup devices (tape devices/disks) in parallel. Parallel backup reduces the backup time and enables unattended operation (backup in unattended mode).

Parallel backup is possible to local or remote backup devices or to hard disks. If the backup devices used support hardware compression use this by setting the `init<DBSID>.sap` parameter `compress = hardware`. You can also make parallel backups with software compression (`compress = yes`).

When you perform parallel backups to several backup devices, BRBACKUP attempts to optimize the distribution of the database files among the backup volumes (load balancing).

- BRBACKUP attempts to optimize the activities to balance the load equally among all the backup devices. If this is not possible, it attempts to divide the dataset equally among the individual backup devices.
- Whenever possible, files from one disk are saved on one volume in one backup device, in order to minimize drive head movement during the backup.

BRARCHIVE only uses the parallel backup option to tape when you start archiving with `brarchive -ss` or `brarchive -ssd`. In this case, the offline redo log files for archiving are saved to both volumes in parallel (or saved and then deleted).

The addresses of the tape devices are defined in the `init<DBSID>.sap` parameters `tape_address` and `tape_address_rew` (or in `tape_address_arch` and `tape_address_rew_arch`). If the `-ss` or `-ssd` option is used, BRARCHIVE only uses the first two tape devices in the list. The addresses of the directories on disk are defined in the `init<DBSID>.sap` parameter `backup_root_dir`.



- Device type:
`backup_dev_type = tape|disk|pipe`
- Addresses for tape devices
`tape_address = (/dev/rmt/0hnc, /dev/rmt/1hnc)`
`tape_address_rew = (/dev/rmt/0hc, /dev/rmt/1hc)`
- Addresses for directories on hard disk:
`backup_root_dir = (/backup/dir1, /backup/dir2)`
- Compression parameters:
`compress = no|software|hardware`
- Tape size:
`tape_size = 1800M|1600M`

The number of parallel copy processes normally corresponds to the number of backup devices available (tape devices, disks). This setting can be changed with the `init<DBSID>.sap` parameter [exec_parallel \[Page 331\]](#) or the command option `-e | -execute`.

BRBACKUP and BRARCHIVE Backups in One Run**BRBACKUP and BRARCHIVE Backups in One Run**

You can execute the database backup using BRBACKUP and the backup of the offline redo log files using BRARCHIVE in one run. This option is intended to make more effective use of the increasing capacity of the storage tapes (DLT2000: up to 20 GB, DLT4000: up to 50 GB, DLT7000: up to 100 GB). It also makes an unattended backup easier, since after a backup with BRBACKUP the BRARCHIVE run does not have to be restarted as well. As of Release 4.5A, you can use this linkage of the two processes for any backup media, including backups with BACKINT. In the CCMS transaction DB13, you can now make the backup "*Database + archive log backup*" with BACKINT, or to a disk.

There are two options for making the backup in one run:

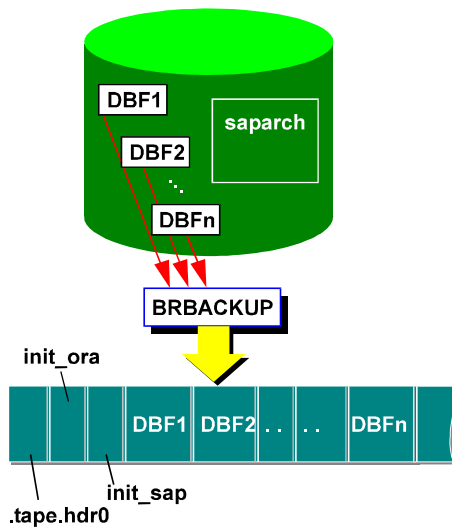
- BRBACKUP starts BRARCHIVE (`brbackup -a`). If the backup is made to tape, the tapes are managed by BRBACKUP.
- BRARCHIVE starts BRBACKUP (`brarchive -b`). If the backup is made to tape, the tapes are managed by BRARCHIVE.

We recommend the first option (`brbackup -a`). If the backup is made to tape, BRBACKUP uses the tapes defined in [volume backup \[Page 362\]](#) and makes the backup in the following steps:

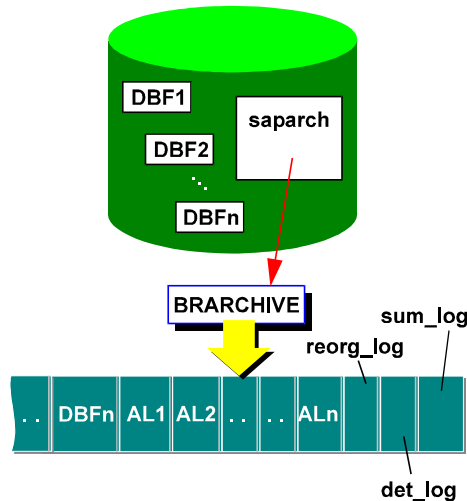
- Check the volume label
 - Backup of the tape header files (`.tape.hdr0, init_ora, init_sap`)
 - Backup of the database files (no logs are saved)
 - Call BRARCHIVE
 - BRARCHIVE then writes the offline redo log files to tape after the backed up database files (without checking the label and without header files).
 - BRARCHIVE backs up all logs (BRBACKUP and BRARCHIVE).

BRBACKUP and BRARCHIVE Backups in One Run

1.) Backup of the database files `brbackup -a`



2.) Backup of the offline redolog files (AL)



Overall view:



Unattended online backup with two tape devices. The database files are backed up first and then the offline redo log files are copied to the same tapes.

```
brbackup -m all -t online -c -a -ssd -c
```

See: BRBACKUP command mode: [brbackup -a|-archive \[Page 366\]](#)

If you run the tape administration under control of BRARCHIVE (`brarchive -b`) the tapes defined in [volume_archive \[Page 361\]](#) are used. The only change to the backup is the sequence in which the data files and the offline redo log files are written to tape. In this case the logs are backed up by BRBACKUP at the end.



Offline backup with one tape device. The offline redo log files are backed up on tape first and then the database files are copied to the same tape.

```
brarchive -sd -c -b -t offline -c
```

See: BRARCHIVE command mode: [brarchive -b|-backup \[Page 393\]](#)

Making a BRBACKUP/BRARCHIVE backup in one run can also be used for a BRBACKUP/BRARCHIVE backup to several tape devices in parallel.

BRBACKUP and BRARCHIVE Backups in One Run

See also: [Parallel Backup \[Page 135\]](#)



Starting BRBACKUP and BRARCHIVE in sequence with BACKINT, or to disk, only joins BRBACKUP and BRARCHIVE, it does not automatically mean that the backup is made to the same volume. If you are using BACKINT, you can do this with BACKINT tools in some situations.

Unattended Backup

Under certain conditions you can make a backup without monitoring or operator intervention. Note that the details are highly specific to the operating system.

Note the operating system-specific options for scheduling background jobs. See also, for example, [BRBACKUP/BRARCHIVE \[Page 261\]](#) with tips for Windows NT.

Parallel backup

You can make unattended backups if you have enough backup devices. This means that you need $\langle n \rangle$ backup devices if $\langle n \rangle$ volumes (for example, tapes) are required for the backup. BRBACKUP can then back up to these devices in parallel, without the operator having to change the volumes.

[Parallel Backup \[Page 135\]](#)

Backup with Automatic Tape Changers

If you want to use autochangers, you need to define the `rewind_offline` parameter appropriately and set `backup_dev_type` to `tape_auto` or `pipe_auto`.

[Automatic Tape Changing \[Page 222\]](#)

Serial Backup

Unattended backup is also possible if fewer parallel copy processes than connected backup devices are available. To do this, change the parameter [exec_parallel \[Page 331\]](#). If you need several volumes for a backup, the backup devices are not used in parallel. In this case they are used in accordance with the number of copy processes set.

In order to be able to make a parallel or serial backup on several tape devices, the addresses of the backup devices must be defined in the `init<DBSID>.sap` parameters `tape_address` and `tape_address_rewind` (or `backup_root_dir` in the case of a disk backup).



```
tape_address = (dev/rmt/0mn, /dev/rmt/1mn)
tape_address_rew = (dev/rmt/0m, /dev/rmt/1m)
```

BRBACKUP/BRARCHIVE Backups in One Run

The complete backup of the database files and the offline redo log files can be executed with a single start of BRBACKUP (command option [-a|-archive \[Page 366\]](#)).

[BRBACKUP and BRARCHIVE Backups in One Run \[Page 136\]](#)

Backup with CRON

For a successful unattended backup using *CRON*, the following requirements must be met:

- The parameters in `init<DBSID>.sap` must be set correctly.
- The *Crontab* entries must be defined under user **root**.
- BRBACKUP/BRARCHIVE must run under database user **ora<sapsid>** or the operating system user `<sid>adm`. The advantage of the `<sid>adm` user is that no password is needed, due to the OPS\$ mechanism. This means that no password can be seen in a script.

Unattended Backup

- Enough volumes (for example, tapes, disks) must be available, as well as a sufficient number of backup devices (if you want to make a parallel backup).
- The correct tapes must be mounted in the backup devices; the operator will then not have to change the tapes.

If automatic tape management is active, first determine the tape names by entering the commands `brbackup|brarchive -q`.

Use `brbackup|brarchive -q check` to verify that the required volumes were actually mounted.

If automatic tape management is not active, mount tapes whose expiration period has expired.



Online backup of the complete database
unattended operation
daily Monday through Friday
backup to be started at 10.00 p.m.

```
#Min(0-59) Hrs (0-23) Day (1-31) Mon(1-12) WD (0-Sun,...,6-Sat)
00 22 * * 1-5
su - ora<sapsid> -c "brbackup -t online -c force -u"%system/<password>
or under <sid>adm:
su - <sapsid>adm -c "brbackup -t online -c force -u /"
```



Offline backup of the complete database
unattended operation
daily Monday through Friday
backup to be started at 10.00 p.m
R/3 System shut down

```
#Min(0-59) Hrs (0-23) Day (1-31) Mon(1-12) WD (0-Sun,...,6-Sat)
00 22 * * 1-5
/backup1.sh
```

The script `backup1.sh` in the root directory could have the following content:

```
su - <sapsid>adm -c "stopsap R3"
su - ora<sapsid> -c "brbackup -t offline -c force -u" <<END
system/<password>
END
su - <sapsid>adm -c startsap
```



Offline backup of the complete database
unattended operation

Unattended Backup

daily; Monday through Friday
backup to be started at 10.00 p.m
R/3 System shut down

```
#Min(0-59) Hrs (0-23) Day (1-31) Mon(1-12) WD (0-Sun,...,6-Sat)
```

```
00 22 * * 1-5
```

```
/backup2.sh
```

The script backup2.sh in the root directory could have the following content:

```
su - ora<sapsid> -c "brbackup -t offline_force -c force -u" <<END
```

```
system/<password>
```

```
END
```

```
su - <sapsid>adm -c "saplicense -check"
```



Archiving the offline redo log files
unattended operation
daily; Monday through Friday
backup to be started at 8.00 a.m.
parallel archiving to two backup devices.

```
#Min(0-59) Hrs (0-23) Day (1-31) Mon(1-12) WD (0-Sun,...,6-Sat)
```

```
00 8 * * 1-5
```

```
su - ora<sapsid> -c "brarchive -ssd -c force -u"%system/<password>
```

Volume Management

Volume Management

The volumes (tapes) are overwritten again at each backup with BRBACKUP/BRARCHIVE. BRBACKUP/BRARCHIVE never use the space remaining on the tapes after a backup. New tapes have to be inserted each time you make a backup.

BRBACKUP/BRARCHIVE can only use volumes that are initialized accordingly ([Initializing Volumes \[Page 143\]](#)).

The backup volumes must be managed to make sure that they are protected from premature deletion, ensuring that you have access to the required volumes at all times (see [Volume Expiration Period \[Page 147\]](#)).

You can choose volumes in three ways:

- BRBACKUP/BRARCHIVE offer [Automatic Volume Management \[Page 153\]](#).
- The operator can select the volume.
- You can also select the volumes using external tools.

Volume Initialization

Volume (tape) initialization refers to the writing of a label (file with the name **.tape.hdr0**) to the volume concerned. This label file is read when the volume is checked. If this label file does not exist, the check fails and the volume is rejected.

You only have to initialize new tapes or tapes that were not used by BRBACKUP/BRARCHIVE. Only initialize the volumes once, not repeatedly before every backup. You do have to reinitialize a volume, however, if want to change the name of that volume. You cannot change volume names during a backup.

Several devices can be used for initializing volumes. The backup devices defined in the `init<DBSID>.sap` profile in parameters `tape_address` and `tape_address_rew` (or in `tape_address_arch` and in `tape_address_rew_arch`) (see [tape_address \[Page 347\]](#) and [tape_address_rew \[Page 351\]](#)) are used serially during a volume initialization. The volumes of all the available backup devices can then be changed at the same time.

Your volumes should also have paper labels in order to make the volume name immediately visible.

The information in the label is overwritten or ignored if you write to a BRBACKUP/BRARCHIVE volume directly using a tool which is not from SAP.

Procedure

Volumes are initialized using the programs BRBACKUP and BRARCHIVE.

All volumes to be used by BRBACKUP/BRARCHIVE for the first time must be initialized. If you want to use automatic volume management, the volume names must be listed in the `init<DBSID>.sap` parameters `volume_backup/volume_archive` (see [Automatic Tape Management \[Page 153\]](#)).

You can use the following options during an initialization:

- **-i|-initialize**
Rename volumes that have already been initialized. Only possible for volumes with an expired expiration period.
- **-i force|-initialize force**
Initialize new volumes or volumes not yet used by BRBACKUP/BRARCHIVE. The expiration period check is not active. Important: This option can also be used to reinitialize locked volumes, which you should never do. If you initialize BRBACKUP/BRARCHIVE tapes with the addition `force`, the `tape_use_count` stored in the tape label is set to 1. Otherwise this value is increased accordingly.
- **-v|-volume**
Enter the names of volumes for initialization.
- **-n|-number**
Number of volumes for initialization.

Procedure



Initialization of volumes with the volume names specified in `volume_backup/volume_archive`:

Use the BRBACKUP/BRARCHIVE option `-i [force]`. Mount a volume and enter the following command:

```
brbackup|brarchive -i [force]
```

All the volumes specified in the parameters `volume_backup/volume_archive` will be initialized in sequence.

New and non-BRBACKUP/BRARCHIVE volumes must always be initialized with the additional specification `force`.

If you initialize a volume without this additional specification, the expiration period of the volume is checked. Locked volumes or volumes without labels are rejected; volumes that are not locked are renamed.



Initialization of `x` volumes with the first `x` volume names specified in `volume_backup/volume_archive`:

```
brbackup|brarchive -i [force] -n x
```



Initialization of volumes with a specific name:

```
brbackup|brarchive -i [force] -v <name1>[,<name2>]...
```

All the volumes for initialization must be mounted in the backup device in the proper order.

Volume Label Check

The volume label is always the first file on a volume. This file has a format specific to BRBACKUP/BRARCHIVE and is written to the volume using `cpio`.

This label file has the name: `.tape.hdr0`.

You can display the contents of a volume label using the call `brbackup|brarchive -i show -n 1` The label includes:

- the volume name (for example, tape name)
- Timestamp: when the volume was used for a backup (start timestamp of the corresponding backup with BRBACKUP or BRARCHIVE)
- Name of the database instance (ORACLE_SID)
- BRBACKUP/BRARCHIVE actions ID (coded timestamp of the BRBACKUP/BRARCHIVE log names)
- BRBACKUP/BRARCHIVE function ID (extension of the detail log file name of BRBACKUP/BRARCHIVE)
- Volume use count: how often the volume has been used

Label Check Before a Backup

Before BRBACKUP/BRARCHIVE writes to a volume, it reads the volume label file. If this file does not exist, you have to either initialize the volume or mount another volume.

The following volume label information is checked:

- **Volume name**
An error message (message number BR216X) is issued if you have mounted a volume with an incorrect name. If a scratch volume was requested, you can mount any volume.
- **Expiration period**
An error message (message number BR217X) is issued if the configured expiration period (number of the days specified in the `init<DBSID>.sap` parameter `expir_period` that must have passed before the volume can be used again) has not ended yet.
- **Volume use count**
A warning message (message BR235W) is issued if the volume has been overwritten more frequently than specified in the `init<DBSID>.sap` parameter `tape_use_count`.

Label Check After a Backup

After a backup to a volume is completed, the volume label is checked once more. This is to detect volume, volume device, driver, or hardware errors that would prevent a successful backup, but would not cause an error message to be issued. In extreme cases, several sequential backups could be unusable if these systematic errors are not recognized. The system checks whether the name of the database instance, the action ID, and the function ID match the current values.

Scratch Volumes

Scratch Volumes

When BRBACKUP/BRARCHIVE requests a scratch volume for backup, this means that any volume whose expiration period has ended can be used. It does not mean that you have to mount a volume with the name SCRATCH.

You can also initialize a volume with the name SCRATCH explicitly (for example, `brbackup -i -v SCRATCH`). Such a volume is always accepted, even when a volume with another name is requested. The program still makes sure that the expiration period has expired. In this case the tape mounted is assigned the name of the requested tape. Only use this option in exceptional cases, for example, when an additional tape is requested unexpectedly during a backup operation.

A backup on a tape named SCRATCH should never exist. This situation may occur when a scratch tape is requested and a tape named SCRATCH is mounted. The old backup would be overwritten in this case.

Volume Expiration Period

The volume expiration period is defined in days with the `init<DBSID>.sap` parameter `expir_period` (see [expir_period \[Page 332\]](#)).



`expir_period = 2` means that writing to a volume (tape) is possible on the second day after the volume has been mounted and used. If you use a volume on Monday, for example, you cannot use it for another backup until Wednesday.

The start time of BRBACKUP/BRARCHIVE determines the first day of the lock for all volumes used for a backup. The time when the volume was initialized does not matter.

The expiration period always expires at midnight (12:00 a.m.) of the last day of the lock.



If you set a expiration period of 0 days, this means that the volume is not locked. The volumes can be overwritten on the same day. Therefore, **do not set** `expir_period` **to zero**.

SAP recommends a expiration period of at least 14 days, 28 days would be even better (the default value is 30 days).

The current value of the parameter `expir_period` is decisive for whether or not a volume is locked, not the value of the parameter during the backup. This means that the backup volumes are locked for `n` days after the last backup operation, `n` being the current value of `expir_period`. If the value of `expir_period` is changed, the expiration period for all volumes is automatically changed.

Volumes can be locked physically and logically. The volume generation date specified on the volume label is decisive for a **physical lock**. This generation date is determined when the volume label is written (when a backup on this volume was started). A volume is locked physically when the system checks the volume label and finds that the expiration period for the volume has not ended yet, i.e., the value of the current date is less than the total of the volume generation date stored in the volume label and the value of `expir_period`.

The internal information in the BRARCHIVE/BRBACKUP logs is decisive for a **logical lock**. The logs are updated when a database file has been backed up successfully. A volume is locked logically when the automatic volume management system checks the volume and finds that the expiration period stored internally has not ended yet; the value of the current date is less than the total of the volume generation date stored in the BRBACKUP/BRARCHIVE logs and the value of **expir_period**. Under certain circumstances, discrepancies may occur between the physical and logical locks.

Example 1

During a backup, the volume label was written to the volume but the backup was terminated before the first database file could be written to the volume.

This means that the volume is locked physically but not logically. It is selected from the list by the automatic volume management system (`volume_backup` or `volume_archive`) but is rejected

Volume Expiration Period

when the physical volume label check takes place. The volume can be reinitialized (with the same name) in order to cancel the physical lock.

- To do this, temporarily set the `init<DBSID>.sap` parameter `expir_period` to 0, to circumvent the physical lock.
- Then start BRBACKUP/BRARCHIVE, for example, by calling `brbackup|brarchive -i -v <volume name>`.
- Reset the parameter `expir_period` to its previous value.

When performing this operation, do not use the `-i force` option, as this would cause the volume use count stored in the volume label to be lost.

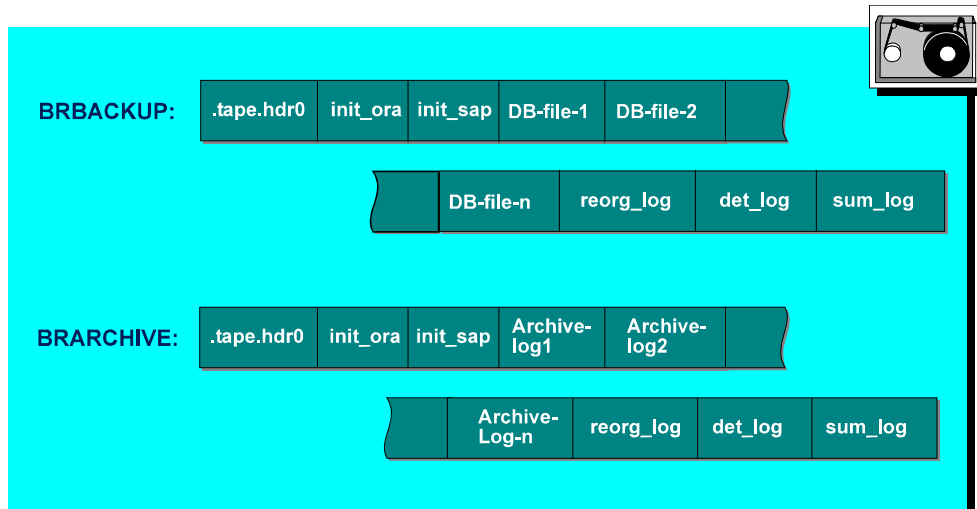
Example 2

A volume was reinitialized before the expiration period ended (for example, with the option `-i force`). This means that the volume is no longer locked physically; however, it is not selected by the automatic volume management system because it is still locked logically. If you want to use this volume nevertheless (before the logical lock has expired), you can switch off automatic volume management temporarily, for example, by calling `brbackup|brarchive -v SCRATCH` and mounting the volume on the backup device.

Used Volumes

BRBACKUP/BRARCHIVE volumes have the following files after a successful backup:

- Label: `tape.hdr0` (position 1)
- `init_ora`: Initialization files `init<DBSID>.ora`, `init<DBSID>.dba` (OPS: the `init.ora` profiles of all Oracle instances) (position 2)
- `init_sap`: Initialization file `init<DBSID>.sap` or the profile file defined under [-p|-profile \[Page 382\]](#) (position 3).
- Database files (data files, online redo log files, control file) and/or non-database files and/or non-database directories or archived redo log files
- `reorg_log`: Main SAPDBA log `reorg<DBSID>.log` and structure log `struct<DBSID>.log` (OPS: `script create_raw_links`) (third-to-last position)
- `det_log`: Detail log of BRBACKUP/BRARCHIVE (second-to-last position)
- `sum_log`: Summary log of BRBACKUP/BRARCHIVE (OPS: the summary BRARCHIVE logs of all the Oracle instances) (last position)



Information about which volumes were already used is contained in the BRBACKUP/BRARCHIVE logs.

BRBACKUP and BRARCHIVE write two types of logs:

- File system logs
 - detail log; recreated in each backup
 - summary log; extended in each backup
- Database logs
 - table SDBAH with information about the whole backup
 - table SDBAD with information about the backup of a file

These log files should only contain information that was written by BRBACKUP/BRARCHIVE (or possibly SAPDBA); do **not** change this information manually. The information in database tables SDBAH and SDBAD is evaluated and deleted by the Computing Center Management System

Used Volumes

(CCMS), for example. Therefore, this information may not be changed except by BRBACKUP/BRARCHIVE/SAPDBA and the CCMS.

[BRBACKUP, BRARCHIVE, and BRRESTORE Logs \[Page 434\]](#)

For [automatic volume management \[Page 153\]](#), BRBACKUP uses the corresponding database log as the foundation for selecting available volumes; BRARCHIVE uses the summary file system log. BRARCHIVE cannot rely on the database logs because the program can only be run when the database has been shut down.

Selecting Volumes Manually

Use

You can also select the volumes for a backup manually. You can do this if the BRBACKUP/BRARCHIVE volume management is deactivated by using the reserved volume name SCRATCH. The appropriate SAP program checks the expiration period. .

Prerequisites

1. Make sure that initialized volumes are available for the backup.
2. Determine whether the required expiration period has been configured in profile parameter `expir_period`; change the value when necessary.

Procedure

There are two ways to start the backup with BRBACKUP or BRARCHIVE:

- Enter the following parameter values in profile `init<DBSID>.sap`: `volume_backup = SCRATCH` or `volume_archive = SCRATCH`
- Then start BRBACKUP/BRARCHIVE.
- Leave the profile unchanged (a volume pool may be defined).
- Start BRBACKUP/BRARCHIVE with the option `-v SCRATCH`.

Result

The SAP programs BRBACKUP and BRARCHIVE request the number of scratch volumes needed for the backup (message BR104I), expired volumes with any name. The volume names in the labels are not changed by the backup operation. Any expired BRBACKUP/BRARCHIVE volumes are accepted. See [Volume Expiration Period \[Page 147\]](#).



If you use scratch volumes, it may make sense (primarily for reasons of clarity) to include the weekdays or days of the month in the volume names.

Selecting Volumes with External Tools

Selecting Volumes with External Tools

Use

You can also use external tools to determine the names of the volumes relevant for the backup. This could involve an external volume management system or simply a shell script. You can do this if you have deactivated BRBACKUP/BRARCHIVE volume management by calling the programs with the option `-v`. However, the expiration period of the volumes is checked anyway.

Prerequisites

The external tool that you use for volume selection must ensure that only non-locked volumes are suggested for backup. Otherwise, BRBACKUP/BRARCHIVE will terminate if they do not find enough free volumes. Make sure that initialized volumes are available for the backup.

Procedure

1. Determine whether the required expiration period has been configured in profile parameter `expir_period`; change the value when necessary.
2. Start the BRBACKUP/BRARCHIVE backup with the option `-v <volume list>`. Before starting the backup, BRBACKUP/BRARCHIVE checks whether the mounted volumes agree with those in the volume list and whether the expiration period has expired.

Result

BRBACKUP/BRARCHIVE only use the volumes listed with option `-v` for one backup.

Activities

You can select the volume names yourself by defining them in the call with the option `-v | -volume` (naming convention, for example, as suggested for automatic volume management).



```
brbackup -v C11B141,C11B142,C11B143
brarchive -ssd -v C11A141,C11A142
```

You can also give the volumes other names. One option would be to include the day of the backup in the volume names. BRBACKUP/BRARCHIVE can use the following sample script to assign the volume names to the day of the month on which the backup was started.

```
<DBSID><X><dd><n>
```

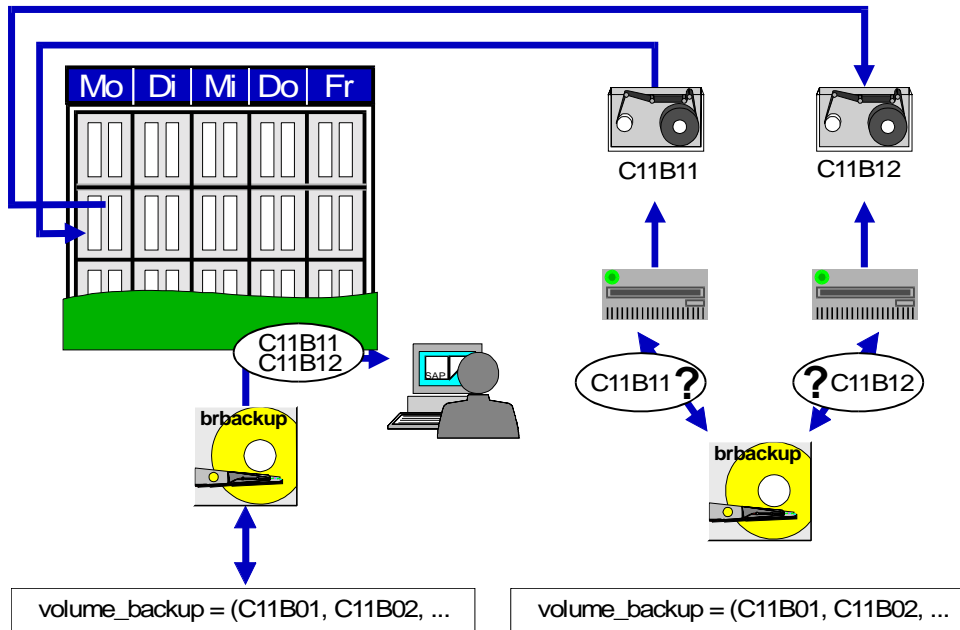
DBSID=ORACLE_SID (name of the database instance), **X**=A for BRARCHIVE or **X**=B for BRBACKUP, **dd**=day of month, **n**=next volume number within one backup.

```
day=`date | cut -f 3 -d " "`
if [ ${day} -le 9 ]; then day=`echo 0${day}`; fi
brbackup -v C11B${day}1,C11B${day}2,C11B${day}3 -c
brarchive -ssd -v C11A${day}1,C11A${day}2 -c
```


Automatic Volume Management

Purpose

BRBACKUP/BRARCHIVE automatically select the volumes that are recommended for the next backup. The name of the mounted volume is compared to the name found in the volume pool. In addition, the procedure ensures that current backups are not overwritten.



Prerequisites

When you use automatic volume management, do not use weekdays or days of the month in the volume names, because BRBACKUP/BRARCHIVE do not check whether a holiday fell in the period since the last backup or not. SAP recommends that you include the name of the database instance and a sequential number in the tape name, for example <DBSID>A<nn> for BRARCHIVE tapes, <DBSID>B<nn> for BRBACKUP tapes.

Procedure

1. Define a pool of the volumes available for the backup. To do this, enter the corresponding volume names in profile parameter `volume_backup` or `volume_archive`. All the volumes defined there should physically exist; initialize new volumes when necessary.
2. Determine whether the required expiration period has been configured in profile parameter `expir_period`; change the value when necessary.
3. Start the backup with BRBACKUP or BRARCHIVE. Only volumes that are not locked are selected from the pool of available volumes. Mount the requested volumes. During the next program run, BRBACKUP and BRARCHIVE will attempt to write to all the available volumes in the pool in sequence.

Activities

Activities

The names of the volumes that will be required for the next volume can be displayed as follows:

```
brbackup/brarchive -q
```

You can check whether you have mounted the right volume in the backup device:

```
brbackup/brarchive -q check
```

Neither call actually starts the backup. In particular, you should use these options before scheduling a backup with CRON or a similar utility.

See [-q|-query \[Page 383\]](#)

If you want to select the volumes manually you can deactivate automatic tape management: Set the `init<DBSID>.sap parameters volume_backup` and `volume_archive` to `SCRATCH` or call `BRBACKUP/BRARCHIVE` with the option `-v SCRATCH`.

Backup Verifications

Database verifications must be checked regularly for readability and completeness. This avoids errors occurring when reading from the volume even though the backup was made successfully.

Verification Strategy

At least once in every backup cycle (tape administration cycle) you should check the contents of a complete database backup. A check on the Oracle internal block structure should also take place once in every cycle. Both checks complement each other. However they *do not* replace each other, since they check different aspects.

Verification of Backups with BRBACKUP -w| -verify

The method and extent of a BRBACKUP check differs according to the type of backup:

- Offline backup: After the backup the files are copied back to a temporary directory (`compress_dir`) and compared to the originals one-by-one.
- Online backup: After the backup the files are copied back to a temporary directory (`compress_dir`) but only the file sizes are compared.

Verification is always carried out between volumes: the volumes are written to one after the other, and then the backup is verified.

See [-w|-verify \[Page 389\]](#)

Independent Verification of Backups with BRRESTORE -w| -verify

You can carry out the backup and the readability check on the backed up files independently from each other. The backup check can then be delayed to a later point in time. You can even execute the verification process on another computer.

When checking a backup with BRRESTORE (`brrestore -verify`) only the readability of the tape or disk backups is checked, not the contents. The files are only read, not restored.

See [-w|-verify \[Page 433\]](#)

Verification of Backups with BRARCHIVE -w| -verify

The extent of the check on archived offline redo log files depends on the type of BRARCHIVE backup:

- `brarchive -s|-sc|-ss|-cs`: The backed up files are restored and compared with the originals one-by-one.
- `brarchive -sd|-scd|-ssd|-cdd`: The file sizes of the archived offline redo log files are checked. Since the originals were deleted, a check on the contents is not possible.

See [-w|-verify \[Page 412\]](#)

Verification of Backups with Oracle DB_VERIFY

From ORACLE Release 7.3 the tool DB_VERIFY from Oracle is available, in order to check the Oracle internal block structure of backups or check the database files directly. This means you

Backup Verifications

can recognize errors early (ORA-1578), before they lead to the termination of a program in an application that accesses the blocks. (There is no one-to-one comparison).

DB_VERIFY has the following options:

- Database backup with subsequent restoration to a temporary directory (`compress_dir`) and check on the Oracle block structure:
 - `brbackup -w use_dbv`
- Online block structure check on the database files without backup:
 - `brbackup -w only_dbv`
 - Any number of these verification processes can run in parallel (parameter: `exec_parallel`, option `-e`).
- Temporary restoration of a database backup (`compress_dir`) and verification of the Oracle block structure:
 - `brrestore -w use_dbv.`

The restoration implicitly checks the readability of the backup.

Verification of RMAN Backups

The BRBACKUP and BRRESTORE functions for verifying backups are also supported for RMAN backups. RMAN verifies savesets with the command VALIDATE. They are verified as follows:

1. The savesets are imported from the backup medium. The readability of the backup is checked.
2. RMAN checks the internal consistency of the data in the savesets. No binary comparison is made with the originals.

BRBACKUP can verify online or offline RMAN backups immediately.

A delayed verification with BRRESTORE can only be done on the database host (as long as the database is mounted or open).

A verification with DB_VERIFY is not supported for RMAN backups, since the Oracle block consistency is checked when the backups are made.

SAP continues to make the same recommendations for the frequency of backup verifications: once a week, however at least once in each backup cycle.

For more information, see [Verification of Backups \[Page 155\]](#) and [Profile Parameters and BRBACKUP Command Options \[Page 170\]](#).

Integration of the Recovery Manager

Integration of the Recovery Manager

The Oracle Recovery Manager (RMAN) is a new backup program in Oracle8. It replaces the Enterprise Backup Utility introduced for Oracle7. You can use it as a command line interface (CLI) or as graphical user interface (GUI) within the Oracle Enterprise Manager (OEM).

Use

RMAN uses the SBT interface (**S**ystem **B**ackup to **T**ape) to back up to tape devices. External backup tools can implement this interface as a dynamic link library (backup library).

The following components are delivered with the standard Oracle8 installation:

- Recovery Manager with the Oracle SBT interface (**S**ystem **B**ackup to **T**ape)
- Backup library and backup tool Networker from the company Legato
- Legato's BACKINT interface implementation (as of Oracle 8.0.5)

The SAP installation also delivers the following for use with RMAN:

- the SAP backup library and the SAP backup tool BRBACKUP.

The Oracle/SAP installation gives you two options for integrating RMAN (SAP implementation and Legato implementation). As of Release 4.5B, the SAP implementation of the Oracle interface SBT (SAP backup library) is automatically activated with the R/3 installation.

However, you can also use the Legato implementation, or other external backup libraries. These solutions differ in the backup media that they support and the characteristics of the backup programs used.

Features

One important new RMAN feature is the option of making incremental backups. This allows you to change your previous backup strategy and considerably reduce the amount of data to be backed up. RMAN also offers other advantages:

- Logical database block errors are recognized automatically during the backup. This makes sure that each backup is consistent. This function replaces the weekly check with DB_VERIFY.
- Any database blocks that have never been used are not backed up; this reduces the amount of data to be backed up.
- You can verify backups to tape with the RMAN command VALIDATE.
- The command BEGIN/ END BACKUP is not needed in online backups, since the blocks are checked to see if the data is consistent. This significantly reduces redo log information.

Prerequisites

You are subject to the following restrictions when you use the Recovery Manager as CLI or GUI in the OEM.

- RMAN places information about backups in a recovery catalog. For security this catalog is held in a separate database on a separate host. This means more administrative work.

Integration of the Recovery Manager

- Restoration or recovery is complicated in a disaster situation (loss of production database and recovery catalog). It is often only possible with the help of Oracle Support. Without the recovery catalog data, RMAN cannot recover the database automatically using previous backups.

The database user making the backups currently needs the SYSDBA authorizations. SYSOPER authorizations are not sufficient.

For more information on Oracle RMAN, see the Oracle8 documentation.

Integration

The integration of the Oracle Recovery Manager into the SAP tool BRBACKUP adds security and flexibility to important functions in existing backup strategies and tools:

- The recovery catalog is not used. The backup information is stored in the control file. After the backup, the control file is also backed up. In a restoration, the control file is imported first, followed by the data files.
- The integration of RMAN into BRBACKUP also guarantees integration into the R/3 System (CCMS).
- BRBACKUP tape management functions as previously (when using the SAP backup library).
- You can still use the BACKINT interface with external tools.
- The SAPDBA user interface is unchanged, however, it has been extended with new options.
- All previous SAP backup strategies are supported while using RMAN. Exceptions: RMAN is not relevant for offline redo log backups with BRARCHIVE, standby database backups and split mirror backups.

For more information on backing up the Oracle database with RMAN, see:

[RMAN Backups with the SAP Backup Library \[Page 160\]](#)

[Backups with an External Backup Library \[Page 162\]](#)

[Incremental Backup Strategy Without a Backup Library \[Page 164\]](#)

RMAN Backups with the SAP Backup Library

RMAN Backups with the SAP Backup Library

Integration

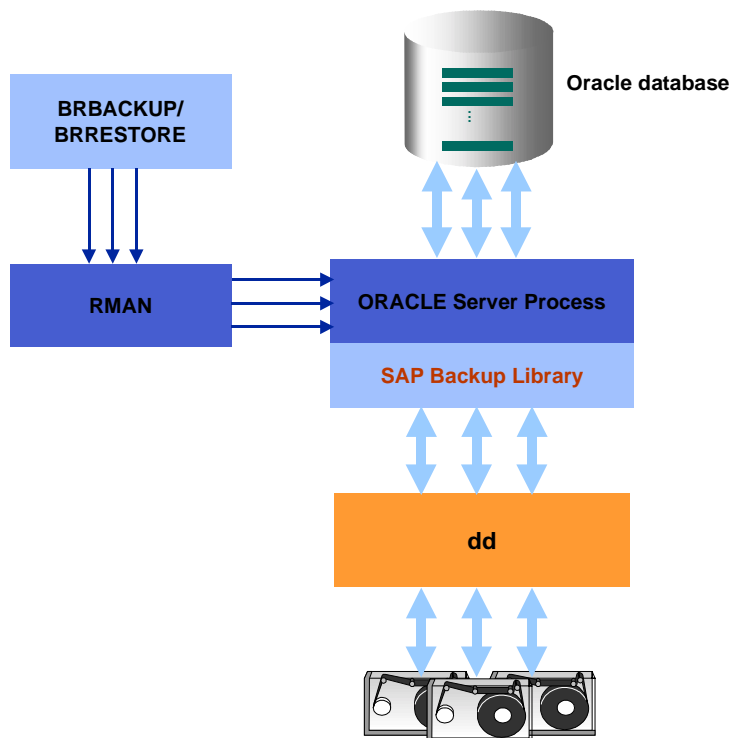
The SAP backup library is an implementation of the Oracle interface SBT (**S**ystem **B**ackup to **T**ape) in the form of a dynamic function library (**D**ynamic **L**ink **L**ibrary). This is called by the Oracle server process to back up data. This is mainly done to tape.

Without this library, the Oracle Server process can only back up to local disk. For tape management reasons, an SBT backup library is always required for a backup to tape.

Features

1) Backup to Tape Devices

BRBACKUP calls the Recovery Manager for a backup to tape devices. For a parallel backup to multiple tape devices, the required number of parallel processes are started. RMAN activates the Oracle server process to read the data from the database and transfer it to the SAP backup library. The library then backs up the data to local or remote tape devices with the operating system command *dd*.



For backups to remote tape devices, a remote shell is used to set up a network connection to the remote host. Since errors can occur when you transfer data over a network, we do not recommend that you back up a production system in this way.

RMAN Backups with the SAP Backup Library

You can use RMAN to back up to tape with the following variants:

tape_copy_cmd \ backup_mode	cpio	dd	rman + backup library
partial	X	X	X
all	X	X	X
full (level 0)	X+cat.	X+cat.	X
incr (level 1)			X

Key: X: Backup possible; X+cat.: Backup with subsequent cataloging possible

2) Backup to Remote Disks

As of Release 4.5A, you can make BRBACKUP backups to remote disks. You can use the following profile parameters:

`backup_dev_type = stage | stage_copy | stage_standby`

With RMAN you can make only incremental backups (level 1) to remote disks. The SAP-specific file transfer protocol SAPFTP is used alongside the SAP backup library. The full backup (level 0) needed for the incremental backup is made with BRBACKUP and cataloged automatically afterwards.

You can use RMAN to back up to remote disk with the following variants:

stage_copy_cmd \ backup_mode	rcp	ftp	RMAN + SAP backup library
partial	X	X	
all	X	X	
full (level 0)	X+cat.	X+cat.	
incr (level 1)			X+SAPFTP

Key: X: Backup possible; X+cat.: Backup with subsequent cataloging possible; X+SAPFTP: Backup with SAP file transfer protocol possible

Backups to remote disks are particularly useful in standby database scenarios (BRARCHIVE copy of offline redo log files to the backup host) and when you use AMASS software to control tape jukeboxes (virtual disks).

For more information, see [Profile Parameters and BRBACKUP Command Options \[Page 170\]](#).

Backup with an External Backup Library

Backup with an External Backup Library

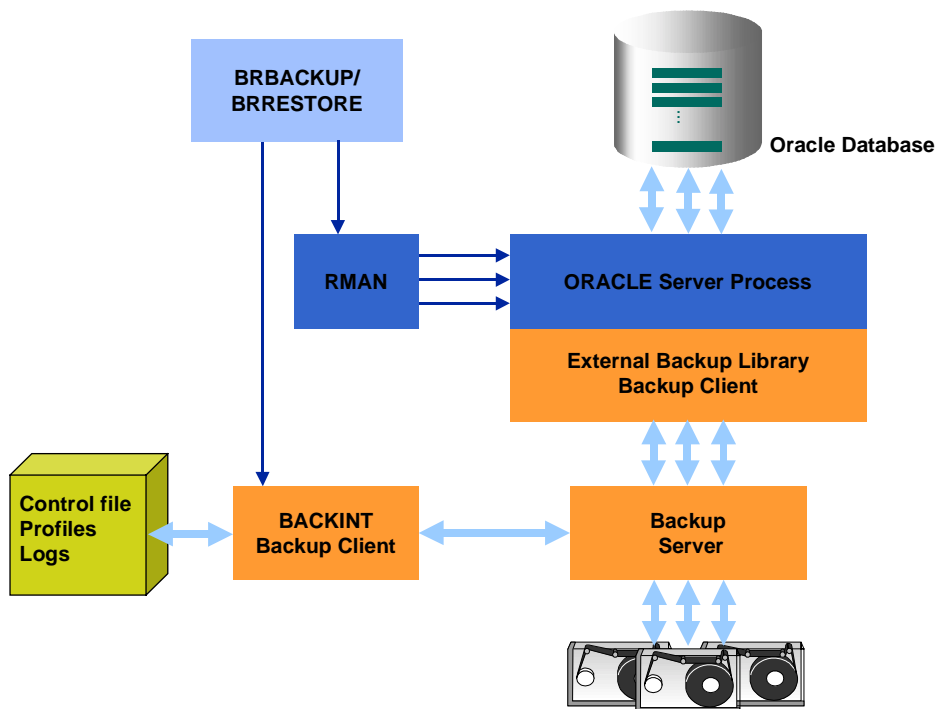
Use

You can use external backup software together with RMAN to back up the database. To back up data to tape the Oracle SBT interface is implemented as a dynamic function library. The type of backup media you can use depends on the external backup tool and the corresponding backup library.

Features

The database is backed up in two phases :

1. BRBACKUP calls the Oracle Recovery Manager and starts the Oracle server process. The Oracle server process reads the data to be backed up from the Oracle database, and transfers it to the external backup library. The external backup library calls a backup client. The backup client passes the data to the backup server. The server backs up the data to the storage medium.
2. BRBACKUP uses the BACKINT interface to pass the control file, the initialization profile and the log files to the backup client. The backup client passes the data to the backup server. The server backs up the data to the storage medium. This control file contains information on the backup of the files from the first phase.



You can use the following backup variant for external software, with or without RMAN:

Backup with an External Backup Library

backup_dev_type backup_mode	util_file without RMAN	rman_util with RMAN and external SBT backup library
partial	X	X+BACKINT
all	X	X+BACKINT
full (level 0)	X+cat.	X+BACKINT
incr (level 1)		X+BACKINT

Key: X: Backup possible; X+cat.: Backup and subsequent cataloging possible; X+BACKINT: Backup with subsequent backup of the profile, log and control files with BACKINT possible

As before, you can make normal database backups with the BACKINT interface and external backup tools:

```
backup_dev_type = util_file
```

Subsequent cataloging means that a full backup can be used as a reference backup in an incremental backup strategy.

Even when you use an external backup tool, you can make incremental backups (level 1) with RMAN only. Since RMAN can back up only database files and not profiles or logs, the profile, log and control files are backed up in a second phase with BRBACKUP and the BACKINT interface.

For more information, see [Profile Parameters and BRBACKUP Command Options \[Page 170\]](#).

Incremental Backup Strategy Without a Backup Library

Incremental Backup Strategy Without a Backup Library

Use

An incremental backup (level 1) must always be made with RMAN, however the full backup (level 0) can be made with BRBACKUP and an external backup tool, and without RMAN. The full backup is cataloged automatically as a level 0 backup.

The above solution is an alternative to the complete RMAN solution that offers better security.

For more information, see [RMAN Backup Strategies \[Page 110\]](#) and [Incremental Backups After Structural Changes \[Page 112\]](#).

Prerequisites

RMAN Full Backup (Level 0) to Disk

You can make a full backup with RMAN and without a backup library only to disk. Afterwards, use BRBACKUP to back up the disk backup to tape. This is a [two-phase backup \[Page 124\]](#).

Advantages:

- You can use RMAN functions (such as automatic block checks).
- There is less redo information in an online full backup, since RMAN checks the consistency of the data at block level, so removing the need for the BEGIN/ END BACKUP commands.

Disadvantage:

- You need to back up the full backup to tape, causing extra work.

Full Backup (Level 0) with BRBACKUP or an External Tool, Without RMAN

You can make the full backup (level 0) without RMAN. After you make the full backup (using BRBACKUP or BACKINT and an external tool) it is cataloged as a level 0 backup. BRBACKUP can make a full backup directly to tape, removing the need for a second phase.

Advantages:

- Increased security when you restore or recover the backup. You can recover the database without RMAN because no savesets are formed during the full backup. You can also replace the import of the last incremental backup with an import of the redo log files created since the last full backup.
- Under certain conditions the backup may be faster, since RMAN is not involved in the full backup. The data is backed up directly using *cpio*, *dd* or BACKINT.

Disadvantages:

- You cannot use RMAN features such as the database block check.
- The commands are used in an online full backup. This produces much more redo log information. This data increases the load on the system and makes an ArchiveStuck more likely.
- If you make a recovery without RMAN you may have to import a lot of redo log files since the full backup is usually only made once a week.

Incremental Backup (Level 1) to Disk with RMAN

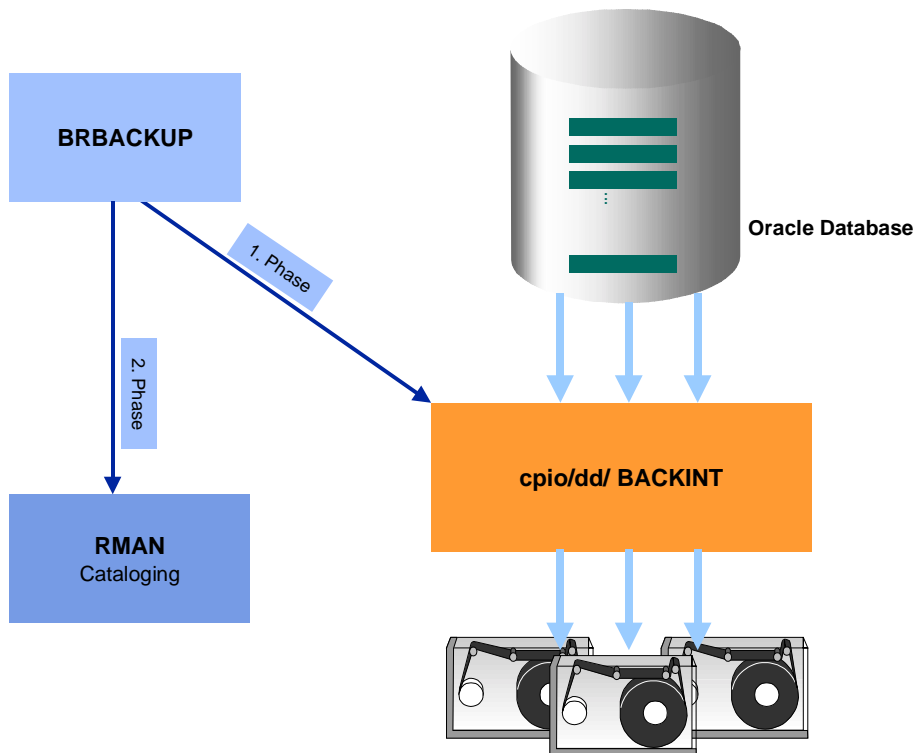
Incremental Backup Strategy Without a Backup Library

You can also make **incremental backups (level 1)** with RMAN and without a backup library, but only to disk. There is no one-to-one copy of the database files; instead savesets are created which means you can only recover the savesets of an incremental backup with RMAN (see [Saveset Grouping \[Page 167\]](#)). In a second phase, back up the incremental backup to tape with BRBACKUP ([Two-Phase Backup \[Page 124\]](#)).

Procedure

Full Backup and Incremental Backup Without Backup Library

a) Full Backup (Level 0) of the Database

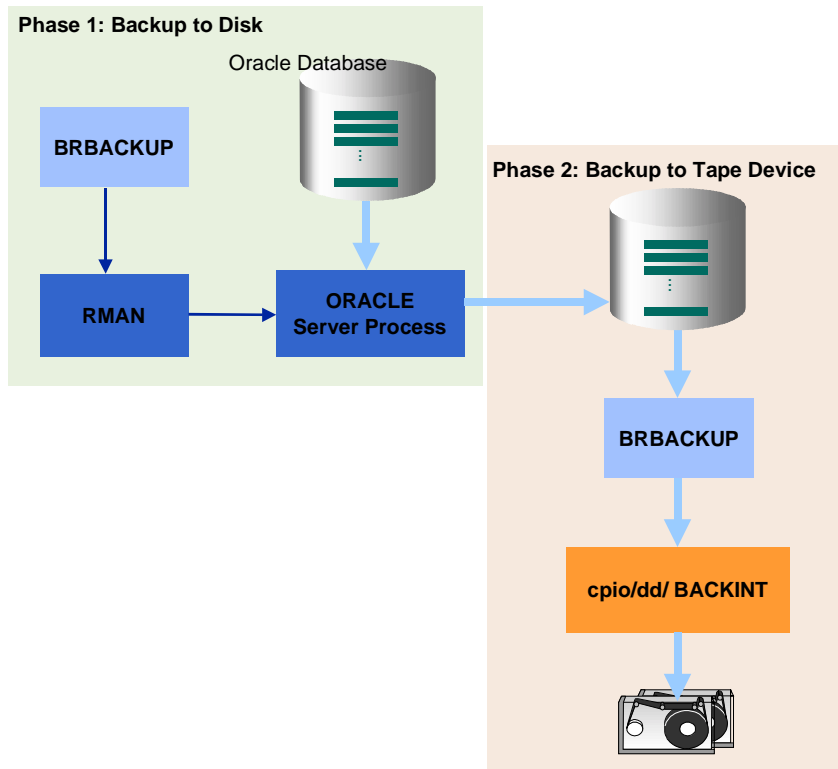


BRBACKUP calls *cpio*, *dd* or BACKINT to back up the database. *cpio* or *dd* read the data from the disk and write it to tape. If you use the BACKINT interface, the data is transferred to external backup software that then backs up the data to the backup medium. The backup is then cataloged automatically as a full backup (level 0). This means it can be used as a reference for later incremental backups.

b) Incremental Backup (Level 1) of the Database

Since you cannot make an incremental backup directly to tape without a backup library, this step is a [two-phase backup \[Page 124\]](#):

Incremental Backup Strategy Without a Backup Library



In the first phase, the Oracle server process reads the data from the Oracle database, and saves it to disk in a save set.

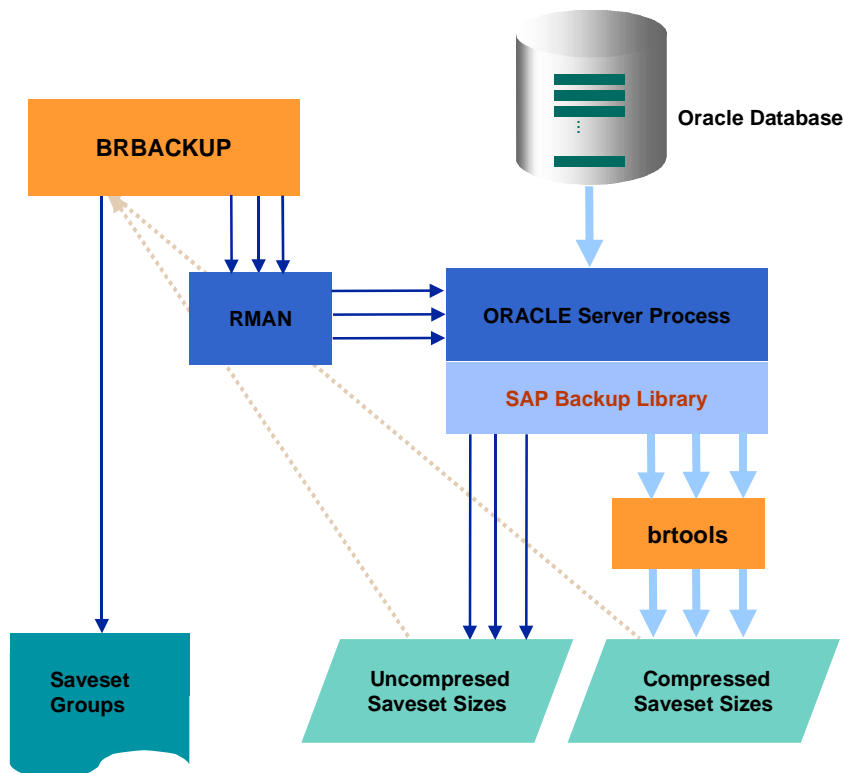
In the second phase, BRBACKUP calls `cpio`, `dd` or `BACKINT` to read the data from the disk and write it to the backup medium. If you use the `BACKINT` interface, the data is transferred to external backup software that then backs up the data to the backup medium.

For more information, see [RMAN Backup Strategies \[Page 110\]](#) and [Profile Parameters and BRBACKUP Command Options \[Page 170\]](#).

Saveset Grouping

The SAP backup library helps to optimize the utilization of quick tape drives by combining multiple data files in savesets. Multiple file access (file multiplexing) maximizes the flow of data (streaming mode). You can define the number of files contained in each saveset with the profile parameter `saveset_members` or the command option `brbackup -s|-savesets`. The SAP backup library optimizes the distribution of the data files across the savesets in a preparation run (`backup_dev_type=rman_prep`):

1. Estimation of the expected compression of the savesets by using BRTOOLS to determine the decompressed and compressed file sizes.
2. BRBACKUP forms savesets and stores the information on the composition of the savesets and the compression rates for future backups in the database.



A saveset can contain individual data files, all files of a tablespace, or the complete data backup. The size of the savesets for the backup must be selected according to the tape device. A fast data flow with a minimum saveset size is the optimum. Large savesets are not recommended, since in a restoration the complete saveset has to be imported, even if only one data file is required.

`saveset_members = all` is set as standard for an incremental backup with the SAP backup library so that only one “incremental saveset” is created including all changed blocks.

SAP recommends that you make the preparation run once a month (`brbackup -d rman_prep`), and each time that major changes have been made to the database, for example

Saveset Grouping

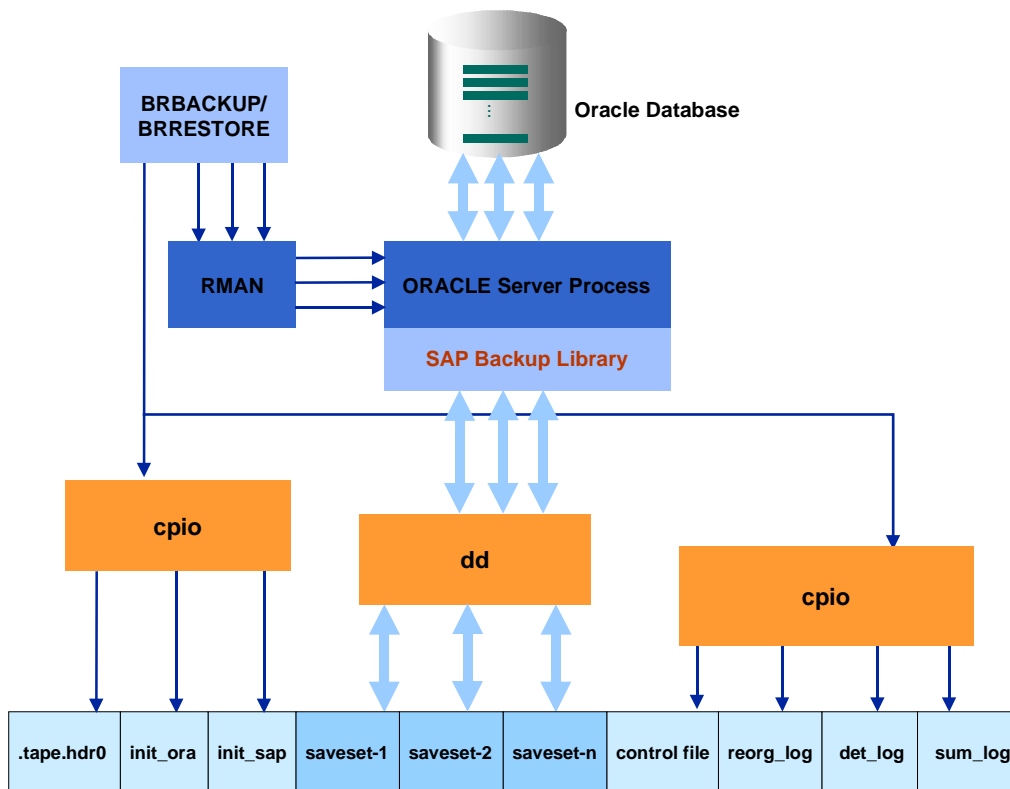
after a reorganization, an SAP release change, database migration or transferral of mass data. After a tablespace extension it is enough to execute the run for the newly created files only.

For more information, see [Profile Parameters and BRBACKUP Command Options \[Page 170\]](#).

Tape Layout

The tape layout is fundamentally unchanged from the previous backup methods without RMAN. This means that you do **not** need to reinitialize the tapes when you use the SAP implementation of the SBT interface (SAP backup library).

1. BRBACKUP uses `cpio` to back up the tape management information (tape label) and the initialization files `init<DBSID>.ora` and `init<DBSID>.sap` to tape.
2. The SAP backup library uses `dd` to back up the savesets to tape.
3. BRBACKUP uses `cpio` to back up the control file and the log files to tape.



For more information, see [Saveset Grouping \[Page 167\]](#) and [Profile Parameters and BRBACKUP Command Options \[Page 170\]](#).

Tape Layout

Profile Parameters and BRBACKUP Command Options

SAVESET Grouping and Preparation Run

To define the size of the savesets by the number of files they contain, make the following entry in the initialization file `init<DBSID>.sap`:

- `saveset_members = 1|2|3|4|tsp|all` (default: 1)

The corresponding command option is: `brbackup -s|-savesets`

You can start the preparation run for determining the optimum distribution of files across the savesets as follows:

- `brbackup -d rman_prep`
- SAPDBA menu: *Backup database* → *Backup function*

RMAN Backup with the SAP Backup Library

For backups with the SAP backup library you need to make the following special entries in the initialization file `init<DBSID>.sap` or set the corresponding command options:

- to local tape devices
 - `backup_dev_type = tape|tape_auto|tape_box`
 - `tape_copy_cmd = rman`
- to remote tape devices
 - `backup_dev_type = pipe|pipe_auto|pipe_box`
 - `tape_copy_cmd = rman`
- to remote disk (incremental)
 - `backup_mode = incr`
 - `backup_dev_type = stage|stage_copy|stage_standby`
 - `remote_user = "<user name> [<password>]"` (requirement for SAPFTP)
 - `remote_host = <host name>` (requirement for SAPFTP)

RMAN Backup with an External Backup Library

For backups with external backup libraries, you need to make the following entries in the initialization profile `init<DBSID>.sap` or the corresponding command options:

- `backup_dev_type = rman_util`

You can set other parameters as well, which BRBACKUP sends directly to the Oracle RMAN (without the prefix `rman_`). These parameters affect how the savesets are formed and the parallel backup to different media.

- `rman_channels`
Number of channels allocated to the parallel processes

- `rman_filesperset`
controls the number of files in each saveset; default: 0 (one files in each saveset for normal backups; all files in one saveset for incremental backups)
- `rman_kbytes`
- `rman_readrate`
- `rman_maxopenfile`

For more information on these parameters, see the *Oracle8 Server – Release 8.0 Backup and Recovery Guide*.

RMAN Backups Without Backup Library

For normal backups to disk with RMAN, you need to make the following entries in the initialization file `init<DBSID>.sap` or the corresponding command options:

- `backup_dev_type = disk`
- `disk_copy_cmd = rman`

The individual steps of the incremental backup strategy without backup library have the following command options:

1. Full backup (level 0) without RMAN:
 - `brbackup -m full -d tape|pipe|util_file`
 - `tape_copy_cmd = cpio|dd`
2. Incremental backup (level 1) to disk with RMAN:
 - `brbackup -m incr -d disk`
3. Backup of the incremental backup to tape without RMAN (two-phase backup):
 - `brbackup -b last -m incr -d tape|pipe|util_file`

Restoring Incremental Backups

To make the database consistent again, for example after a media error, you can use the following BRRESTORE command (RMAN adjusts the changes in the files that have been affected):

1. If necessary: Restoration of the control file and the online redo log files from the last incremental backup:
 - `brrestore -b last -m 0[,00]`
2. Restoration of a full backup (level 0):
 - `brrestore -b <brb_log_name> -m all|<object list>|..`
3. Restoration of the last incremental backup (level 1):
 - `brrestore -b last -m incr`
4. Import of the offline redo log files with the Oracle Server Manager.

Tape Layout

Restoring Incremental Backups with Structural Changes

In an incremental backup with structural changes, the new files are backed up in full to a second saveset in subsequent incremental backups. The following savesets are created if the SAP backup library is used:

- `<coded timestamp>.INCR` (changes to the 'old' files)
- `< coded timestamp >.FULL` (newly added files)

The backup of the new files to a separate saveset allows a precise specification of which files are to be restored.

Restoration of changes to all files that were in the database at the time of the last full backup (level 0) (first saveset of the last incremental backup):

- `brrestore -b last -m incr_only`

Restoration of the files that have been added since the last full (level 0) (second saveset of the last incremental backup):

- `brrestore -b last -m incr_full`

Restoration of the whole incremental backup (both savesets, if they exist):

- `brrestore -b last -m incr`

More information:

Oracle8 Server – Release 8.0 Backup and Recovery Guide

Backup of Large Oracle Databases

Large databases contain 110 GB - 1 TB of data. For various reasons, you may not be able to back up the database daily:

- **Performance problems:** A heavy load is placed on database server resources (particularly CPU time, system and I/O buses, disk and volume controllers); online operations in the R/3 System are restricted.
- You do not have enough time; you should make backups when there is little activity in the R/3 System, usually at night, however you may easily exceed the 12 hours available.
- The amount of database data is much greater than all other data managed by the system.
- A common backup strategy for database and non-database files is not a solution since non-database data can easily be backed up with operating system tools (such as `tar`).

The system is normally configured so that there is a single server for the database, where no other large applications run.

You should not back up large databases across a network because of instability and performance problems.

Your ability to make backups of large databases with BRBACKUP (and `cpio` for the copy processes) depends on the following factors:

- Capacity and maximum throughput of the tape devices
- Disk access times
- Maximum throughput of the I/O (SCSI) buses
- Maximum throughput of the system bus
- `cpio` performance; determined by internal buffering and blocking

BRBACKUP itself places minimal load on the backup process. Any hardware-specific restrictions can only be improved by the hardware vendor.

The hardware configuration for a large database needs careful planning for an optimal backup. This may require multiple tape changes and the management of hundreds of volumes. If possible, use tape jukeboxes supported by the BACKINT interface and external backup programs.

You are informed of the support provided by the SAP utility program BRBACKUP when backing up large databases.

Hardware and Duration Example for a 100 GB Backup

Hardware and Duration Example for a 100 GB Backup

The following example shows how a database backup may run under normal conditions:

- Backup devices:

The database backup is carried out onto locally mounted tape units. The DAT DDS-2 disk drives are used presently with hardware compression. Ten tape units are available in our example.
- Throughput:

The throughput of the backup is about 2 GB per hour.
- Tape capacity assuming 120 meters of tape:
 - about 2.4 GB for tape units without hardware compression
 - about 7 GB for tape units with hardware compression (assuming the average compression rate for SAP data: about 3-4)
- Time window:

The backup should be possible in less than 10 hours overnight.

 - Size of the database:

The size of the database is 100 GB.

If the database can be backed up in parallel to ten tape devices, then it can be made within ten hours (with hardware compression).



Remember that scalability is restricted. This means that backup performance does not rise in proportion to the number of tape devices; if you use more tape devices, backup time will increase for each device.

Backup Devices for Large Databases

The backup of large databases creates an extreme load, both on the system resources and on the backup device being used.

Some examples follow, as well as some orientation values for the hardware environment of the backup devices.

- The backup should not be carried out over a network, but either to volumes in locally mounted backup devices or directly to hard disks.
- Backup devices:
 - The maximum number of locally connected tape devices supported by BRBACKUP:

as of Release 2.1K/ 2.2E	50
as of Release 3.0A	255
 - Backup devices (for example magneto-optical media), which are addressed via an external backup program can be reached via the interface program BACKINT (see [External Backup Programs for Large Databases \[Page 178\]](#)).
 - BRBACKUP/BRARCHIVE do not support tape jukeboxes. However, you can address such devices via the BACKINT interface to external backup programs.

The following backup devices are the most common: 4 mm (DAT DDS-1, DDS-2) or 8 mm (video).

However, these devices should only be used to backup large databases where there is no other option, due of the low tape capacity and small throughput. Experience shows that for a database backup of about 500 GB, about 100 tapes would be necessary. The backup time would also greatly exceed 12 hours, even with 20 tape units running in parallel.

Tape units with a larger capacity and higher throughput rate should be used if possible.



The following examples are not a representative selection of the available backup devices.



IBM 3490

Throughput with backup: about 5-10 GB per hour

Tape capacity: about 2 GB per tape

These devices offer high throughput, but relatively low capacity. These would only be useful if a tape jukebox were used, running an external backup program, for example, via the BACKINT interface



IBM 3590

Throughput with backup: about 5-10 GB per hour

Tape capacity: about 20-30 GB per tape

Backup Devices for Large Databases

These new devices from IBM have promising technical possibilities. However, there is little practical experience available, as these devices are not yet very widespread.



DLT (Digital Linear Tape)

Throughput with backup: about 3-6 GB per hour

Tape capacity: about 10-15 GB per tape

If a throughput of 3 GB per hour is assumed for 20 tape units in parallel, the backup of a 500GB database could conceivably run within 12 hours on less than 50 tapes.

The numbers given in the examples are based solely on the abilities of the backup devices. However, other factors also play a large role, for example, the throughput values of the computer (in particular, hard disk access times, system bus speed, I/O (SCSI) bus speed). As a result, the total throughput decreases significantly. Do not mount too many backup devices on one I/O bus, so as not to overload it. If hard disks and tape units are mounted on the same I/O bus, you must expect the load to be split between the two.

Backing Up Large Databases to Tape with BRBACKUP

BRBACKUP calls `cpio` in order to copy individual files from the hard disk to tape. As a result, throughput is basically determined by `cpio`.

BRBACKUP offers a number of functions to enable an optimal backup of large databases.

- BRBACKUP can backup in parallel to several locally mounted tape units.
- In a parallel backup, BRBACKUP also supports tape units with automatic tape swapping (so-called autochangers). As a result, a completely automatic backup is also possible if tape swapping is to occur for one or more tape units during a backup.
- All files to be saved are distributed to the volumes (tapes) inserted in the tape units. BRBACKUP has different optimization targets (see [Optimization of File Distribution \[Page 180\]](#)).

Restrictions for Backup with BRBACKUP:

- As the database server is working at 100% during the backup, other activities on the computer should be reduced to the absolute minimum during the backup. Virtually all of the computer's resources should be available for the backup.

This is obviously contradictory to the high availability required of the R/3 System. There is no simple solution to this conflict.
- No `cpio` continuation mechanism is supported during parallel backups (see [cpio Continuation Tape \[Page 224\]](#)), that is, individual files must always be completely saved onto one volume. Tape swapping must always be carried out by BRBACKUP. Therefore, the size of a file to be saved must not exceed the tape capacity. If you work with tape units with hardware compression, the size of the compressed file must not exceed the tape capacity. Make sure that the tape capacity is not set too high, in order to avoid reaching the end of the tape.
- Since individual files can only be written in their entirety to a volume, there is a certain amount of wastage when distributing the files to the available tapes. That is, the tape capacity cannot be fully used. When distributing the files, BRBACKUP makes sure that the tape capacity is never exceeded. The size of the tape header files (label, `init<DBSID>.ora`, `init<DBSID>.sap`) and the log files at the end of the tape (central, detailed and summary log) are not taken into account.
- Special features when using tape units with hardware compression: see *Tape Units with Hardware Compression*.

[Optimization of File Distribution \[Page 180\]](#)

[Optimization when Using a Logical Volume Manager \[Page 182\]](#)

[Partial Backups \[Page 184\]](#)

External Backup Programs for Large Databases

Information on the BACKINT interface can be found in [Using External Backup Programs \[Page 241\]](#).

The essential advantage of using this interface is pointed out here: You can use other backup media (for example magneto-optical (MO) media) for a backup or you can transfer volume management to other systems (for example tape jukeboxes)..

The disadvantage of these programs was that in an online backup (before SAP R/3 Release 3.0B) all the tablespaces to be backed up were in backup mode during the whole backup and considerably more redo log entries were created for this reason (that is, the volume of offline redo log files was considerably increased in a backup with BACKINT). The interface was extended to eliminate this problem. See information about the option `util_file_online` of parameter [backup_dev_type \[Page 314\]](#) or about the corresponding command option `-d util_file_online`.

Information on the throughput and performance of external backup programs can be obtained from the vendors providing the interface program BACKINT and its support for the respective backup program.

Parallel Backup of Large Databases to Disk with BRBACKUP

BRBACKUP offers the option to backup onto several hard disks (see [Backup to Multiple Disks \[Page 119\]](#)).

- Parallel backup onto the specified hard disks is possible. The directories must be defined in parameter `backup_root_dir`. The backup can be carried out with or without software compression. You can make a [two-phase backup \[Page 124\]](#).

The degree of parallelism (the number of parallel copy processes) is controlled by parameter `exec_parallel` of initialization profile `init<DBSID>.sap` (see [exec_parallel \[Page 331\]](#)) or the command option `-e`.

In order to avoid weighing the database server down with the second level of backup, the following procedure would be conceivable: Unmount (`umount`) the file system from the database server, and mount (`mount`) this file system on a second host. Then start the backup from this host. A requirement for this procedure is that the hard disk controllers for the backup disks can be physically mounted on both hosts simultaneously. The operations `umount` and `mount` are necessary, since the file system cannot be mounted on different computers at the same time due to the buffering mechanism.

Optimization of File Distribution

Optimization of File Distribution

When distributing the files to the tapes, BRBACKUP has different optimization targets:

- BRBACKUP attempts to save all files from a hard disk to a volume (tape). As a result, competing hard disk accesses should be avoided, and the number of read/write accesses should be kept to a minimum.
 - If a Logical Volume Manager (LVM) is used, this can only be attained if the logical volumes are not unnecessarily distributed over several physical hard disks. See [Optimization with a Logical Volume Manager \[Page 182\]](#).
 - Using the backup times for individual files stored in the database, BRBACKUP attempts to minimize the total backup time by keeping the backup time equal on each of the individual tape units.
 - This mechanism helps to avoid the following problem: When using tape units with hardware compression, the backup time cannot be estimated accurately from the file size or the size of the compressed file. The optimization is carried out therefore from the backup times stored by BRBACKUP.
 - The tape capacity in general is not used to its maximum due to temporal optimization (if the optimal backup time is reached for a tape, no further files are saved to this tape, though the tape still has plenty of space left). The total capacity of all available tapes inserted at the same time must therefore significantly exceed the total size of the files to be saved.
 - The temporal optimization is cancelled internally if it would cause a tape change (BRBACKUP always attempts to avoid volume changes). In this case, the entire capacity of the tapes is used. As a result, the backup time can vary for individual volumes.
 - BRBACKUP always attempts to use all tape units mounted, even if the backup would fit onto a smaller number of tapes. This is done in order to minimize the total backup time.
 - The one exception is when the number of files to be backed up is less than the number of tape devices.
 - BRBACKUP sorts the files to be saved according to their size and distributes the largest files to the available tapes first, followed by the smaller ones. As a result, wastage is reduced on individual volumes, as the tape capacity (parameter `tape_size`) can never be exceeded and the total tape capacity can be better utilized.
 - BRBACKUP is capable of “learning”. The backup times of individual files are stored in the database and used in the next backup for temporal optimization (if this is carried out).
 - As a result, the changes to backup time are taken into account for individual files. The backup time can change, for example, if the filler level of the individual database files varies and the files should be saved with hardware compression using a tape unit.
 - The option `-o|-output dist,time` for BRBACKUP/BRARCHIVE can be used to:
 - display the file distribution carried out by BRBACKUP before starting the actual backup.
 - display the backup times of the individual files after the backup is completed.
- See [Log Supplements \[Page 442\]](#).

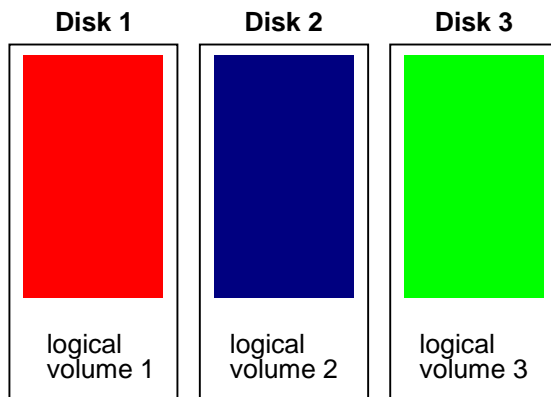
Optimization with a Logical Volume Manager

Optimization with a Logical Volume Manager

If a Logical Volume Manager (LVM) is used, BRBACKUP can only save all files from a hard disk to tape if the logical volumes are not unnecessarily distributed over several hard disks (see configuration A).

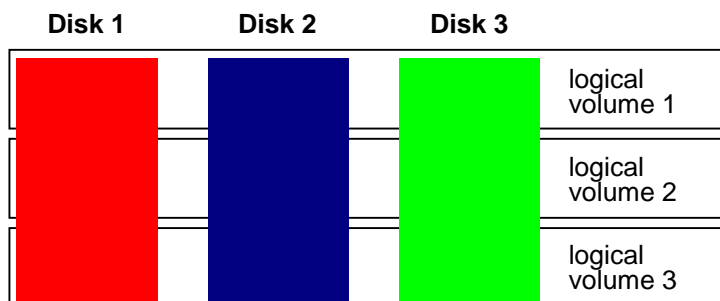
Configuration A:

Each logical volume exactly corresponds to one hard disk (for example logical volume 1 = disk 1, logical volume 2 = disk 2, logical volume 3 = disk 3).



Configuration B:

Each logical volume is set up so that it covers areas of all the hard disks (for example logical volume 1 = area1.disk1+area1.disk2+area1.disk3, logical volume 2 = area2.disk1+area2.disk2+area2.disk3, logical volume 3 = area3.disk1+area3.disk2+area3.disk3).



Configuration A is more efficient than configuration B with respect to database backup. Configuration B, however, can provide a better performance for online operation of the SAP system.

If you plan the configuration of logical volumes for large databases, you should find a compromise for the options which meets your requirements. You should consider which is more important: a more effective backup or higher performance of the online operation of the SAP system.

Some notes on this point:

Optimization with a Logical Volume Manager

- **Advantages** of using an LVM: easier administration, high flexibility, higher security by using RAID systems
 - **Disadvantages** of using an LVM: performance loss through management overhead and possible (configuration B) deterioration of optimization through BRBACKUP
- Although the higher security and availability of your datasets provided by using the LVM has a generally high priority, you should consider whether you could do without an LVM. You lose the above-mentioned advantages this way, but, on the other hand, you can carry out a backup of the datasets considerably more effectively by means of BRBACKUP.
- In large databases, the configuration of the database files is not variably selectable. Every change to the configuration (for example, structure changes due to a tablespace extension or a tablespace reorganization with data files) must always be planned with respect to the physical hard disk configuration and the influence that this will have on the performance. The use of an LVM therefore has no major advantage in this case.

Partial Backups

Partial Backups

You can reduce long backup times by dividing the backup of the entire database into several partial backups.

Advantages

- The partial backups can be carried out daily without time problems.
- The database can be recovered at any time if the corresponding redo log entries exist. It is also possible to restore the entire database and then recover it.
- Recovery after a media error (crash of individual hard disks) can be done quickly as only the files of the affected hard disks must be restored.

Disadvantages

- Restoring the entire database takes as long as the total of all of the partial backups. This situation only normally occurs with logical errors (for example program errors or user errors). There is, however, a low probability of this happening.
 - It must be ensured that all database files were saved in a cycle of the partial backups. This is the responsibility of the database administrator. Backup at tablespace level (not at database file level, which is, in principle, possible), as BRBACKUP can then ensure that all files from a tablespace are backed up. Splitting extremely large tablespaces into several smaller tablespaces can be useful in this case (for example, extremely large tables could be put into separate tablespaces, and then backed up separately at tablespace level).

Standby Database Scenario

The standby database is supported officially by ORACLE as of Version 7.3. The ORACLE documentation contains detailed information on this database configuration. The following sections provide an overview of the main features of the disaster recovery configuration.

See:

[Standby Database Configuration \[Page 186\]](#)



The disaster recovery scenario should only be used by experienced users.

Customers who decide to use this scenario **are fully responsible themselves** for configuring and performing disaster recovery.

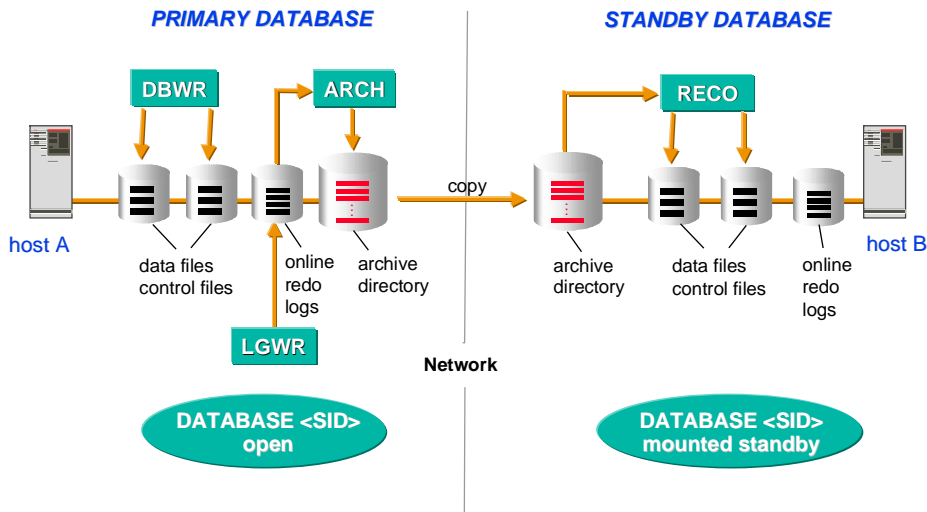
Customers who want to use disaster recovery on their own responsibility can be supported by the SAP utilities BRBACKUP/BRARCHIVE.

[Standby Database Scenario: Advantages and Disadvantages \[Ext.\]](#)

Standby Database Configuration

Standby Database Configuration

When the primary (productive) database is duplicated on a standby database, this is referred to as a standby database configuration. The aim of this configuration is to minimize downtime if the primary database suffers an error, since the standby database can assume the role of the productive database in a very short time. The following diagram illustrates the standby database concept:



Two identically configured databases operate on two identically configured hosts.

- The primary (productive) ORACLE instance is located on the first host, the database is open and fully available for all SQL prompts of the R/3 System. The primary database system is also the system which directly executes all database requests.
- The standby database is a copy of the primary database and is only intended as a recovery system.

The standby ORACLE instance on the second host is in a mounted standby state (not opened!) and is recovered constantly. This means that the standby instance incorporates all changes to the data of the primary instance either immediately, or with a slight delay. To do this, the offline redo log files created in the primary database system are imported (only the redo entries already archived by ORACLE can be imported). The standby instance therefore 'follows' the state of the primary instance.

If it is necessary to recover the primary database system (for example, after a media error), the standby instance can assume the functions of the primary instance in very short time ('takeover'). The recovery mode of the standby instance is therefore ended, and the standby database opened for online operation.

Since all data files are already located on the standby host, costly reloading of the files is avoided. Some redo entries may still need to be applied to the files to enable all transactions to be incorporated in the standby instance. This means that you must first import the missing offline redo log files from the primary instance. You can then try to archive the current online redo log file of the primary instance with the ORACLE command `ALTER SYSTEM ARCHIVE LOG CURRENT` and also to import these redo entries in the standby instance. If this command fails,

Standby Database Configuration

the current online redo log file can be copied to the standby host. It may be possible to directly import the redo entries from the online redo log file.

After the takeover, a standby database needs to be set up again (usually on what was the primary host).



Changes to the physical structure of the primary database (creating new files, renaming files, changes to online redo log and control files) are not automatically incorporated in the standby database in every case. The DBA may need to intervene depending on the type of change.

If it is not possible to incorporate the changes automatically, the recovery process is stopped, and the DBA needs to intervene manually to incorporate the structural change in the standby database. After that, the recovery process needs to be started again.

Renaming of database files in the standby database is not supported by BRBACKUP. The original names of the primary database need to be retained.



If commands are executed in the primary database with the UNRECOVERABLE option, these changes do not appear in the redo log files. It is therefore not possible for the standby instance to receive any information about such changes. In this case, no error messages appear during the recovery process. They are, however, recorded in the standby database ALERT file. You should therefore check the ALERT file regularly.



You will find more detailed information in the ORACLE documentation. The new and/or changed SQL and SVRMGR commands are also described there as well as the necessary init.ora parameters, which are required for working with a standby database.

See also:

[Standby Database: Support by BRARCHIVE \[Page 188\]](#)

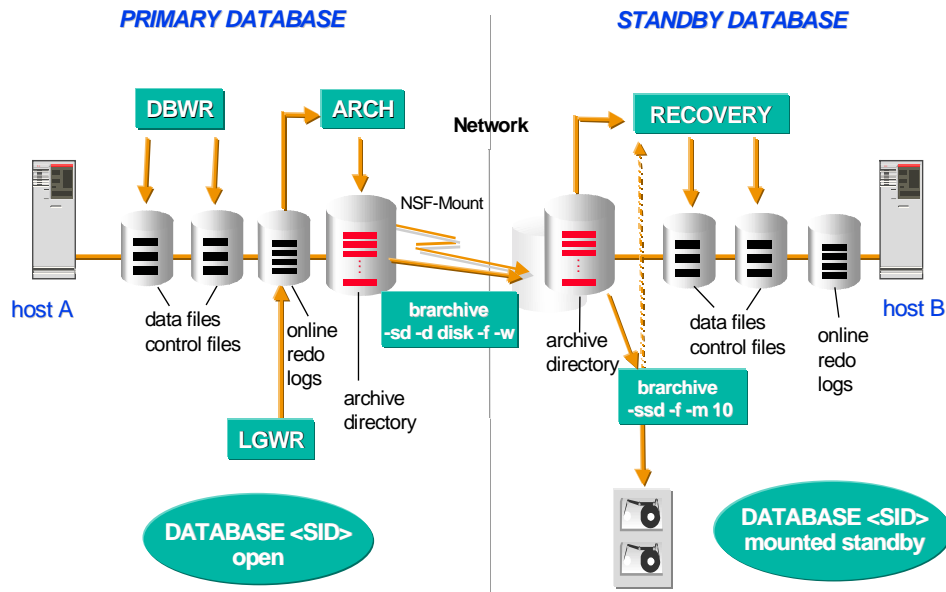
[Standby Database: Backup with BRBACKUP \[Page 190\]](#)

[Standby Database Scenario: Advantages and Disadvantages \[Ext.\]](#)

Standby Database: Support by BRARCHIVE

Standby Database: Support by BRARCHIVE

In the standby database scenario, transfer of the offline redo log files from the primary to the standby instance can be controlled by the SAP utility BRARCHIVE. This is possible, since BRARCHIVE is able to copy offline redo log files to a hard disk.



A BRARCHIVE process runs on the primary host. It copies the offline redo log files to a mounted directory, which represents the archive directory (usually `saparch`) on the standby host. The copy process runs through the network, BRARCHIVE must therefore be used with the verify option (`-w`).

A BRARCHIVE process also runs on the standby host. This process waits for the offline redo log files created in the mounted archive directory. If a redo log file was copied completely, BRARCHIVE assumes the task of importing these redo entries into the standby instance (option `-m` | `-modify`), backing up the redo log file and deleting it if necessary. BRARCHIVE therefore initiates the recovery process of the standby database, in which the offline redo log files are processed individually.

Importing the redo entries can be delayed by a few minutes (the delay is specified in the option `-m <Delay(Minutes)>`). If a logical error occurs in the primary instance (for example, accidental deletion of a table), it is possible to prevent this error from being imported in the standby instance.

The offline redo log files are imported with the following ORACLE command: `RECOVER STANDBY DATABASE;`

See: [-m|-modify \[Page 401\]](#)



To import the redo log files, the DB user (usually `SYSTEM`) must have the `SYSDBA` authorization.

Standby Database: Support by BRARCHIVE



When redo entries are imported in which a structural change of the primary database is recorded, the BRARCHIVE process is terminated with the following ORACLE errors:

ORA-01670: new datafile <file_id> needed for standby database recovery

ORA-01157: cannot identify data file <file_id> - file not found

ORA-01110: data file <file_id>: '<file_name>'

The structural change now needs to be incorporated manually in the standby database. To do this, you can use the command

ALTER DATABASE CREATE DATAFILE '<file_name>';
BRARCHIVE can then be started again.

Standby Database: Backup with BRBACKUP

Standby Database: Backup with BRBACKUP

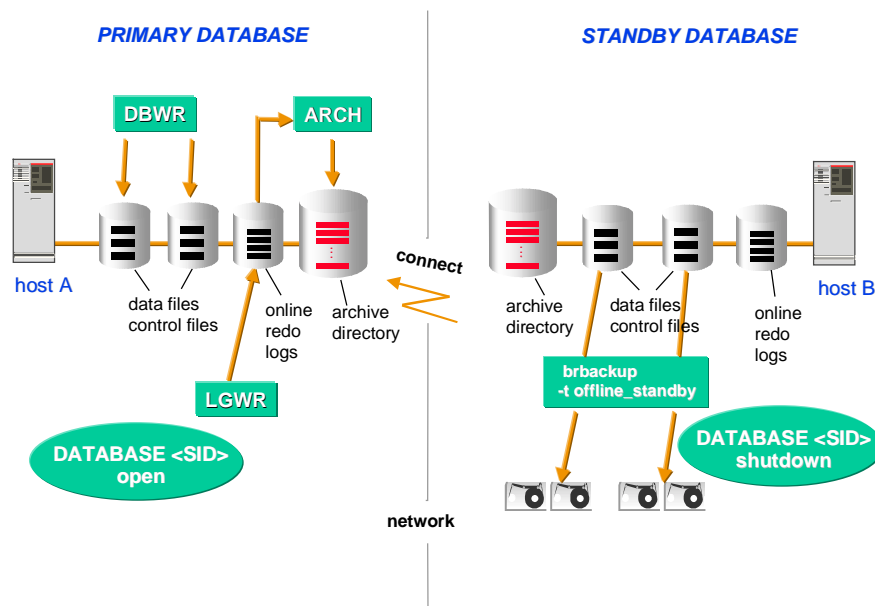
Use

One main advantage of the standby database scenario is that backups do not have to be performed in the primary (production) database system. Instead it allows the datasets of the standby database to be backed up. This means that the database backup does not add to the load on the host of the primary database instance. Since online operation does not occur on the standby database, all the host's resources can therefore be made available for the database backup. The SAP utility program BRBACKUP backs up the standby database data.

Prerequisites

1. The standby instance is in the recovery state and must not be opened. You can make an offline backup only. For the BRBACKUP backup of the standby data, you must set the `init<DBSID>.sap` parameter `backup_type = offline_standby`.
2. Renaming of database files in the standby database is not supported by BRBACKUP. The original names of the primary database need to be retained.
3. As for the OPS configuration, certain requirements must be satisfied for the connection to a remote host.

Features



- BRBACKUP logs on to the primary database instance (entries for the instance string in the `init<DBSID>.sap` parameter `primary_db`) and retrieves the information required on the database structure. This information is entered in the backup logs.
- The standby database instance is stopped.

Standby Database: Backup with BRBACKUP

- BRBACKUP backs up the standby data.
- After the backup has been made, the original state of the standby database instance is recovered. If the database was in a recovery state, this state is restored (ORACLE commands *STARTUP NOMOUNT*, *ALTER DATABASE MOUNT STANDBY DATABASE*).

Activities

To create and configure a standby database you can make a BRBACKUP backup (`backup_type = offline_stop`) of the production database. This is not opened later: Instead it is transferred directly into the mount standby mode, where it takes over the role of the standby database. The backup becomes the production system.

[backup_type \[Page 319\]](#)

[primary_db \[Page 337\]](#)

To recreate a standby configuration, you can make a copy of the production database. You can then use this as the standby database (`backup_dev_type=disk_standby`).

[Structure-Retaining Database Copy \[Page 130\]](#)

Split Mirror Disks Backup

Split Mirror Disks Backup

If you want to make an offline or online backup of your database, whilst avoiding downtime, the 'split mirror disks' configuration is a recommended solution (especially for large databases). From SAP Release 4.0 this configuration is completely supported by the SAP backup tool BRBACKUP.

In the split mirror configuration disks of the primary host, on which the productive database runs, are mirrored. BRBACKUP runs on a secondary host, which carries out the database backup for this 'backup host' after the mirror disks have been split and mounted. This saves the primary (productive) host backup work and the performance of the SAP R/3 System is not affected.

The resynchronization of the 'backup database' with the productive database can be done after the backup using BRBACKUP. If you want to use the backup host as a 'reporting server' you can carry out the synchronization yourself at a later time.

BRBACKUP also communicates with the productive database. By doing this it receives information on the database structure and saves the results of the backup. This enables the CCMS (Computing Center Management System) of the productive R/3 System to monitor the backups.

The BACKINT interface is also supported in the split mirror configuration. This configuration is transparent for BACKINT and no additional aspects have to be taken into account.



The split mirror disks scenario only includes backups with BRBACKUP. BRARCHIVE backups are not included as they do not place any significant load on the productive host. As standard the backup of the offline redo log files is done on local or remote tape devices, on disk or with BACKINT.



If you want to carry out an offline backup the database must be shut down for a few minutes for the splitting of the disks.

[Split Mirror Offline Backup \[Page 194\]](#)

[Split Mirror Online Backup \[Page 193\]](#)

Split Mirror Online Backup

In an online backup (`backup_type = online_split`) the system remains available the whole time. The tablespaces to be backed up are merely set to BACKUP status temporarily while the mirror disks are split. In comparison to the normal online backup, in which the tablespaces retain this status during the whole backup period, the extent of the redo information is significantly reduced.

Online backup in the split mirror disks configuration contains the following steps:

1. Start BRBACKUP on the backup host.
2. BRBACKUP sets the tablespaces to be backed up to the status 'BACKUP' (`ALTER TABLESPACE.....BEGIN BACKUP`).
3. Splitting of the mirror disks (`split_cmd`).
4. BRBACKUP resets the tablespaces to the normal status (`ALTER TABLESPACE....END BACKUP`).
5. Backup of the mirror disks.
6. Synchronization of the mirror disks with the originals (`resync_cmd`). The resynchronization process requires no CPU on the productive machine if, for example, you use an EMC disk system.

All steps (step 6 optionally) are carried out under control of BRBACKUP.

See also:

[Split Mirror Backup: Software Configuration \[Page 195\]](#)

[Description of Parameters \[Page 309\]](#)

Split Mirror Offline Backup

Split Mirror Offline Backup

In an offline backup (`backup_type = offline_split`) the database is shut down for the splitting of the mirror disks. All the following steps (step 5 optionally) are carried out under control of BRBACKUP.

Offline backup in the split mirror disks configuration contains the following steps:

1. Start BRBACKUP on the backup host.
2. Shutdown of the database on the productive host, in order to guarantee the consistency of the database for an offline backup.
3. Splitting of the mirror disks and linking to the backup host (parameter `split_cmd`).
4. Start the database. It is then available for the R/3 System again.
5. Offline backup of the mirror disks on the backup host.
6. Synchronization of the mirror disks with the originals (`resync_cmd`). The resynchronization process requires no CPU on the productive machine if, for example, you use an EMC disk system.

All steps (step 6 optionally) are carried out under control of BRBACKUP.

[Split Mirror Backup: Software Configuration \[Page 195\]](#)

[Description of Parameters \[Page 309\]](#)

Split Mirror Backup: Software Configuration

The ORACLE database software must be installed on the backup host. The Oracle_Home directory structure must match the SAP standard installation. This is expected by BRBACKUP. Standard directories:

UNIX:

\$ORACLE_HOME/dbs (UNIX)
 \$ORACLE_HOME/bin
 \$SAPDATA_HOME/sapbackup
 \$SAPDATA_HOME/sapreorg
 \$SAPDATA_HOME/saparch
 \$SAPDATA_HOME/sapcheck
 \$SAPDATA_HOME/saptrace

WINDOWS NT:

%ORACLE_HOME%\DATABASE
 %ORACLE_HOME%\BIN
 %SAPDATA_HOME%\sapbackup
 %SAPDATA_HOME%\sapreorg
 %SAPDATA_HOME%\saparch
 %SAPDATA_HOME%\sapcheck
 %SAPDATA_HOME%\saptrace

- The ORACLE and BRBACKUP profiles should be available in \$ORACLE_HOME/dbs or %ORACLE_HOME%\DATABASE.
- The parameter `backup_type` in the profile `init<SID>.sap` must be set as follows:
 - `backup_type = offline_split|online_split`
- To set up the link between the backup host and the productive host an instance string must be defined in the profile parameter `primary_db` (SQLNET configuration).
- The directories `/usr/sap/<SID>/SYS/exe/run` or `\\<hostname>\sapmnt\<SID>\SYS\exe\run` must be available on the backup host and should at least contain the programs BRBACKUP, BRCONNECT and BRTOOLS.
- The path names of all databases with access to both hosts, must be identical.
- If you use a backup tool from another manufacturer, then the manufacturer must install and configure the BACKINT interface and the backup client on the backup host.

Init<SID>.sap profile parameter

- [backup_type \[Page 319\]](#) `backup_type = offline_split|online_split`
- `split_cmd = "<split_cmd> [$]"`

`split_cmd` is a program or shell script (with or without options) called by BRBACKUP to split the mirror disks.

- `resync_cmd = "<resync_cmd> [$]"`

`resync_cmd` is a program or shell script called by BRBACKUP to resynchronize the mirror disks. If this parameter is not set then the resync process is not carried out under the control of BRBACKUP.

Split Mirror Backup: Software Configuration

When running BRBACKUP replaces the optional character '\$' with the name of a file containing the names of all files to be backed up.



If the split or resync command was carried out successfully the exit code 0 only and no other output should be returned. Exception: Only messages beginning with '#INFO' are acceptable, that is they are not interpreted as an error message. If a command failed a return code >0 and messages describing the cause of the error should be output.

- `primary_db = <inst_str>`

`inst_str` = Instance string to the productive database, in order to connect the backup host with the productive host. The connections (alias names) are defined in the configuration file `tnsnames.ora` or by the user.

(This parameter was already introduced in connection with the standby DB configuration for SAP Release 3.1).

- `brbackup -t|type =`

`online_split:` Online backup of the mirror disks

`offline_split:` Offline backup of the mirror disks

The command line option overwrites the value set in the parameter `backup_type`.

See also:

[Initialization Profile `init<DBSID>.sap` \[Page 306\]](#)

Backup with Jukebox/Autoloader

For SAP Release 3.0, BRBACKUP supported backups on local disks (`disk`), local (`tape`) and remotely connected (`pipe`) tape devices with manual tape swapping, and on tape devices with automatic tape swapping in fixed sequence (`tape_auto`, `pipe_auto`).

Backup devices such as jukeboxes and autoloaders, which select tapes in any order and mount them automatically, could until now only be addressed with the interface BACKINT interface.

In SAP R/3 Release 4.0, new types of backup device were introduced (`tape_box` for local tape devices and `pipe_box` for remote tape devices). The new commands, defined by the parameters `mount_cmd` and `dismount_cmd`, control the automatic mounting and dismounting of the tapes through the backup device. This means that jukeboxes and autoloaders can administrate several dozen tapes already present in the device.

[Mount / Dismount Command \[Page 198\]](#)

[Example: Backup with an Autoloader \[Page 200\]](#)

Mount / Dismount Command

Mount / Dismount Command

Before accessing a tape or several tapes for the first time the mount command defined in parameter [mount_cmd \[Page 333\]](#) is called up so that the automatic mounting of the tape or tapes can be carried out.

When the backup has been completed the corresponding tape devices are transferred from BRBACKUP/BRARCHIVE into the offline modus (EJECT). After this the dismount command defined in the parameter [dismount_cmd \[Page 330\]](#) is called up to dismount the tapes.



For some devices (e.g. HP 48AL) the tapes may not be transferred into offline mode. The dismount command is then sufficient for the dismounting of the tapes. The `init<SID>.sap` parameter `rewind_offline` should be marked off in these cases.

The mount and dismount commands use special control drivers which are defined in the parameter [tape_address_ctl \[Page 349\]](#) or [tape_address_ctl_arch \[Page 350\]](#).

Syntax of the commands mount/dismount:

```
mount_cmd = "<mount_cmd> $ $ $ [$]"
```

```
dismount_cmd = "<dismount_cmd> $ $ [$]"
```

There are four position options for the mount command and three for the dismount command, represented by the \$ character. The last option can be left out, depending on the implementation.

Meaning of the options \$:

- The first option identifies the database to be backed up and has the following structure:
<DBSID>-A for BRARCHIVE and BRRESTORE with option `-a|-a1|-a2`,
<DBSID>-B for BRBACKUP and BRRESTORE without option `-a|-a1|-a2`.

<DBSID> : ORACLE system ID (database instance)

This option can be used for the identification of a subset of tapes ("subpool") for the database to be backed up. The additional extension "-A" and "-B" enables separate subpools for BRARCHIVE- and BRBACKUP tapes to be defined and administrated.

- The second option identifies the tape devices, on which the mount or dismount operations are to be carried out, through the control driver addresses:

```
<dev_addr1>[,<dev_addr2>,...]
```

<dev_addr> : Driver address of the tape unit

One or more of the tape devices defined in the parameter [tape_address_ctl \[Page 349\]](#) can be spoken to.

- Only for the parameter `mount_cmd`: The third option defines the tape names which are to be mounted on the tape devices stated in the second option:

```
<tape name1>[,<tape name2>,...]|SCRATCH
```

<tape name> : Name of the tape to be mounted

Mount / Dismount Command

The tape names are chosen from the data carrier list [volume_backup \[Page 362\]](#) (from BRBACKUP) or [volume_archive \[Page 361\]](#) (from BRARCHIVE) by the automatic tape administrator. The assignment of the tapes on the tape devices can be defined freely by the mount command. The number of tape names can be bigger than the number of driver addresses. In this case any number of tapes from this list can be mounted on the given tape devices.

The reserved tape name SCRATCH means that any unlocked tapes (in the sense of the BRBACKUP/BRARCHIVE waiting period) can be mounted. Which of the available tapes is mounted is decided by the mount command.

- The last option (the fourth for mount_cmd, the third for dismount_cmd) transfers the name of a parameter file to the mount or dismount command, which contains additional configuration parameters for these commands. The name of the file is defined in the `init<DBSID>.sap` parameter [mount_par_file \[Page 334\]](#) or by the command option `-r|-parfile [Page 384]`.



The user must provide the mount or dismount command in the form of a program, shell script or a batch file.

The successful completion of the command is indicated by the exit code 0 and the absence of any output.

Example: Backup with an Autoloader**Example: Backup with an Autoloader**

This example shows how to integrate an autoloader HP 48AL into the BRBACKUP/BRARCHIVE backup strategy. During this the `mtx` command is used for the mounting and dismounting of the tapes. We assume that the autoloader has 20 slots. The fifth and sixth characters of the band name are used for the number of the slot:

Slot 1-10: tapes C11A01, C11A02,, C11A10.

Slot 11-20: tapes C11B11, C11B12,, C11B20.

The backup of the database C11 with BRBACKUP and the backup of the offline redo log files each require one tape. The automatic tape administration is switched on.

An extract from the profile `initC11.sap` may look like the following:

```
...
backup_dev_type = tape_box
tape_address = /dev/rmt/0mn
tape_address_rew = /dev/rmt/0m
tape_address_ctl = dev/scsi/3
mount_cmd = "mount_tape.csh $ $ $"
dismount_cmd = "dismount_tape.csh $ $"
volume_archive = (C11A01, C11A02, ....., C11A10)
volume_backup = (C11B11, C11B12, ....., C11B20)
...
```

C shell script `mount_tape.csh`:

```
#!/bin/csh -f
set_slot=`echo $3 | cut -b 5-6`
mtx -d $2 -l $slot
echo $slot > $ORACLE_HOME/sapbackup/.slot
```

C shell script `dismount_tape.csh`:

```
#!/bin/csh -f
set_slot=`cat $ORACLE_HOME/sapbackup/.slot`
mtx -d $2 -u $slot
echo $slot > $ORACLE_HOME/sapbackup/.slot
```

These scripts must be feasible and be located correspondingly in the directory `/usr/sap/C11/SYS/exe/run`.

Procedure for a BRBACKUP backup with mount and dismount commands:

After BRBACKUP is started, the automatic tape administration of BRBACKUP chooses a tape (e.g. C11B11) from the data carrier list. Internally the mount command is called up to mount the chosen tape in the tape device:

```
mount_tape.csh C11-B /dev/scsi/3/ C11B11
```

In this script the following `mtx` command is executed:

Example: Backup with an Autoloader

```
mtx -d /dev/scsi/3 -l 11
```

This command causes the tape to be mounted from the 11th slot in the tape device. If the command is carried out without an error message and with exit code 0 BRBACKUP assumes that the tape has been mounted successfully. After a tape label check BRBACKUP starts the backup to the tape.

After the backup has finished the tape is again dismounted in the tape device:

```
dismount_tape.csh C11-B /dev/scsi/3
```

In this script the following mtx command is executed:

```
mtx -d /dev/scsi/3 -u 11
```

This command dismounts the current tape and places it back in slot 11.

The BRARCHIVE backup procedure is analog to the BRBACKUP backup described here. In this case a tape C11A01,....,C11A10 is chosen from the slots 1-10.



For simplification neither the first option (ORACLE-SID) nor the last option (mount_par_file) of the mount or dismount command has been used in the example described here. You can make use of these options in more demanding scripts/programs, e.g. to carry out parallel backups on several tapes, or to backup several databases and form subpools for these in a tape robot.

The loading of the mount and dismount commands is the complete responsibility of the user.

Backing Up the Database with SAPDBA

Backing Up the Database with SAPDBA

Purpose

In the SAPDBA program you can use the *Backup database* menu to back up data files and the control file (and also the online redo log files if you are making an offline backup of the whole database). You must have these files if you have lost data and want to recover it. You can also [back up non-database files and directories \[Page 228\]](#).

The SAPDBA option *Backup database* calls the SAP utility [BRBACKUP \[Page 227\]](#). You can also make a backup directly using the [SAP Utility Programs for Backups \[Page 218\]](#) by calling BRBACKUP yourself.

You can also use the [Computing Center Management System \[Page 24\]](#) to schedule a backup, start it and then view its log. For more information, see the R/3 Library on the *Computing Center Management System (CCMS)*.

Prerequisites

- Read about the technical options of the [SAP Utility Programs for Backups \[Page 218\]](#).
- Familiarize yourself with backup concepts and strategies.
- Remember that the SAP System is not available for production work during an offline backup, since the database is closed. You can make an online backup when the database system and R/3 System are running. Make sure that the database is running in *ARCHIVELOG* mode.
- Make sure the necessary parameters have been set in the initialization profile **init<DBSID>.sap** for BRBACKUP.
- Make sure you have the right backup volumes. When you back up to tape, make sure that you have read the notes on managing and initializing the tapes.
- Allow enough time for the backup: It takes approximately an hour to back up 600-800 MB to tape with a DAT drive (without hardware compression); using this as a basis, schedule around ten hours for backing up a 6 GB database. Using hardware compression can double the backup speed.

Procedure

1. In SAPDBA choose *Backup database* to call the BRBACKUP utility program. The default parameters for BRBACKUP, which are set in the initialization profile **init<DBSID>.sap**, trigger an **offline full** database backup to a **local tape device** (tape size 1200 MB) **without file compression**. This means that the online redo log files and a control file are backed up as well as the data files.
2. If necessary, change these defaults in the menu [SAPDBA: Backup Database \[Page 204\]](#). SAPDBA only allows you to change some of the parameters in **init<DBSID>.sap**. If you have to make other changes, you will have to change the initialization profile manually and then restart SAPDBA.
3. Select *Start BRBACKUP* to start the backup. If the backup is being made locally or remote to tapes or disks, then the backup is monitored and an estimation is made of the backup time, based on time elapsed and the size of the files that still have to be backed up. You also see success or error messages.

Backing Up the Database with SAPDBA

4. Check the summary log (**back<DBSID>.log**) to see which return code was sent when the backup was completed. For information on the progress of the backup, see the detail log (**b<encoded timestamp>.<ext>**). See [BRBACKUP Logs \[Page 438\]](#)


SAPDBA: Backup Database

SAPDBA: Backup Database

a	- Backup function	Normal backup
b	- Parameter file	initC11.sap
c	- Backup device type	tape
d	- Objects for backup	all
e	- Backup type	offline
f	- Backup volume(s)	
g	- Query only	no
h	- Special options	
i	- Standard backup	yes
j	- Backup from disk backup	
k	- Restart backup	
l	- Make partial backup complete	
s	- Start BRBACKUP	
q	- Return	

Choose the settings you want from the sub-menu. Use *Return* to leave the sub-menu in order to set any other parameters necessary for your chosen operation and to start the operation.

- **Backup function**

- **Normal backup:** The volume (tape) is not initialized (default). You can start the backup.
- **Initialize BRBACKUP tape:** Use this option to initialize a volume .
 - Check label before initialization no*
The label of the volume is not checked. You can initialize new non-SAP volumes or make a new label for previously initialized volumes.
 -  The expiration period is not checked.
 - Check label before initialization yes*
The label of the volume is checked. You can only re-initialize SAP volumes that have expired. (This option is often used to rename volumes.)
 - Number of tapes 1*
Specify the number of volumes that you want to initialize.
- **Show BRBACKUP tape header information:** The information in the volume label is displayed.
- **Determine compression rates:** The compression rates of all data files specified with *Objects for backup* are determined. Note that you need to allow a certain amount of time for this procedure. SAP recommends that you use this option about once a month, if you want to perform backups on tape devices with hardware compression. This is so that the current compression rate of the files can be determined (see [compress \[Page 320\]](#)).
- **RMAN preparation run:** Preparation run for a backup using RMAN (see [Saveset Grouping \[Page 167\]](#) and [Tape Layout \[Page 169\]](#)).
- **Verify BRBACKUP tape:** Check backup to tape. Use the menu option **Select verification type** to determine the extent of the check.

SAPDBA: Backup Database

- **Reading tape(s):** Only the readability of the tapes is checked, and not the contents (with BRRESTORE [-w|-verify \[Page 433\]](#)). The files are only read, not restored.
- **Restore to stage and DB VERIFY:** The backup is restored to the directory [compress_dir \[Page 322\]](#); the block structure is checked with the Oracle tool DB_VERIFY (brrestore -w use_dbv).

See [Backup Verifications \[Page 155\]](#) and [Verification of RMAN Backups \[Page 157\]](#).

- **Parameter file:** Default: `init<DBSID>.sap`
To use a different initialization profile for BRBACKUP, specify the name of the profile you want to use.
- **Backup device type:**
 - **local tape:** Backup to a local tape device (default: corresponds to `backup_dev_type = tape`).
 - **local tape auto changer:** Backup to tape when using a device with automatic tape changer; any messages telling you to change the tape are suppressed. (corresponds to `backup_dev_type = tape_auto`)
 - **local tape juke box:** Backup to jukeboxes or autoloaders that can be addressed locally (corresponds to `backup_dev_type = tape_box`).
 - **remote tape:** Backup to a tape device in a remote system, using the command specified in the initialization profile (corresponds to `backup_dev_type = pipe`).
 - **remote tape auto changer:** Backup to a device with automatic tape changer using a remote system; any messages telling you to change the tape are suppressed. (corresponds to `backup_dev_type = pipe_auto`)
 - **remote tape juke box:** Backup to jukeboxes or autoloaders that can be addressed remotely (corresponds to `backup_dev_type = pipe_box`).
 - **external backup tool (backint):** Backup using a backup program addressed by the interface program BACKINT (corresponds to `backup_dev_type = util_file`), see [Using External Backup Programs \[Page 241\]](#).



If you want to perform the backup using an external backup program, then you may have to specify the necessary parameters in a parameter file. Enter this file name under **Backup utility parameter file**.

- **external backup tool (backint) online:** File-by-file backup using a backup program addressed by the interface program BACKINT; in an online backup, the *BACKUP* status is set and completed dynamically for the tablespaces being backed up. This significantly reduces the volume of offline redo log files during an online backup (corresponds to `backup_dev_type = util_file_online`).
- **external backup tool (backint) with recovery manager:**
(corresponds to `backup_dev_type = rman_util`)
See [Backups with External Backup Libraries \[Page 162\]](#).
- **local disk:** Backup to a locally installed or mounted disk (corresponds to `backup_dev_type = disk`).

SAPDBA: Backup Database

- **remote disk:** Backup to a remote disk; this option is recommended if no disk is installed locally, since the transfer protocol is secure (*ftp* or *rcp*), and no verification is required, unlike an NSF mount (corresponds to `backup_dev_type = stage`).
- **database copy to disk:** Use this option to make a copy of the database files, including the directory structure, for example when setting up a test system (corresponds to `backup_dev_type = disk_copy`).
See [Structure-Retaining Database Copy \[Page 130\]](#)
- **Objects for backup:** The following entries are possible (See [m\]-mode \[Page 378\]](#)):
 - **all:** Backs up the whole database (default).
 - **sap_dir/ ora_dir:** Backs up the SAP directories and the Oracle non-database directories.
 - **tablespace name:** Backs up the specified tablespace.
 - **File ID or File ID1 - File ID2:** Backs up one or more files, which are specified using an Oracle File ID or a File ID area.
 - **Generic path:** You can specify an arbitrary file by using its full path name or several data files by specifying a generic path.
 - **Object list:** You can specify a combination of objects by separating the individual objects with commas.
- **Backup type:**
 - **online:** (*ARCHIVELOG* mode must be activated). The backup is made with the database still running.
 - **offline:** If the R/3 System is shut down, the database is also shut down for the backup. Otherwise, the database is not shut down and BRBACKUP terminates with an error message (default).
 - **online consistent:** As well as the database files, the offline redo log files generated during this online backup are also copied to the same volume. This makes the data logically consistent.
 - **offline (force):** The database is shut down and an offline backup is made. The system does not check whether R/3 System users are active.
 - **offline (standby):** The standby database is shut down during the backup.
See [Standby Database: Backup with BRBACKUP \[Page 190\]](#)
 - **online (split):** The mirror disks are split and backed up while the database is running. The tablespaces to be backed up are only placed in *BACKUP* status during the split.
See [Mirror Disk Backup \[Page 192\]](#)
 - **offline (split):** The split mirror disks are backed up. The database is only shut down for a short time for the mirror disks split.
See [Mirror Disk Backup \[Page 194\]](#) and the parameter [backup_dev_type \[Page 314\]](#)
- **Backup volume(s):** The IDs of the volumes (tapes), which you want to use for the backup. If you specify more than one volume name, then you must separate the names using commas.
- **Query only:** (See [-q\]-query \[Page 383\]](#))

- **no**: Backup is started (default).
- **with tape check**: No backup; you are told which tapes you have to mount for the backup, and a check is made on whether you have actually mounted these tapes.
- **without tape check**: No backup; you are told which tapes you have to mount for the backup.
- **Special options**: Choose this menu option to go to a sub-menu where you can choose other special functions for backing up the database.
See [Backup Database: Special Options \[Page 208\]](#)
- **Standard backup**: Default: yes
The display tells you whether you are making a normal backup, or whether you are using a special BRBACKUP function. If you are making a [two-phase backup \[Page 124\]](#) (backup from disk backup) the display switches to *no*.
- **Backup from disk backup**: Choose this option if you want to back up a disk backup to tape ([two-phase backup \[Page 124\]](#)). The display *Standard backup* is set to *no* automatically.
- **Restart backup**: Use this function to complete a cancelled backup. Choose one of the backups on offer. The default is the latest backup.
See: [-f|-fillup \[Page 373\]](#)
- **Make partial backups complete**: Use this function to complete partial backups. Specify the number of previous days for which you want to complete partial backups (default: 0).
See [-f|-fillup \[Page 373\]](#).

SAPDBA: Backup Database: Special Options

SAPDBA: Backup Database: Special Options

In the submenu *Special options* you can choose other functions for backing up the database;

<i>a</i>	- Confirm backup parameters	<i>no</i>
<i>b</i>	- Compress	<i>hardware</i>
<i>c</i>	- Verification after backup	<i>no</i>
<i>e</i>	- Enter password interactively	<i>no</i>
<i>f</i>	- Level of parallel execution	<i>0</i>
<i>g</i>	- Append archive backup	<i>no</i>
<i>k</i>	- Show all objects for backup	
<i>l</i>	- Language	<i>English</i>
<i>q</i>	- Return	

- **Confirm backup parameters**

- **no:** (Default) Some of the confirmation messages of BRBACKUP are suppressed (see [-c|-confirm \[Page 369\]](#)).
- **yes:** All the confirmation messages are displayed.

- **Compress**

- **no:** No compression (default).
- **yes:** Software compression.
- **hardware:** Set this option if your tape device can perform hardware compression.

- **Verification after backup**

- **no:** No verification (default).
- **yes:** Verification. Use this function to check the readability and the completeness of the backup (see [-w|-verify \[Page 389\]](#)).

- **Enter password interactively**

- **no:** You cannot enter the password interactively (default).
- **yes:** You must enter the password interactively. The password is not visible when you enter it and does not appear in the process list.

- **Level of parallel execution**

Maximum number of parallel copy processes; the default 0 means that the number of parallel processes matches the number of backup devices.

- **Append archive backup**

- **no:** The database is backed up using BRBACKUP.
- **yes:** After the database has been backed up using BRBACKUP, BRARCHIVE is started to back up the offline redo log files to the same backup tape. Use the menu option **Configure archive backup** to make other settings for the backup of the offline redo log files (see [SAPDBA: Backup Offline Redo Lologs \[Page 212\]](#)). Any options that do not need to be set

SAPDBA: Backup Database: Special Options

are hidden (for example, *Archive device type*, *Archive volume(s)*, and so on).

[BRBACKUP and BRARCHIVE Backups in One Run \[Page 136\]](#)

- **Show all objects for backup**
All objects from `backup_mode` are displayed.
- **Language**
Choose between:
 - **English:** BRBACKUP messages appear in English (default).
 - **deutsch:** BRBACKUP messages appear in German.

Archiving the Offline Redo Log Files

Archiving the Offline Redo Log Files

Purpose

Oracle backs up the online redo log files for the database system in the archiving directory `<ORACLE_HOME>/saparch`, so creating the offline redo log files. If you need to recover your database you must have all offline redo log files that have been written since the recovery backup; otherwise you can only recover the database to the point in time before any missing redo log entry. You must always archive the offline redo log files after a database backup, and immediately after an online backup. In production systems, make two copies of the offline redo log files; in test systems one copy is enough.

The Oracle database will hang if the database is operated in *ARCHIVELOG* mode but the archiving process *ARCH* cannot save the online redo log files because the archiving directory is full. This situation is called **Archiver Stuck**. If this occurs you must shut down the database with `shutdown abort`; the subsequent startup of the database takes a long time, since Oracle must perform an instance recovery. To avoid this, back up the offline redo log files regularly to tape. How often you do this depends on the amount of activity in your R/3 System. If a lot of redo log entries are written, and the redo log files are therefore switched frequently, you should constantly monitor the archiving directory and archive and delete the offline redo log files when necessary.

Prerequisites

You can archive the offline redo log files regardless of the current status of the database and the SAP System. If you have configured your system according to the SAP recommendations, the Oracle database system will save the online redo log files (where the changes made to the database are recorded) automatically (offline redo log files are written). If you have not changed the standard profile `init<DBSID>.ora`, the offline redo log files are located in the archive directory `<ORACLE_HOME>/saparch` (see [Archiving Parameters \[Page 34\]](#)).

You can also directly archive the offline redo log files by calling the right [command option for BRARCHIVE \[Page 390\]](#). Read about the options in the [SAP Backup Utilities \[Page 218\]](#), especially [using BRBACKUP and BRARCHIVE \[Page 219\]](#). Make sure the necessary parameters have been set for BRARCHIVE in the the profile `init<DBSID>.sap`.

Procedure

1. Select the SAPDBA option *Backup archive logs* to call the BRARCHIVE program. Based on the BRARCHIVE parameters configured in the `init<DBSID>.sap` profile, archiving is started with the following defaults:
 - simple archiving
 - all offline redo log files
 - to a local tape device, tape size 1200 MB
 - without file compression
2. You can change these default values if necessary. SAPDBA only allows you to change some of the parameters in `init<DBSID>.sap`. If you have to make other changes, you will have to change the initialization profile manually and then restart SAPDBA.
3. Start BRARCHIVE.

Result

Check the summary log (`arch<DBSID>.log`) to see which return code was sent when the archiving was completed. For information on the progress of the archiving, see the detail log (`a<encoded timestamp>.<ext>`).

See [BRARCHIVE Logs \[Page 445\]](#)

Example

[Sample Logs \[Page 454\]](#)

SAPDBA: Backup Offline Redo Logs

SAPDBA: Backup Offline Redo Logs

a	- Archive function	Save offline redo logs
b	- Parameter file	initC11.sap
c	- Archive device type	tape
d	- Number of redo logs	10000
e	- Archive volume(s)	
f	- Query only	no
g	- Fill tape(s) permanently	no
h	- Special options	
i	- Standard backup of offline redo logs	yes
j	- Use disk backup	no
q	- Return	
s	- Start BRARCHIVE	

- **Archive function:** Choose one of the following entries (Use *Return* to leave the submenu in order to set any other parameters necessary for the chosen operation and to start the operation):
 - **Save offline redo logs:** Archives the offline redo log files that have not yet been backed up (default).
 - **Second copy of offline redo logs:** Creates a second copy of the offline redo log files.
 - **Double save (parallel) offline redo logs:** Archives the offline redo log files that have not yet been saved, to two tape devices in parallel.
 - **Save and delete offline redo logs:** Archives the offline redo log files and then deletes these files.
 - **Second copy and delete offline redo logs:** Creates a second copy of offline redo log files and then deletes these files from the archiving directory.
 - **Double save and delete offline redo logs:** Archives the offline redo logs that have not yet been backed up, to two tape devices in parallel, and then deletes the files from the archiving directory.
 - **Delete saved offline redo logs:** Deletes the offline redo log files that have already been archived (the oldest backed up offline redo log files is deleted first).
 - **Delete copied offline redo logs:** Deletes offline redo log files that have already been archived twice.
 - **Copy and save offline redo logs:** Creates a second copy of offline redo log files that have already been archived and then archives the offline redo log files that have been created in the meantime.
 - **Copy, delete and save offline redo logs:** Creates a second copy of offline redo log files that have already been archived and then deletes them and archives the offline redo log files that have been created in the meantime.
 - **Initialize BRARCHIVE tape:** Use this option to initialize a volume.

Check label before initialization no

SAPDBA: Backup Offline Redo Logs

The label of the volume to be used is not checked. You can initialize new non-SAP volumes or make a new label for previously initialized volumes. (Important: The expiration period is not checked.)

Check label before initialization yes

The label of the volume to be used is checked. You can only re-initialize SAP volumes that have expired. (This option is often used to rename volumes.)

Number of tapes 1

Specify the number of volumes that you want to initialize.

- **Show BRARCHIVE tape header information:** The information in the volume label is displayed.
- **Stop BRARCHIVE run with fill option:** Use this option to stop a BRARCHIVE backup run that was started with the option *Continue until end of tape safely*.

BRARCHIVE uses its log files (mainly the summary log) to identify offline redo log files that have already been archived and/or deleted. Offline redo log files are deleted only if they have been successfully archived.

- **Parameter file:** Default `init<DBSID>.sap`
To use a different initialization profile for BRARCHIVE, specify the name of the profile you want to use for the backup.
 - **Archive device type:** Choose one of the following entries (use *Return* to leave the submenu in order to set any other parameters necessary for the chosen operation and to start the operation):
 - **local tape:** Archiving to a local tape device (default: `backup_dev_type = tape`).
 - **local tape juke box:** Archiving to jukeboxes or autoloaders that can be addressed locally (`backup_dev_type = tape_box`).
 - **remote tape:** Archiving to a remote system with the command specified in the backup profile (`backup_dev_type = pipe`).
 - **remote tape juke box:** Archiving to jukeboxes or autoloader tape devices that can be addressed locally (`backup_dev_type = pipe_box`).
 - **external backup tool (backint):** File-by-file archiving using a backup program addressed by the BACKINT interface program (`backup_dev_type = util_file`). See [Using External Backup Programs \[Page 241\]](#)
Backup utility parameter file: If you want to make the backup using an external backup program, then you must specify the necessary parameters in a parameter file.
 - **local disk:** Archiving to a locally installed or mounted disk (`backup_dev_type = disk`, especially when using the Oracle Parallel Server).
 - **remote disk:** Backup to a remote disk; this option is recommended if no disk is installed locally, since the transfer protocol is secure (*ftp* or *rcp*), and no verification is required, unlike an NSF mount (corresponds to `backup_dev_type = stage`).
- See also parameter `init<DBSID>.sap` [backup_dev_type \[Page 314\]](#)
- **Number of redo logs:** Default: 10000

SAPDBA: Backup Offline Redo Logs

Enter the number of redo log files that you want to process.



Archive function *Save and delete offline redo logs, Number of redo logs 2:*

BRARCHIVE archives and deletes the two oldest offline redo log files that have not yet been backed up to tape.

Archive function *Save and delete offline redo logs, Number of redo logs 10.000:*

BRARCHIVE archives and deletes all offline redo log files in the archiving directory.

- **Archive volume(s):** IDs of volumes (tapes) that you want to use for the archiving; if you want to specify more than one volume name then you must separate them with commas.
BRARCHIVE determines how many offline redo log files fit onto a volume. If the files do not all fit onto one volume, then you will have to restart BRARCHIVE for the rest of the files.
- **Query only:** Choose one of the following entries (see [-q|-query \[Page 383\]](#)); use *Return* to leave the submenu in order to set any other parameters necessary for the chosen operation and to start the operation):
 - **no:** Archiving is started (default).
 - **with tape check:** No archiving; you are told which tapes must be mounted for the archiving process, and a check is made on whether you have actually mounted these tapes.
 - **without tape check:** No archiving; you are told which tapes must be mounted for the archiving process.
- **Fill tape(s) permanently:** Use this function to decide whether the offline redo log files are to be archived individually or in groups.
 - **no:** An offline redo log file is archived as soon as it is created. This process is not stopped until the volume is full or the number of redo log files specified under **number of redo logs** has been backed up.
 - **<Number>:** Number of offline redo log files that you want to archive together as a group; BRARCHIVE stops archiving until this number of offline redo log files is reached. [Grouping Offline Redo Log Files \[Page 217\]](#), [-f|-fill \[Page 396\]](#)
- **Special options:** The sub-menu [SAPDBA: Backup Offline Redo Log Files - Special Options \[Page 215\]](#) appears.
- **Standard backup of offline redo logs:** The display tells you whether you are making a regular backup of the offline redo log files, or whether you are using a special BRARCHIVE function. If you are making a [two-phase backup \[Page 124\]](#) (**Use disk backup**) the display switches to *no*.
- **Backup from disk backup:** Choose this option to archive the offline redo log files from a disk backup. The display **Standard backup of offline redo logs** is set to *no* automatically.

For detailed information on the SAP utility BRARCHIVE, see [SAP Backup Utilities \[Page 218\]](#) and [Initialization Profile init<DBSID>.sap \[Page 306\]](#).

SAPDBA: Backup Offline Redo Logs: Special Options

In the submenu *Special options* you can choose other functions for archiving the offline redo log files:

<i>a</i>	- <i>Confirm archive parameters</i>	<i>no</i>
<i>b</i>	- <i>Compress</i>	<i>hardware</i>
<i>c</i>	- <i>Verification after backup</i>	<i>no</i>
<i>e</i>	- <i>Enter password interactively</i>	<i>no</i>
<i>f</i>	- <i>Append database backup</i>	
<i>h</i>	- <i>Apply logs into standby DB</i>	<i>don't apply</i>
<i>l</i>	- <i>Language</i>	<i>English</i>
<i>q</i>	- <i>Return</i>	

- **Confirm archive parameters:** The following entries are possible:
 - **no:** Some of the confirmation messages of BRBACKUP are suppressed (default) (see [-cl-confirm \[Page 369\]](#)).
 - **yes:** All the confirmation messages are displayed.
- **Compress:** Choose one of the following entries:
 - **no:** No compression (default).
 - **yes:** Software compression.
 - **hardware:** Set this option if your tape device can perform hardware compression.
- **Verification after backup:** The following entries are possible:
 - **no:** No verification (default).
 - **yes:** Verification. You can use this function to check the readability and the completeness of the backup of the offline redo log files (see [-w|-verify \[Page 412\]](#)).
- **Enter password interactively:** The following entries are possible:
 - **no:** You cannot enter the password interactively (default).
 - **yes:** You must enter the password interactively. In this case, the password does not appear in the process list, nor is it visible on the screen when entered.
- **Append database backup:** The following entries are possible:
 - **no:** Backup of the offline redo log files using BRARCHIVE.
 - **yes:** After the offline redo log files have been backed up with BRARCHIVE, a database backup is made with BRBACKUP. Use the menu option *Configure database backup* to make other settings for the database backup, (see [SAPDBA: Backup Database \[Page 204\]](#)). Any options that do not need to be set are hidden (for example, *Backup device type*, *Backup volume(s)*, and so on).

See also [BRBACKUP and BRARCHIVE Backups in One Run \[Page 136\]](#)
- **Apply logs into standby DB:** The following entries are possible:

SAPDBA: Backup Offline Redo Logs: Special Options

- **no**: If you do not have a standby database, or if you do not want to import the redo log files into the standby database (default).
- **<n>**: BRARCHIVE waits for this period of time (minutes) before it imports the redo log files into the standby database (see [-m|-modify \[Page 401\]](#)).
- **Language**: The following entries are possible:
 - **English**: BRARCHIVE messages appear in English (default).
 - **Deutsch**: BRARCHIVE messages appear in German.

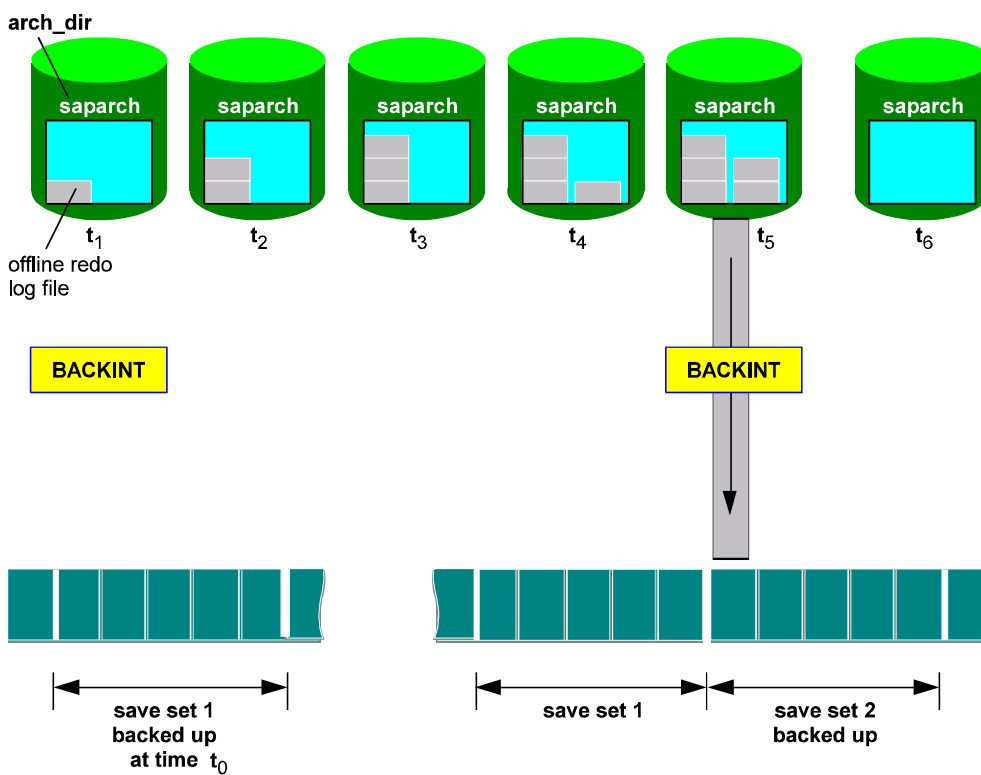
Grouping the Offline Redo Log Files

With permanent backup of the offline redo log files using BRARCHIVE (`brarchive -f|-fill`) you prevent a possible overflow of the archiving directory. You can also collect a certain number of offline redo log files before BRARCHIVE backs them all up together on tape (`brarchive -f <number>`).



`brarchive -sd -d util_file -f 5`

`brarchive -f 5 -d util_file`



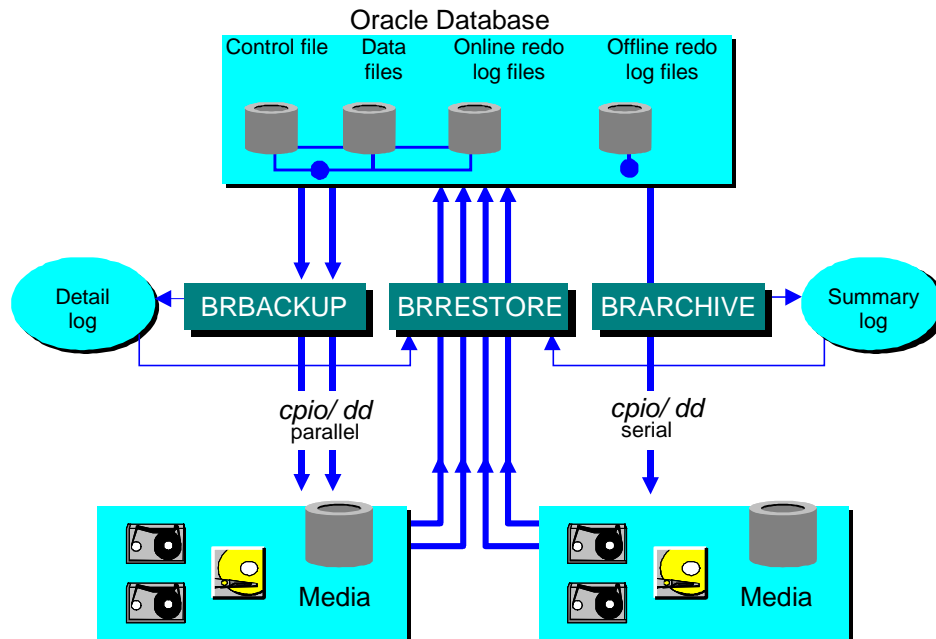
When you use an external backup program via the Backint interface (see [Using External Backup Programs \[Page 241\]](#)), you must use the option to group offline redo log files. In this case every archiving action entails the activation of the interface and often the repositioning of the tape. This usually leads to the creation of a new 'Save Set'. A reduction in the number of Save Sets by grouping the offline redo log files is beneficial for backup time.

See: [-f|-fill \[Page 396\]](#).

SAP Backup Utilities

SAP Backup Utilities

SAP offers the utility programs BRBACKUP, BRARCHIVE and BRRESTORE. Use these programs to realize the recommended backup concepts. Each of these programs has its own range of functions (backup, archiving the redo log files, restore). In addition, the BR programs also have their own utility programs, BRCONNECT and BRTOOLS. These are not called by the database user, the BR programs call them themselves for a particular task.



For more detailed information on the BR utility programs, and their own utility programs, see the relevant section:

[Using BRBACKUP and BRARCHIVE \[Page 219\]](#)

[BRBACKUP \[Page 227\]](#)

[BRARCHIVE \[Page 233\]](#)

[BRRESTORE \[Page 236\]](#)

[BRCONNECT and BRTOOLS \[Page 240\]](#)

Using BRBACKUP and BRARCHIVE

The SAP utilities BRBACKUP/BRARCHIVE can be called directly from the UNIX command level, using the SAPDBA menu (see [Backing up the Database \[Page 202\]](#), [Archiving the Offline Redo Logs \[Page 210\]](#)), or from the R/3 System through the CCMS (see the online help of the *Computing Center Management System*).

BRBACKUP/BRARCHIVE do not use a graphical user interface in a UNIX environment. You can run them in any UNIX window and under any shell.

Both programs use standard commands for backing up the relevant files on a volume:

- `cpio` for backup to tape, `cp` for backup to disk if you are working with file systems.
- `dd` for backup of raw devices on tape or disk

BRBACKUP/BRARCHIVE must be configured. Some of the parameters for the initialization profile `init<DBSID>.sap` (see [Description of Parameters \[Page 309\]](#)) are described below.

The default configurations of both programs require online user interaction in several places. You can also run the programs without interaction - see [Unattended Backup \[Page 139\]](#).

- BRBACKUP saves objects in the database (in accordance with the selected options):
 - Data files of the database
 - Control file
 - Online redo log files

You can also use BRBACKUP to back up non-database files and directories. See [Backing up Non-Database Files and Directories \[Page 228\]](#).

- BRARCHIVE normally archives the online redo logs that ORACLE backs up in the archiving directory (offline redo log files) to tape. It is also possible to back up the offline redo log files to disk for special purposes (for example in a parallel server configuration).
- BRBACKUP/BRARCHIVE also backs up the following files to the volume (when database objects or offline redo log files are backed up) or to the first directory named in `backup_root_dir` for a backup of database files to disk:
 - A copy of profile `init<DBSID>.ora`, `init<DBSID>.dba` (OPS: `init.ora` / `init.dba` profiles of all the involved ORACLE instances)
 - A copy of profile `init<DBSID>.sap`
 - The main log of SAPDBA `reorg<DBSID>.log`, the structure log `struct<DBSID>.log` (OPS: script `create_raw_links`)
 - The detail BRBACKUP/BRARCHIVE log
 - The summary BRBACKUP/BRARCHIVE log (OPS: the summary logs of all the involved ORACLE instances)

The logs are saved on every volume (e.g. tape) used for the backup. As a result, you can still determine the contents of the volume, even when the BRBACKUP/BRARCHIVE database and file system logs have been lost.

- BRCONNECT monitors the database during a BRBACKUP process. See [Monitoring the Database Status with BRCONNECT During a BRBACKUP Backup \[Ext.\]](#). BRTOOLS is

Using BRBACKUP and BRARCHIVE

a utility that is internally called by BRBACKUP and BRARCHIVE. See [BRTOOLS \[Page 240\]](#). Neither program is called directly by the user; they are only called by the appropriate utilities.

[Supported Backup Media \[Page 221\]](#)

[Automatic Tape Changing \[Page 222\]](#)

[Profile init<DBSID>.sap \[Ext.\]](#)

Supported Backup Media

You can use BRBACKUP/BRARCHIVE for direct backup to the following media:

- Local tape devices:
`backup_dev_type = tape`
- Remote tape devices:
`backup_dev_type = pipe`
- Local disks and those mounted via NFS:
`backup_dev_type = disk`
- Tape devices with automatic tape changing (tape stacker, for example). These backup devices can be addressed:
 - locally with `backup_dev_type = tape_auto`
 - remotely with: `backup_dev_type = pipe_auto`
- Jukebox and autoloader. These backup devices can be addressed:
 - locally with `backup_dev_type = tape_box`
 - remotely with `backup_dev_type = pipe_box`

See [Automatic Tape Changing \[Page 222\]](#).

See [Backup with Jukebox/Autoloader \[Page 197\]](#)

BRBACKUP/BRARCHIVE support:

- Automatic volume management. See [Volume Management \[Page 142\]](#).
- An open interface to interface program BACKINT, to enable backups using external backup programs. See [Using External Backup Programs \[Page 241\]](#).

Using the parameter [tape_copy_cmd \[Page 353\]](#) you can choose whether files (not on raw devices) are copied from disk to tape with cpio or dd.

Automatic Tape Changing

Automatic Tape Changing

BRBACKUP supports backup on tape devices with automatic tape changing.

Some devices with automatic tape changing require additional time to change the tape. This additional time can be taken into account by adjusting the definition of parameter `rewind_offline` in profile `init<DBSID>.sap`.



Settings (operating system specific):

- Device type:

```
backup_dev_type = tape_auto|pipe_auto
```

- Addresses for tape devices

```
tape_address = /dev/rmt/0hn
```

```
tape_address_rew = /dev/rmt/0h
```

- Rewind set offline parameter:

```
rewind_offline = "mt -t $ offline && sleep 30"
```

- Tape size:

```
tape_size = 1600M
```

Effects of the Command Options

A description of all command options can be found in [Command Options for BRBACKUP, BRARCHIVE, and BRRESTORE \[Page 363\]](#).

Several parameters of the `init<DBSID>.sap` profile can be overridden; simply call BRBACKUP/BRARCHIVE with the appropriate command option.

The following list displays the corresponding profile parameters and the options that override them.

Parameter	Command option	SAP utility that can use the option
<code>archive_function</code>	<code>-s</code> <code>-save</code> <code>-cs</code> <code>-copy_save</code> (and others)	BRARCHIVE
<code>backup_mode</code>	<code>-m</code> <code>-mode</code>	BRBACKUP, BRRESTORE
<code>backup_type</code>	<code>-t</code> <code>-type</code>	BRBACKUP
<code>backup_dev_type</code>	<code>-d</code> <code>-device</code>	BRBACKUP, BRARCHIVE, BRRESTORE
<code>compress</code>	<code>-k</code> <code>compress</code>	BRBACKUP, BRARCHIVE, BRRESTORE
<code>exec_parallel</code>	<code>-e</code> <code>-execute</code>	BRBACKUP, BRRESTORE
<code>volume_archive</code>	<code>-v</code> <code>-volume</code>	BRARCHIVE
<code>volume_backup</code>	<code>-v</code> <code>-volume</code>	BRBACKUP
<code>util_par_file</code>	<code>-r</code> <code>-parfile</code>	BRBACKUP, BRARCHIVE, BRRESTORE

cpio Continuation Tape

cpio Continuation Tape

If the volume size is defined too large, the cpio command may reach the physical end of the volume. In such cases, cpio issues a message displaying the end of the volume (`end of tape...`, `end of medium...`, `end of volume...`).

You can continue the backup on another volume or cancel the backup. Please note that the cpio continuation mechanism is only possible during serial backups. If a parallel backup to several tape devices is involved, the backup is terminated for the relevant tape device when the end of tape is reached, but the backup is continued on the other tape devices. In such cases, it is essential to make sure that all the files were saved.

Make sure that the continuation tape is not one of the volumes initialized for BRBACKUP/BRARCHIVE, since no label check takes place for the continuation volume.

The cpio continuation tape is not “visible” for BRBACKUP/BRARCHIVE, e.g. it is regarded as one logical tape together with the first one. In restore situations, SAPDBA will therefore request one tape; however, both tapes must be mounted.



To avoid the situation described above, none of the database files should be larger than the size specified in `tape_size` (after compression, when applicable).



Do not confuse the cpio continuation mechanism with the management of tape continuation, which BRBACKUP operates when another tape is required to back up the files.

cpio Error

cpio may report an I/O error:

I/O Error on output.

A detailed overview of the possible error messages can be found in the description of message BR278 in [Messages \[Page 495\]](#) or in the SAPNet Notes database, if you enter cpio as selection criteria.

If this error only occurs sporadically, you should check whether it might be related to the use of a specific volume. If so, you should no longer use that volume for backups.

If the error occurs with different volumes, you should have your backup device inspected, as a hardware problem is probably to blame.

Canceling a Backup

Canceling a Backup

You can cancel the backup processes. You can use:

- The key combination `CTRL-C` and enter `stop` (entering `cont` to continue the backup is then only possible with restrictions)
- The UNIX command `kill` (but **never** `kill -9`)
- The command `brarchive -fill stop` to properly stop the archiving of offline redo log files started with `brarchive -f`. Do not use `CTRL-C` or a `kill` command to stop such `BRARCHIVE` runs, as this may terminate an active copy process.

BRBACKUP

The following contains advice specifically for BRBACKUP. Further information about this utility can be found in [Using BRBACKUP and BRARCHIVE \[Page 219\]](#).

The smallest unit that can be saved with BRBACKUP is a file. BRBACKUP can be used for backing up both files in the database and non-database files and directories. Use the `init<DBSID>.sap` parameter `backup_mode` (see [backup_mode \[Page 316\]](#)) or command option `brbackup -m|-mode` (see [-m|-mode \[Page 378\]](#)) for this purpose.

[Backing up Non-Database Files and Directories \[Page 228\]](#)

[Backing up Database Files \[Page 231\]](#)

[Examples \[Ext.\]](#)

[Monitoring the Database Status with BRCONNECT During a BRBACKUP Backup \[Ext.\]](#)

[BRTOOLS \[Page 240\]](#)

[Hardware Compression \[Page 232\]](#)

[Logging \[Ext.\]](#)

[Completion of BRBACKUP Backups \[Page 229\]](#)

Backing up Non-Database Files and Directories

Backing up Non-Database Files and Directories

Non-database files should only be backed up with BRBACKUP after an SAP upgrade or an ORACLE migration. This backup method is not a replacement for a file system backup using operating system features.

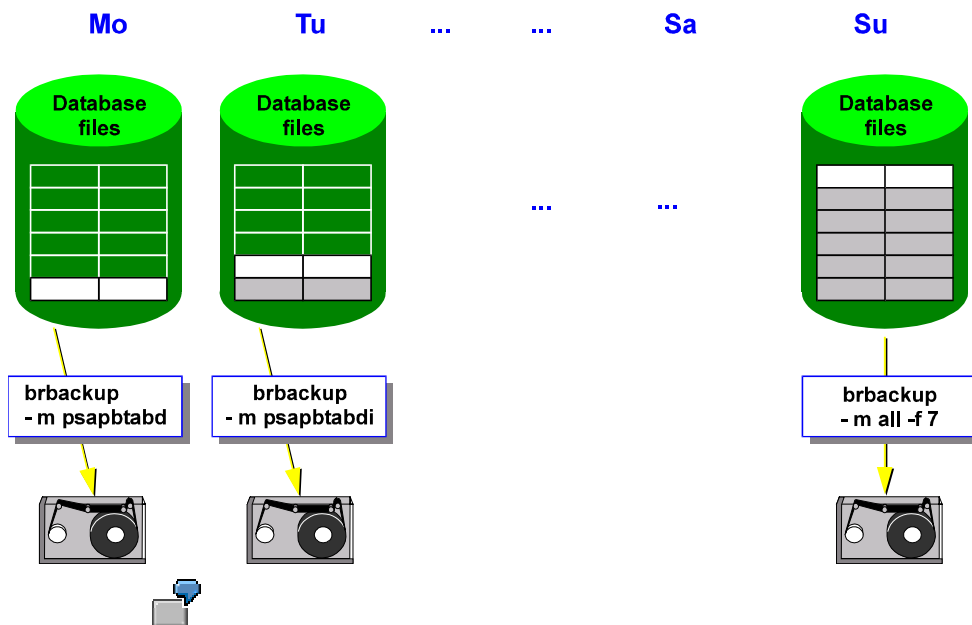
- Directories and non-database files for backup must be specified with their complete paths.
- If you use BRBACKUP to save directories, only the files in that directory are saved; files of any existing subdirectories will not be saved. The use of parameter `sap_dir` or `ora_dir` is an exception, as it is possible to make a backup of all the non-database files of the SAP or ORACLE environment with it.
- When you save directories with software compression (`compress = yes`), their contents are not compressed.
- If you want to back up a large number of non-database files and directories (all the SAP executables and profiles, for example), we recommend carrying out this backup separately.

See also [backup_mode \[Page 316\]](#) or [-m|-mode \[Page 378\]](#).

Completion of BRBACKUP Backups

The completion of database backups is relevant in the following situations:

- A backup has terminated and you do not want to repeat it. Using the BRBACKUP option [-f|-fillup \[Page 373\]](#) you can simply continue the backup. BRBACKUP automatically determines the files that are still to be backed up: Target volume (defined with the parameter backup_mode or the option -m) minus volume backed up (BRBACKUP detailed log). The completion of a backup can also refer to several terminated backups. In this case a new volume of files to be backed up can be specified, differing from the original target volume.
- You are making partial database backups and want to ensure, or monitor, that these will produce a complete backup. This is especially important for large databases, for which it is recommended to spread the backup over several days (e.g. a week), in order to reduce the volume of data to be backed up every day. At the end of the week, to ensure that you have a complete backup, complete the partial backups on the last day of the cycle with the option -f|-fill. As above BRBACKUP automatically determines the files to be backed up.



`brbackup -f b<timestamp>.pft`: The backup with the log name `b<timestamp>.pft` is completed.

`brbackup -f last`: the last executed backup is completed.

`brbackup -f 1`: the current day's backup is completed.

`brbackup -f 7`: the backups of the last seven days (including the current day) are completed (vgl. Abbildung).

See also:

[-f|-fillup \[Page 373\]](#)

Completion of BRBACKUP Backups

[Completion of BRRESTORE Runs \[Page 238\]](#)

[Partial Backups \[Page 184\]](#)

Backing up Database Files

You can back up individual database files, tablespaces, or the entire database. BRBACKUP uses the procedures recommended by ORACLE for carrying out online and offline database backups. See [Backup Mode \[Page 127\]](#).

To specify database files, you can use the file ID, a generic name, or the full path name. By specifying ID intervals or a generic path, you can back up all the database files that meet these specifications.

- When you specify `all`, all the database data files (and therefore all the tablespaces) and control file are saved. In case of an offline backup, a file for every online redo log group is also saved.
- You can combine the value `all` with an `<object list>`. This enables you to back up other non-database files in addition to the database itself. This procedure is not highly recommended, however; whenever possible, save the database files and the non-database files in separate backup runs.
- The control file can only be addressed explicitly using the file ID 0. It is not usually necessary to back up this file, since it is always backed up automatically whenever at least one database data file is backed up.
- Online redo log files can only be addressed explicitly using the redo log group number, which must be assigned a leading zero (`0<n>`). To save all the online redo log files, specify the file ID `00`.
- All file IDs in the interval specified by `<file_ID1>-<file_ID2>` must be known in the database.
- If you use a generic path to define database data files, note that this path must contain the `ORACLE_HOME` directory and an additional generic specification (e.g. `sapdata<n>` directory).

See also [backup_mode \[Page 316\]](#) or [-m|-mode \[Page 378\]](#).

Hardware Compression

Hardware Compression

When backup devices with hardware compression are used, BRBACKUP requires the current compression rates of the database files in order to determine the quantity of data there will be after the hardware compression. Only in this manner can BRBACKUP ensure that the specified tape size is not exceeded and that the database files are correctly divided between the tapes.

BRBACKUP can only estimate the quantity of data that can be written to the volume, because BRBACKUP cannot directly determine the compression rates for hardware compression. BRBACKUP uses the software compression rates as an estimate. SAP therefore assumes that hardware and software compression return similar results. See [compress_cmd \[Page 321\]](#).

Before the first backup using tape devices with hardware compression, you should start a compression run to determine the compression rates: `brbackup -k only`. This call does not actually start a backup as it only determines the compression rates. The database files are merely compressed (not saved) and the determined compression rates are stored in table SDBAD and in a detail log (see [BRBACKUP, BRARCHIVE, and BRRESTORE Logs \[Page 434\]](#)).



Repeat this activity at least once a month to update the compression rates.

Also repeat this activity after a reorganization or after the loading of a large amount of data.

Also see [Tape Devices with Hardware Compression \[Page 115\]](#).

BRARCHIVE

The following information is specific to BRARCHIVE. You can find further information about this utility in [Using BRBACKUP and BRARCHIVE \[Page 219\]](#).

BRARCHIVE is used to archive offline redo log files. Make sure that the initialization profile `init<DBSID>.sap` is configured properly (see [Description of Parameters \[Page 309\]](#)) and/or use the appropriate command options of BRARCHIVE for this purpose (see [Command Options for BRARCHIVE \[Page 390\]](#)).

- You can also start BRARCHIVE when the database is shut down.
- You should archive the offline redo log files on tape using BRARCHIVE.
- In contrast to BRBACKUP, BRARCHIVE does not have its own management of tape continuation. When a tape is full, you must restart BRARCHIVE to write to the next volume.
- For security reasons, we recommend using the option of archiving the offline redo log files to two backup devices in parallel (`brarchive -ss`, `brarchive -ssd`). You can also make this second copy serially (either by restarting BRARCHIVE with `brarchive -sc` or `brarchive -scd` or by using the option `-cs` or `-cds`). See [-s|-sc|-ds|-dc|-sd|-scd|-ss|-ssd|-cs|-cds \[Page 407\]](#).
- Use the option `-f|-fill` to archive the offline redo log files permanently. In this way, you can make sure that the archiving directory does not fill up. See [-f|-fill \[Page 396\]](#).
- To ensure that the offline redo log files are archived smoothly, the summary log must exist in readable form. Please do not delete or change any entries in this log.

[Hardware Compression \[Page 234\]](#)

[Logging \[Page 235\]](#)

Hardware Compression

Hardware Compression

When you archive the offline redo log files with BRARCHIVE, performing software compression in advance is useless, since the previously determined compression rates cannot be used for archiving the next redo log files. If BRARCHIVE does not have any information on the compression rates, the default value of 1 is assumed. This means, for example, that when the tape size is 1600 MB and the redo log files are 20 MB, up to 80 offline redo log files can be archived on a volume.

If, however, a larger number of offline redo log files were written on one day that actually fit on the tape, you can change the parameter `tape_size` accordingly.

Consider the following: We know from experience that during a hardware compression, the size of the offline redo logs shrinks by around a third. Therefore, you can increase the `init<DBSID>.sap` parameter `tape_size` by around 50% (to 2400 MB in the above example), thus reflecting the actual compression rate. This enables you to archive many more (up to 120 in the example) offline redo log files with one BRARCHIVE call.

You can define a separate `init<DBSID>.sap` profile for BRARCHIVE - if you do, changes to individual parameters will not affect the next database backup BRBACKUP.

Also see [Tape Devices with Hardware Compression \[Page 115\]](#).

Logging

BRARCHIVE writes the following file system logs:

- `<ORACLE_HOME>/saparch/a<encoded timestamp>.<ext>`
Every detail log contains information about the actions performed during an archiving run.
- `<ORACLE_HOME>/saparch/arch<DBSID>.log`
This summary log contains a brief entry for every archiving operation that was performed.

BRARCHIVE also records its actions in the database tables SDBAH and SDBAD.

See [BRARCHIVE Logs \[Page 445\]](#).

Using BRRESTORE

Using BRRESTORE

This SAP utility enables you to restore an entire database backup or parts of it, when the backup was performed with BRBACKUP. Any non-database files and directories you saved can also be restored. In the process, the subdirectories in sapdata<n> directories are automatically created, when necessary.

You can also restore the offline redo log files that were archived with BRARCHIVE. This operation can be performed at the same time as the restore of the corresponding backup.

- BRRESTORE can not run unattended. The option `-c` only suppresses the first confirmation prompts for mounting a volume.
- BRRESTORE uses the BRBACKUP logs and the summary log from BRARCHIVE to decide where to restore the requested file. You can manually specify a different directory as well.
- One or more incomplete BRRESTORE runs can be completed with the option `-f`. BRRESTORE automatically determines the files to be restored.

BRRESTORE only restores the selected backup; it does not recover the database. To do this, start the recovery afterwards using the SAP utility SAPDBA (see [SAPDBA Database Recovery \[Page 697\]](#)) or with the ORACLE utility SVRMGR (see [Recovery with SVRMGR \[Page 726\]](#)).

[Restoring Files \[Page 237\]](#)

[Examples \[Page 239\]](#)

[Logging \[Ext.\]](#)

Restoring Files

BRRESTORE can be called directly from the operating system command level. A list of the command options can be found in the sections [Command Options for BRRESTORE \[Page 413\]](#).

BRRESTORE also requires several parameters to be configured in the initialization profile `init<DBSID>.sap`. See [Description of Parameters \[Page 309\]](#).

- The options `-d`, `-k`, `-m`, and `-r` may be preset using the appropriate BRRESTORE profile parameters. See [Effects of the Command Options \[Page 223\]](#).
- Only one of the options `-a`, `-b`, or `-n` may be set. If you do not select any of these options, `-b last` is selected.

BRRESTORE can, however, be started with option `-a` in parallel to BRRESTORE with option `-b` or `-n`.

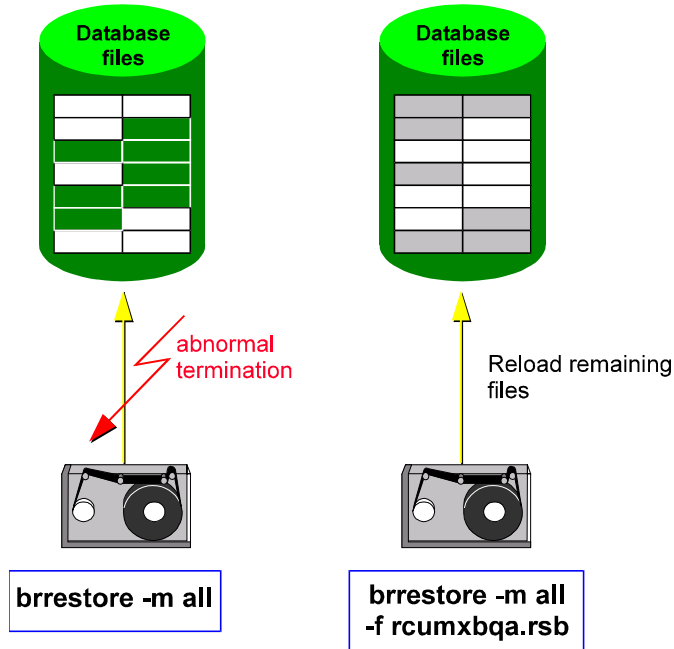
BRRESTORE can restore a database in parallel. BRRESTORE can also restore archived redo log files in parallel if they are located on several volumes. In both cases, several tape devices must be available, and the number of parallel copy processes must correspond to the number of tape devices (this is the standard setting with [exec_parallel \[Page 331\]](#)).

- To restore a complete backup (reset the database), use the option `-m full`.
- The key word `all` can be combined with other options in an `<object list>` (e.g. non-database files or directories). However, database files and non-database files are restored separately from each other, if they were backed up in separate backup runs, as recommended.
- Non-database files and directories must be defined with their complete path. Single files from the backed up directory can also be restored.
- Database data files can be defined with either a full or generic path. A generic path must contain the directory `ORACLE_HOME` and a generic specification (e.g. `sapdata<n>` directory).
- All the file IDs used in a specified interval `<file_ID1>-<file_ID2>` must be known in the database.
- Online redo log files can only be addressed directly by using the redo log group numbers with an additional leading zero (`0<n>`). To address all the online redo log files, specify file ID `00`.
- The control file can only be addressed directly by using file ID `0`.
- When you start the restore of online redo log files or the control file, the mirror copies of these files are automatically recreated.
- Archived redo log files can be addressed using their log sequence numbers.

Completion of BRRESTORE Runs

Completion of BRRESTORE Runs

If a BRRESTORE restore is incomplete, for example, due to a termination of the BRRESTORE program you can complete the remainder of the files in another BRRESTORE run with the option `-f|-fillup`.



If you want to complete a terminated BRRESTORE run, start BRRESTORE with the option `-f|-fillup` [\[Page 422\]](#) and specify the log name of the BRRESTORE run to be completed. If you want to complete several runs, specify all log names individually.

See also:

[Completion of BRBACKUP Backups \[Page 229\]](#)

Examples

```
brrestore -b last -m all
```

Restore all tablespaces without the control file and online redo log files, starting from the last successful backup.

```
brrestore -b bcnmhluz.aft -m full
```

Restore all the files from backup `bcnmhluz.aft`, including the control file and the online redo log files. Restore the mirror copies of the control file and the online redo log files.

```
brrestore -m /usr/sap/C11/SYS/profile
```

Restore the SAP profiles.

```
brrestore -m /usr/sap/C11/SYS/exe/run:sapdba
```

Restore the SAP executables.

```
brrestore -m /oracle/C11/sapdata1=/oracle/C11/sapdata5
```

Restore all the database data files that were originally stored in the subdirectories of `/oracle/C11/sapdata1` in directory `/oracle/C11/sapdata5`.

```
brrestore -b last -m 1-10,01-04,0
```

Restore all the database data files with ORACLE file IDs from 1 through 10, the four online redo log files, and the control file starting from the last successful backup. Restore the mirror copies of the control file and the online redo log files.

```
brrestore -m 0
```

Restore the control file. Restore the mirror copies of the control file.

```
brrestore -b last -m /oracle/C11/sapdata2/ddicd_5/ddicd.data5
```

Restore a database data file starting from the last successful backup.

```
brrestore -a 200-220
```

Restore the archived redo log files with the log sequence numbers from 200 through 220 into the archiving directory.

```
brrestore -a 40-70=/oracle/C11/sapbackup, 71-90=/oracle/C11/sapreorg
```

Restore the archived redo log files with the log sequence numbers from 40 through 70 in directory `sapbackup` and those with the log sequence numbers from 71 through 90 in directory `/oracle/C11/sapreorg`.

```
brrestore -a 40-69,70-100=/oracle/C11/sapbackup
```

Restore the archived redo log files with the log sequence numbers from 40 through 69 in the archiving directory, and those with the log sequence numbers from 70 through 100 in directory `sapbackup`.

```
brrestore -n det_log
```

Restore a detail log to the local directory.

BRCONNECT and BRTOOLS

BRCONNECT and BRTOOLS

Prerequisites

BRTOOLS and BRCONNECT are utility programs that are not started by the user; instead they are started internally by BRBACKUP and BRARCHIVE.

Using BRCONNECT (Control Database Status)

BRBACKUP starts the SAP utility program BRCONNECT during the backup to see if the status of the database corresponds to the backup mode. If the database is not already open, BRBACKUP always starts it. If you select backup mode **online** (`backup_type = online`), the database remains in this state during the backup. If you select backup mode **offline** (`backup_type = offline` or `backup_type = offline_force`), the database is shut down and remains in the closed state during the backup. If an error occurs, BRCONNECT terminates the backup and displays the [messages \[Page 495\]](#) BR312E or BR313E. After the backup, the database is always restored to its original status. This means that the database is left started, if it was running before the backup, or it is shut down, if it was shut down before the backup.

Using BRTOOLS

The SAP utility program BRTOOLS is used for:

- backup verification: the saved files are compared with their originals.
See [-w|-verify \[Page 389\]](#)
- the option `-c force`: This denies the subroutines that are started (such as `cpio`) access to the console.
See [-c|-confirm \[Page 369\]](#)
- Parallel backup to a remote host (`backup_dev_file = pipe`). See [Parallel Backup \[Page 135\]](#).

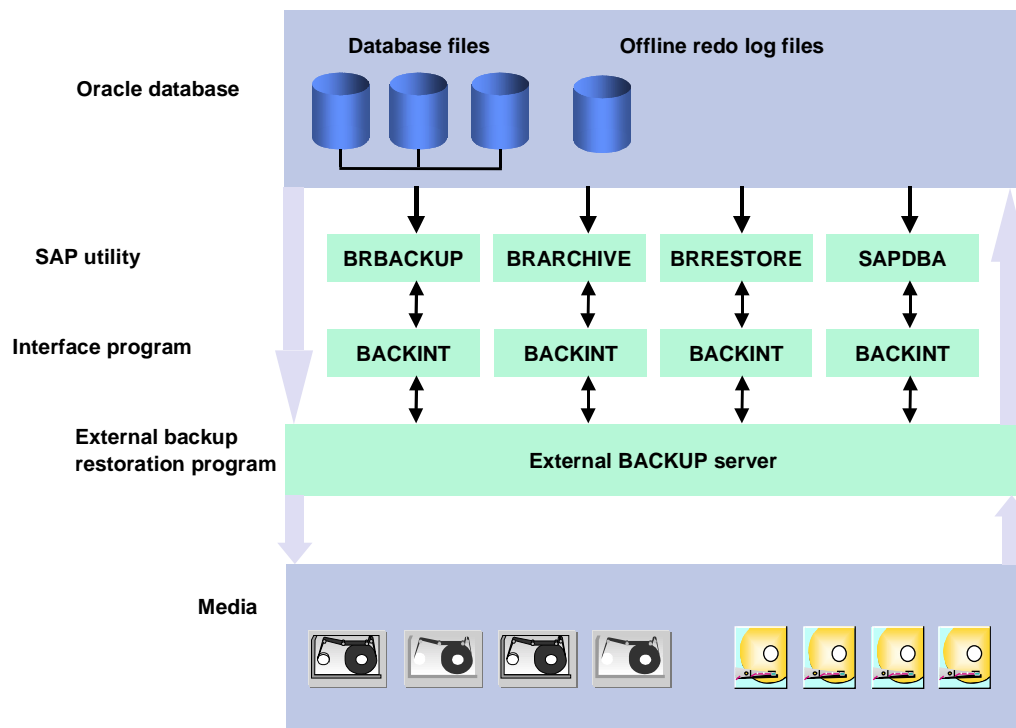
These are just a selection of the many functions offered by BRTOOLS.

Using External Backup Programs

The SAP utilities BRBACKUP, BRARCHIVE, and BRRESTORE provide an interface that can be used to access external backup programs.

This interface can only be used if the BACKINT interface program is provided by the supplier of the external backup program.

If you use this interface, tasks are distributed as follows:



- BRBACKUP/BRARCHIVE take care of the database handling.
- The external backup program is responsible for managing backup media.
- BRBACKUP/BRARCHIVE use BACKINT to pass a backup request on to the external backup program. This request represents a list of the files for backup. BRRESTORE also uses BACKINT to trigger the external program to restore the requested files. Any parameters that are passed on are contained in a parameter file you define. The external backup program performs all the backup operations.
- BRBACKUP/BRARCHIVE/BRRESTORE evaluate the confirmation messages of the external backup program.



Settings:

- Device type:

Using External Backup Programs

```
backup_dev_type = util_file|util_file_online
```

- Parameter file:

```
util_par_file = initC11.utl
```

[Advantages \[Page 243\]](#)

[Disadvantages \[Ext.\]](#)

[Companies that Support the Interface to External Backup Programs \[Page 244\]](#)

Advantages

This interface enables you to utilize new, manufacturer-specific backup media. For example, BRBACKUP/BRARCHIVE/BRRESTORE do not support direct backup to or restore from optical storage media. Such media can now be used via an external backup program using the interface.

By using an external backup program, you can set up a consistent backup procedure for file systems and databases

Many backup programs are not hardware-specific and can be used in the network (for example, backup to an automatic tape system on a mainframe).

Companies that Support the Interface to External Backup Programs

Companies that Support the Interface to External Backup Programs

The following companies support the interface:

SAP Hardware Partners

- HP for Omniback
- IBM for ADSM

Software Suppliers

- Legato for Networker (supported by the hardware partners: DG, DIGITAL, SNI, SUN)
- Hinrichs&Hinrichs for Hiback
- Cheyenne for ArcServe

Other Suppliers of the Interface Program:

- MultiVision
- MultiStream
- OpenVision
- Newlog
- MIT

For more information, please contact the relevant partner (Competence Center) of the software supplier or the vendor of the interface program directly.

The SAP Utilities under Windows NT

This chapter describes several special features of the SAPDBA, BRBACKUP, BRARCHIVE, and BRRESTORE programs under Windows NT.

[SAP Conventions \(Windows NT\) \[Page 246\]](#)

[Backup Strategy \(Windows NT\) \[Page 259\]](#)

SAP Conventions (Windows NT)

SAP Conventions (Windows NT)

Due to the directory structure of Windows NT, the main differences to the documentation for the SAP utilities are in the UNIX-specific file structures. The meanings and contents of file names, environment variables, profiles, etc., are assumed to be known, and will therefore not be discussed below.

[Environment Variables \(Windows NT\) \[Page 247\]](#)

[Directory Structure \(Windows NT\) \[Page 249\]](#)

[Naming Conventions for Files \(Windows NT\) \[Page 252\]](#)

[Executables \[Page 253\]](#)

[Starting the SAP Utility Programs \[Page 254\]](#)

[Reorganization \(Windows NT\) \[Page 256\]](#)

[Database Analysis \[Page 257\]](#)

[Restrictions on SAPDBA \(Windows NT\) \[Page 258\]](#)

Environment Variables (Windows NT)

- **ORACLE_SID**: System ID of the database instance

Example: C11

SAPSID or **sapsid** refers to the name of the R/3 System (SAP System ID).

DBSID or **dbsid** refers to the name of the database instance (database instance system ID).

When a single instance is installed, **SAPSID** and **DBSID** are the same.

- **ORACLE_HOME**: HOME directory of the Oracle software.

Standard: <drive>:\orant

Example: D:\orant

- **SAPDATA_HOME**: Directory of the database files.

Standard: <drive>:\oracle\<DBSID>

Example: E:\oracle\C11



The variables **ORACLE_SID**, **ORACLE_HOME** and **SAPDATA_HOME** must always be set during installation. There is no default.

The following environment variables must only be set if the corresponding paths deviate from the defaults specified here:

- **SAPARCH**: Directory for the BRARCHIVE logs.
Default value: %SAPDATA_HOME%\saparch
- **SAPBACKUP**: Directory for the BRBACKUP logs.
Default value: %SAPDATA_HOME%\sapbackup
- **SAPCHECK**: Directory for the sapdba -check/analyze logs.
Default value: %SAPDATA_HOME%\sapcheck
- **SAPREORG**: Directory for all other SAPDBA logs, as well as shell and SQL scripts. It is also the standard directory for export and unload dump files, if the parameter `exireo_dumpdir` in the profile `init<DBSID>.dba` is not set.
Default value: %SAPDATA_HOME%\sapreorg
- **SAPTRACE**: Directory for Oracle trace files and the alert file.
Default value: %SAPDATA_HOME%\saptrace
- **SAPDATA1**: Directory of the DB data files.

Environment Variables (Windows NT)

Default value: %SAPDATA_HOME%\sapdata1

(Analog for SAPDATA<n>, n=1, ...99).

Note: You can distribute the SAPDATA directories across several different drives, without defining the environment variables SAPDATA<n>. The environment variables only have to be defined individually if directories are used that deviate from the default. The drive is automatically recognized by SAPDBA.

(Example: SAPDATA1= F:\data\prod\sapdata1).

Other environment variables that can be set:

- **LINES:** Definition of the screen height.
Recommended height: ≥ 24 lines.
- **COLUMNS:** Definition of the screen width.
Recommended width: ≥ 80 columns
- **SAPDBA_DEBUG:** Setting the trace function for error analysis.
See Note 29324 for setting options.

Directory Structure (Windows NT)

The directory ORACLE_HOME contains, amongst others:

File name/Directories	File type/Directories
database\init<SID>.ora	ORACLE profile
database\init<SID>.sap	BRBACKUP, BRARCHIVE, BRRESTORE profile
database\init<SID>.dba	SAPDBA profile
bin\svrmgr30	SVRMGR from ORACLE

The directory SAPDATA_HOME contains, amongst others:

\sapreorg	Directory for export and unload dump files, Buffer for compress from BRBACKUP
<timestamp>	Directory for files and scripts
<timestamp>.<ext>	detailed log files

\sapcheck

<timestamp>.chk	detailed log file for results from SAPDBA runs with the option -check
<timestamp>.aly	detailed log file for results from SAPDBA runs with the option -analyze
<timestamp>.nxt	detailed log file for results from SAPDBA runs with the option -next
\sapbackup	Directory for any backups to disk
back<SID>.log	Summary log
b<encoded timestamp>.<ext>	detailed log files
\saparch	Directory for the Oracle offline redo log files.
arch<SID>.log	Summary log
a<encoded timestamp>.<ext>	detailed log files

\saptrace

background	Directory for the Oracle alert files and trace files for the background processes.
usertrace	Directory for Oracle trace files from application programs.
\sapdata1,...\sapdata<n>	Directories for the database data files

Directory Structure (Windows NT)



The directory for backups to disk (SAPDATA_HOME\sapbackup) is defined via the `init<SID>.sap` parameter `backup_root_dir` ;
the directory for export and unload dump files (SAPDATA_HOME\sapreorg) via the `init<SID>.dba` parameter `exireo_dumpdir` and
the directory for the Oracle offline redo log files (SAPDATA_HOME\saparch) via the `init<SID>.ora-` parameter `log_archive_dest`.

See also:

[Directory Tree \(Windows NT\) \[Page 251\]](#)

Directory Tree (Windows NT)

Most important directories for SAP (>=4.0) & ORACLE

```

%ORACLE_HOME%\<drive>\oramt)
├── database)
│   ├── init<SID>.ora (profile for ORACLE)
│   ├── init<SID>.dba (profile for SAPDBA)
│   ├── init<SID>.sap (profile for BRBACKUP, BRARCHIVE and BRRESTORE)
│   ├── sdxexE.dba (English help text)
│   ├── sdxexD.dba (German help text)
│   ├── sdxindexE.dba (created once SAPDBA's help is used)
│   └── sdxindexD.dba (created once SAPDBA's help is used)
├── bin)
│   ├── binaries like oracle80.exe, svrmgrf30.exe, sgplus.exe, exp80.exe, imp80.exe, etc.:
│   ├── nwr-xr-x ora-csid- dba oracle80
│   ├── nwr-xr-x ora-csid- dba svrmgrf30
│   └── ...
├── %SAPDATA_HOME%\<drive>\oracle\<SID>
│   ├── origlog)
│   │   ├── log_g11m1.dbf (g = group, m = member)
│   │   └── log_g13m1.dbf
│   ├── origlogB)
│   │   ├── log_g12m1.dbf
│   │   └── log_g14m1.dbf
│   ├── mirlog) for mirrored redo logs, optional, so specification can vary:
│   │   ├── log_g11m2.dbf
│   │   └── log_g13m2.dbf
│   ├── mirlogB)
│   │   ├── log_g12m2.dbf
│   │   └── log_g14m2.dbf
│   ├── sapdata1)
│   │   ├── cntrf<cntrf<SID>.dbf (database control file)
│   │   ├── system_1) system.data1
│   │   └── btabd_1) btabd.data1 (typical SAP data file for tablespace PSAPBTABD)
│   ├── sapdata2)
│   │   ├── cntrf<cntrf<SID>.dbf (mirrored database control file)
│   │   └── btabi_1) btabi.data1 (typical SAP data file for tablespace PSAPBTAB)
│   ├── sapdata <n>1)
│   ├── sapreorg)
│   │   ├── Logs written by SAPDBA:
│   │   ├── <timesamp>-ch (Log check option)
│   │   ├── <timesamp>-rat (Alter storage in Reorg Menu)
│   │   ├── <timesamp>-rsi (Reorg. table/index)
│   │   ├── <timesamp>-rtc (Reorg. TSP without data file)
│   │   ├── <timesamp>-rid (Reorg. TSP with data file)
│   │   ├── <timesamp>-ext (TSP extension)
│   │   ├── <timesamp>-exs (Export structure)
│   │   ├── <timesamp>-exd (Export table with data)
│   │   ├── <timesamp>-exg (Export constraints, views, seq,...)
│   │   ├── <timesamp>-ims (Import structure)
│   │   ├── <timesamp>-imd (Import table with data)
│   │   ├── <timesamp>-cv (Recovery)
│   │   ├── <timesamp>-brb (Backup)
│   │   ├── <timesamp>-bra (Archive)
│   │   ├── <timesamp>-dba (Startup/Shutdown, etc.)
│   │   ├── <timesamp>-rmv (Move/Rename)
│   │   ├── <timesamp>-rli (Reorg. Tab./Ind.-List)
│   │   ├── <timesamp>-sn (Restore/Recovery : now)
│   │   ├── <timesamp>-sp (Restore/Recovery : point-in-time)
│   │   ├── <timesamp>-rst (Restore/Recovery : Reset DB)
│   │   ├── struct<SID>.log (structure log)
│   │   ├── reorg<SID>.log (main log)
│   │   └── <timesamp>\)
│   │       ├── Scripts, parameter files, restart files written by SAPDBA:
│   │       ├── alk<TSP>.sql (Delete statistics, alter storage)
│   │       ├── con<TSP>.sql (Create constraints)
│   │       ├── ccsqg.sql (Create constraints, views, seq., syn., views, grants)
│   │       ├── dro<TSP>.sql (Drop table or tablespace)
│   │       ├── grants.sql (Create grants)
│   │       ├── ind<TSP>.sql (Index structure file)
│   │       ├── inx<TSP>.sql (Create indexes)
│   │       ├── osw<TSP>.sql (Log switch, backup control file)
│   │       ├── lsp<TSP>.sql (Create tablespace PSAP-TSP)
│   │       ├── tab<TSP>.sql (Create tables)
│   │       ├── del<TSP>.sh (Remove data files)
│   │       ├── exp<TSP>.sh (Export)
│   │       ├── imp<TSP>.sh (Import)
│   │       ├── ex<TSP>.sh (Chop(un)-compress for export/import)
│   │       ├── im<TSP>.sh (Chop(un)-compress for export/import)
│   │       ├── un<TSP>.sh (Unload)
│   │       ├── ld<TSP>.sh (Load)
│   │       ├── exp<TSP>.par (Parameter file for export)
│   │       ├── ex<TSP>.par (Parameter file for chop)
│   │       ├── ld<TSP>.par (Parameter file for (un)load)
│   │       ├── restart.rli (Restart for Reorg. list)
│   │       ├── restart.rmv (Restart for Move/Rename)
│   │       ├── restart.rsi (Restart for Reorg. table/index)
│   │       ├── restart.rtc (Restart for Reorg. TSP)
│   │       ├── restart.rid (Restart for Reorg. TSP with data files)
│   │       ├── restart.exs (Restart for import of structures)
│   │       └── restart.exd (Restart for export/import of tables/indexes)
├── %SAPDATA_HOME%\<drive>\oracle\<SID>
│   ├── sapcheck)
│   │   ├── <timesamp>-chk (log of SAPDBA-check)
│   │   ├── <timesamp>-mt (log of SAPDBA-mvt)
│   │   ├── <timesamp>-aly (log of SAPDBA-analyze)
│   │   └── <timesamp>-opt (log of SAPDBA-checkopt)
│   ├── sapbackup)
│   │   ├── back<SID>.log (BRBACKUP summary log)
│   │   ├── <coded timesamp>-xyz (BRBACKUP detail log)
│   │   │   ├── where x = a (all)
│   │   │   │   ├── p (partial)
│   │   │   │   ├── y = n (online)
│   │   │   │   │   ├── f (offline)
│   │   │   │   │   ├── z = p (pipe)
│   │   │   │   │   ├── t (tape)
│   │   │   │   │   ├── d (disk)
│   │   │   │   │   └── f (util. file)
│   │   │   ├── xyz = qub (query for which tapes will be used)
│   │   │   ├── xyz = cmb (software compression rate determination)
│   │   │   └── xyz = tio (tape initialization)
│   │   ├── <coded timesamp>-xyz (BRRESTORE detail log)
│   │   │   ├── xyz = rab (restore backup files)
│   │   │   ├── xyz = rsa (restore archive files)
│   │   │   └── xyz = rsf (restore individual files)
│   │   ├── <coded timesamp>\) (disk backups)
│   │   │   ├── cntrf.dbf (if online backup was carried out)
│   │   │   ├── cnf<SID>.dbf (if offline backup was carried out)
│   │   │   ├── back<SID>.log (summary log)
│   │   │   ├── init<SID>.ora
│   │   │   ├── init<SID>.sap
│   │   │   └── <coded timesamp>-xyz (detail log)
│   ├── saparch)
│   │   ├── offline redo logs = archive logs :
│   │   ├── ARC<logseq, number>, <thread_number>
│   │   │   (e.g. ARC4711,1)
│   │   ├── arch<SID>.log (BRARCHIVE summary log)
│   │   ├── detail logs:
│   │   │   ├── <coded timesamp>-sve (original saved)
│   │   │   ├── <coded timesamp>-svd (original saved and deleted)
│   │   │   ├── <coded timesamp>-cpy (original copied/saved a second time)
│   │   │   ├── <coded timesamp>-cpd (original copied/saved a second time and deleted)
│   │   │   ├── <coded timesamp>-div (deleted which were saved once)
│   │   │   ├── <coded timesamp>-dcp (deleted which were saved twice)
│   │   │   ├── <coded timesamp>-sav (parallel saved on two stations)
│   │   │   ├── <coded timesamp>-sdd (parallel saved on two stations and deleted)
│   │   │   ├── <coded timesamp>-zps (copy and save)
│   │   │   ├── <coded timesamp>-cde (copy, delete and save)
│   │   │   ├── <coded timesamp>-qua (query which tapes will be used)
│   │   │   ├── <coded timesamp>-cna (software compression rate determination)
│   │   │   ├── <coded timesamp>-lia (tape initialization)
│   │   │   └── <coded timesamp>-lst (stop archiving using 'brarchive -f stop')
│   ├── saptrael)
│   │   ├── background)
│   │   │   ├── alert_<SID>.log (ORACLE ALERT file)
│   │   │   └── 'trc (ORACLE trace files)
│   │   ├── usertrac)
│   │   │   └── 'trc (user trace files)
├── <drive>\usr\teap<SID>\SYS\exe\run
│   ├── sapdba.exe
│   ├── brbackup.exe
│   ├── brarchive.exe
│   ├── brconnect.exe
│   ├── brrestore.exe
│   ├── brtools.exe
│   ├── R3chop.exe
│   └── ...
└── %SAPDATA_HOME%\<drive>\oracle\<SID>
    ├── %SAPDATA_HOME%\<drive>\oracle\<SID>
    ├── %SAPDATA_HOME%\<drive>\oracle\<SID>
    └── %SAPDATA_HOME%\<drive>\oracle\<SID>

```

Syntax explanation:

- %_* = Environment variable
- <SID> = SORACLE_SID
- TSP = Tablespace
- <TSP> = Tablespace Short Name (e.g. btabd or BTABD for PSAPBTABD)

Colors:

- Directories
- Profiles, logs, scripts, etc (SAP)
- Profiles, logs (ORACLE)
- Database files
- Executables

Naming Conventions for Files (Windows NT)

Naming Conventions for Files (Windows NT)

The regular SAP naming conventions apply - that is, the data files are in directory `<drive>\oracle\<SID>\sapdata<n>\<tablespace name>_<file number>`.

Therefore, the first file of tablespace PSAPPOOLD could be called:

```
F:\oracle\C11\sapdata2\poold_1\poold.data1.
```

You can store the files of one tablespace on different disks. SAPDBA requires that only the name of the drive be changed, and the remainder of the path (`\oracle\C11` in our example) remain the same, to avoid confusion with other databases.

SAPDBA also requires that the logical directory `sapdata<n>` is specified. A further subdirectory `<tablespace name>_<file number>` is created automatically.

For reasons of clarity, SAP recommends that you **not** spread a data pool (indicated by the subdirectory `sapdata<n>`) among different disks.

For security reasons, SAPDBA never creates a subdirectory `sapdata<n>` itself; it only uses an existing one created in the File Manager or using the DOS box.



Programs like the File Manager or SVRMGR can keep files open, which can result in significant problems when SAPDBA attempts to access these files. You should therefore shut down these programs before calling SAPDBA.

Executables

The programs that SAPDBA calls must be accessible from everywhere. Therefore, the paths of the following executables must be set in the `PATH` environment variable.

SAP Programs

SAPDBA.EXE, BRBACKUP.EXE, BRARCHIVE.EXE, BRRESTORE.EXE, BRCONNECT.EXE, BRTOOLS.EXE

Interface Program for External Backup Utility

BACKINT.EXE

Operating System Programs

CPIO.EXE (MKS Tools), DD.EXE (MKS-Tools), MT.EXE (MKS Tools), PSTAT.EXE (MS ResourceKit), MKSZIP.EXE (MKS Tools, compression), UNCOMPRESS.EXE (MKS Tools, decompression)

So far, MKSZIP.EXE and UNCOMPRESS.EXE are only used by BRBACKUP, BRARCHIVE and BRRESTORE.

The `compress` command will be required if you want BRBACKUP or BRARCHIVE to determine the compression rate of tape devices with hardware compression.

ORACLE Programs

SVRMGR30, EXP80, IMP80, SQLLDR80

See also:

[Starting the SAP Utility Programs \[Page 254\]](#)

Starting the SAP Utility Programs

Starting the SAP Utility Programs

To avoid authorization problems when starting the SAP utility programs SAPDBA and BRBACKUP, bear in mind the following notes:

1. On the database server create a local group `ORA_OPER` (or `ORA_<SID>_OPER`). Include the SAP users `<SID>ADM` and `SAPSERVICE<SID>` in this group, if this has not been done during the installation.

When you log on to the database using `connect internal` via `SVRMGR`, you are now no longer required to enter a password. The user is authorized to start and stop the database.

You can now call SAPDBA, BRBACKUP and BRARCHIVE as usual:

```
sapdba -u system/<password>
```

2. SAPDBA

If you call SAPDBA at operating system level, the following alternative to the procedure described in 1. is available. Use the password mechanism to start SAPDBA. The command line option is as follows:

```
sapdba -U <internal>/<password>
```

If you call SAPDBA without the option `-U`, the password is requested interactively. If the database logon is as super user `internal`, the database user is `sys`, not `system`.

3. BRBACKUP

If you call BRBACKUP at operating system level, the following alternatives (A;B) to the procedure described in 1. are available:

- Start BRBACKUP not with the standard ORACLE user `system`, but with the user `internal`.

Example: `brbackup -u internal/<password>`

If the database logon is as super user `internal`, the standard user is `sys`, not `system`.

- Activate the full authorization check in ORACLE. Perform the following steps:
 - In the profile `init<SID>.ora`, enter the parameter `remote_login_passwordfile = exclusive`.
 - Create an Oracle password file (if it does not already exist):

```
orapwd80 file=%ORACLE_HOME%\DATABASE\PWD<SID>.ORA  
password=<internal_password> entries=100
```
 - Restart the database.
 - Call `SVRMGR30` as the user `internal`. Give the authorization `SYSOPER` to the user `system`:

```
grant sysoper to system;
```
 - Give a new password to the user `system` (optional):

```
alter user system identified by <password>;
```

Starting the SAP Utility Programs

Now you can call BRBACKUP as usual:
`brbackup -u system/<password>`

Reorganization (Windows NT)

Reorganization (Windows NT)

A reorganization is not required as often as is commonly assumed. The main reason for this is that Windows NT can theoretically manage an unlimited number of data files. The limitations of the database still apply, however (see [Limitations of the Database System \[Page 579\]](#) **Error! No bookmark name given.**).

You can start the reorganization in the background. SAPDBA uses the `at` command and creates a script `reorg<SID>.bat` in the appropriate SAPREORG directory. The script can be started at the required time, and contains all the commands necessary for starting the requested reorganization. SAPDBA also checks whether the scheduler service required for executing the `at` command is active.

See [Procedure for a Reorganization \[Page 592\]](#).

Database Analysis

By using `sapdba -check`, `-analyze`, or `-next` (see [SAPDBA Command Mode \[Page 69\]](#) **Error! No bookmark name given.**), you can display information about all relevant parts and parameters of the database (tablespaces, fragmentation, data files, extents, directories, freespace, redo log files, SGA parameters, init.ora parameters, etc.) or database analyses. The information is stored in a corresponding log in the SAPCHECK directory.

You can use the Scheduler in the MS ResourceKit (MICROSOFT) to schedule the creation of a check/analysis log, or enter the following command in the command line (example for using `sapdba -check`):

```
at \\<host name> <time> check.bat
```



```
at \\ps0001 00.00 check.bat
```

The database must be active at the specified time. If the database is usually shut down, you can also schedule database startups and shutdowns. To do this use `sapdba -startup` or `sapdba -shutdown` (see [SAPDBA Command Mode \[Page 69\]](#)).

The `check.bat` file should have the following contents:

```
set PATH=%PATH%;<x:>\usr\sap\<SID>\SYS\exe\run
set ORACLE_HOME=<path>
set ORACLE_SID=<SID>
set SAPDATA_HOME=<path>
sapdba -check
```

Make sure the path for SAPDBA is set correctly.



For file `check.bat`

```
set PATH=%PATH%;D:\usr\sap\C11\SYS\exe\run
set ORACLE_HOME=D:\orant
set ORACLE_SID=C11
set SAPDATA_HOME=E:\oracle\C11
sapdba -check
```

To check any of the background statements that have been set, enter `at` in the command line.

Information about the tables and tablespaces can also be displayed online by selecting the menu items *Tablespace administration* → *Freespace and fragmentation of all tablespaces* (see [SAPDBA: Tablespace Administration \[Page 681\]](#) **Error! No bookmark name given.**) or *Reorganization* → *Check extents and fragmentation* (see [SAPDBA: Reorganization \[Page 602\]](#) **Error! No bookmark name given.**).

Restrictions on SAPDBA (Windows NT)**Restrictions on SAPDBA (Windows NT)**

The options below, which are described in this manual, are **not yet supported** for SAPDBA under Windows NT:

- Export to tape (also during a reorganization)
- Compression of the export dump file during export/reorganization
- Expert mode
- Parallel export/import

Backup Strategy (Windows NT)

Three backup programs are available:

NTBackup (MICROSOFT), BRBACKUP, and BRARCHIVE (SAP).

This section will briefly discuss the differences, advantages, and disadvantages of the individual programs. The SAP program BRRESTORE is available for restoring files which were backed up with BRBACKUP/BRARCHIVE.

[NTBackup \[Page 260\]](#)

[BRBACKUP/BRARCHIVE \[Page 261\]](#)

[Other Backup Programs \[Page 262\]](#)

NTBackup

NTBackup

Use

You can use NTBackup to perform offline backups of the SAP data files, the system/rollback data files, the control files, and the online and offline redo log files. Online backup is not possible.

Files saved with NTBackup can be restored using a corresponding graphic menu.

Constraints

SAP only recommends this procedure for small databases or test databases for which an offline backup of the complete database can be performed regularly (for example, every night).

Files saved with NTBackup can be restored via a corresponding graphic menu.

It is not possible to perform a recovery with the check/find/restore/recovery mechanism of SAPDBA or with one of the functions in the menu *Restore/Recovery* (see [SAPDBA Database Recovery \[Page 697\]](#) **Error! No bookmark name given.**) if the backup was created using NTBackup.

BRBACKUP/BRARCHIVE

Use

In contrast to NTBackup (file backup) BRBACKUP is a backup program (online and offline) that was specially designed for the Oracle database. SAP recommends BRBACKUP for large databases, because it also enables you to back up individual tablespaces online when the database is running (with [ARCHIVELOG mode \[Page 33\]](#)).

Offline redo log files can be archived to tape using BRARCHIVE. You can restore a complete backup using BRRESTORE. In a [recovery with SAPDBA \[Page 697\]](#), SAPDBA automatically restores the backup.

Prerequisites

BRBACKUP enables parallel backups to tape and disks. Some parameters in the initialization profile `init<SID>.sap` are specific to the operating system. (For example: `rewind_offline = "mt -f $ offline"`)

Read the notes on authorization checks in [Executables \[Page 253\]](#) **Error! No bookmark name given..**

Activities

Make sure that the environment variables are set correctly. Create a file in which the correct environment is defined before you start BRBACKUP/ BRARCHIVE. This file is scheduled in the `at` job.



```
set PATH=%PATH%;d:\usr\sap\C11\SYS\exe\run
set ORACLE_SID=C11
set ORACLE_HOME=d:\orant
set SAPDATA_HOME=d:\oracle\C11
brbackup -c force <other options>
```

[Database Analysis \[Page 257\]](#)

Other Backup Programs

Other Backup Programs

Use

If you already use external backup programs for large systems, you can use an interface (BACKINT) to link them with BRBACKUP, BRARCHIVE, and BRRESTORE.

If you use a specific backup server with an operating system other than Windows NT, you can also use other external backup programs.

Prerequisite

The BACKINT interface must support Windows NT as client.

Activities

Contact the manufacturer of the external backup program to find out about the capabilities of the BACKINT interface.

See also [Using External Backup Programs \[Page 241\]](#)

Initialization Profile init<DBSID>.dba

You can configure the SAPDBA program to suit your own needs. The parameters and how they are used are described in the following. (default directory <ORACLE_HOME>/dbs, operating system-specific). You can edit the file <ORACLE_HOME>/dbs/init<DBSID>.dba with the appropriate system editor. In some situations, SAPDBA starts the programs BRBACKUP, BRARCHIVE or BRRESTORE. You also have to configure these programs. If this initialization profile does not exist, SAPDBA uses its own internal settings.

After the descriptions of the parameters there is a [Sample Initialization Profile \[Page 288\]](#).

`backup_util_name`

backup_util_name

If you use SAPDBA (BRBACKUP, BRARCHIVE, BRRESTORE) with an external backup program, you can specify the product name here.

Default value: None

Possible values: `<product name>` - Specify the name of the backup program used. This product name can be a maximum of two words and 20 characters long. SAPDBA uses this parameter value in several places to refer to the backup program used and record its name in the logs. If parameter `backup_util_name` does not appear in the profile, SAPDBA uses the name `ext. backup utility` for the program when the external backup program is called.

See [Using External Backup Programs \[Page 241\]](#).

blkdev_dir

If this parameter is set, SAPDBA uses this value and the value specified in `rawdev_dir` to display the raw device assignment (for example in the menu options of the tablespace administration).

Default value: `/dev` (also if the parameter is not set)

Possible values: Block device directories (operating system specific). If this parameter is set, the parameter `rawdev_dir` should also be set.



HP-UX: `/dev/dsk`; AIX: `/dev`; SUN: `/dev/dsk`

SAPDBA searches directory `/dev` and its subdirectories for block device drivers and the corresponding raw device drivers. This definition can be made more precise by setting the parameters `blkdev_dir` and `rawdev_dir`.



HP-UX (version earlier than 10.0): The block and raw device drivers are in separate subdirectories. The search for these drivers can be accelerated with two precise entries in `init<DBSID>.dba` and in addition no exceptions such as floppy disks are displayed.

```
rawdev_dir  /dev/rdsk
blkdev_dir  /dev/dsk
```

check_controlfile_mirror

check_controlfile_mirror

ORACLE offers the possibility to mirror the control files. SAPDBA can check the presence and validity of the control files and their ORACLE mirrors. See also [check_initial_settings \[Page 267\]](#).

Default value: `Y`

Possible values: `Y/N` (Y(es) - control file check is executed, N(o) - control file check is not executed).

The setting `Y(es)` means that, on starting up, SAPDBA issues a warning if one of the control files defined in `init<DBSID>.ora` is not present or invalid. Only then does the initial menu appear. If SAPDBA is already running, this check is performed when ORACLE reads the file `init<DBSID>.ora` (e.g. on starting the database).

See also [Mirroring the Control File \[Page 36\]](#).

check_initial_settings

SAPDBA checks several requirements that must be fulfilled in order to safely operate the ORACLE database (for interactive work with SAPDBA).

Default value: `Y`

Possible values: `Y/N` (Y(es) - checks are performed, N(o) - no checks are performed).

Setting `Y(es)` causes SAPDBA to check the following:

1. No combination of Database log mode `archive` and Automatic archival `disabled`.
2. Existence and validity of the control file and its mirror copies.
3. Existence and validity of the four online redo log files and of at least one mirror copy of each online redo log file.

If one or more of the above conditions are not met, SAPDBA displays a warning when starting up. The initial menu then appears. If SAPDBA is already running, some of these checks are performed when ORACLE reads the file `init<DBSID>.ora` (e.g. on starting the database).

Setting `N(o)` deactivates all these checks. It is possible to deactivate only some of the checks and keep others active by using the parameter `check_controlfile_mirror` and/or `check_redolog_mirror` (see [check_controlfile_mirror \[Page 266\]](#) and/or [check_redolog_mirror \[Page 268\]](#)).

`check_redolog_mirror`

check_redolog_mirror

ORACLE offers the possibility to mirror the online redo log files. SAPDBA can check the presence and validity of the online redo log files and their mirrors (at least 2 members per group should be present). See also [check_initial_settings \[Page 267\]](#).

Default value: `Y`

Possible values: `Y/N` (Y(es) - online redo log group check is executed, N(o) - online redo log group check is not executed).

The setting `Y(es)` means that, on starting up, SAPDBA issues a warning if one of the members of the online redo log group is not present or invalid. Only then does the initial menu appear.

See also [Mirroring Online Redo Log Files \[Page 37\]](#).

chop_util_name

This parameter is used for the chop tool, which can split or concatenate large files. Enter `R3chop` or the name of a chop tool with the corresponding attributes. If you do not define a path, the environment variable `$PATH` will be used.

Default value: No chop option available

See also [SAPDBA: Reorganization Menu \[Page 615\]](#).

`exe_dir`

`exe_dir`

This parameter can be used to define the directory for programs BRBACKUP, BRARCHIVE and BRRESTORE. SAPDBA calls the programs from the directory specified here.

Default value: `/usr/sap/<SAPSID>/SYS/exe/run`

exireo_dumpdir

You can define the default value for the working directory and the directory for the export dump files with this parameter. SAPDBA uses this working directory for `<timestamp>` subdirectories and scripts generated for the individual operations. If this default value is not overwritten by the corresponding interactive entries, SAPDBA also stores the export dump files in this directory.

Default value: `<ORACLE_HOME>/sapreorg`

Possible values: A directory with sufficient space for the scripts and, if necessary, for the data to be exported. Note that this information depends on the operating system. See also [Parallel Export/Import \[Page 643\]](#).



The SAPDBA logs are always under `<ORACLE_HOME>/sapreorg`, whatever the working or dump directory.

exireo_exp_imp_buf

exireo_exp_imp_buf

This parameter is used to specify the size of the buffer for export/import and unload/load (used, for example, by the reorganization operation in SAPDBA).

Default value: 3000000

Possible values: Size of the buffer in bytes. Please do not change the default value.

`expir_period_brarchive`

Period for which the deletion of BRARCHIVE logs is blocked.

Default value: 30

Possible values: Number of days for which deletion of the files by SAPDBA is not possible. See [Displaying and Managing Logs and Profiles \[Page 87\]](#).

`exp_imp_degree`

`exp_imp_degree`

You can define the maximum possible degree of parallelity for export/import processes with this parameter.

Default value: 1

Possible values: $n \geq 1$. This value should be set depending on the CPUs available. See also [Parallel Export/Import \[Page 643\]](#).

file_or_raw

This parameter determines the initial behavior of the tablespace management. This setting can be changed at any time in the SAPDBA menu (except for NT : value is always 1).

Default value:	0 for Unix
	2 for Unix with OPS
	1 for NT (cannot be changed)

Possible values: 0, 1, 2 . These values have the following meaning:

0: The user can choose whether a data file is created on a file system or on a raw device.

1: A data file is created on a file system.

2: A data file is created on a raw device.

See also: [Files/Raw Devices of a Tablespace \[Page 685\]](#), [Extending a Tablespace \[Page 686\]](#), [ORACLE Databases on Raw Devices \[Page 766\]](#).

help_<xy> and language

help_<xy> and language

help_dir

Use this parameter to specify the directory to contain the help file (CD mount).

Possible values:

help_dir = <directory>/<language> (for UNIX)

help_dir = <directory>/<language> (for Windows NT)

<directory>: Installation directory of the help files

<language>: English (EN) or German (DE)

help_file

Use this parameter to specify the name of the initial page of the SAPDBA online documentation *R/3 Database Guide: Oracle*. You do not usually need to set this parameter. SAPDBA will then use the above default.

Default:

4251b3144a1211d182b80000e829fbfe/frameset.htm (for UNIX)

4251b3144a1211d182b80000e829fbfe\frameset.htm (for Windows NT)

This file must be located together with the other HTML help files in the directory defined in the parameter help_dir.

help_viewer

Use this parameter to set the browser for displaying the documentation.

Possible values:

netscape (for UNIX)

iexplore or netscape (for Windows NT)

Default:

help_viewer = netscape (for UNIX)

help_viewer = iexplore (for Windows NT)

language

This parameter determines the language of the help texts.

Default: E

Possible values: E (English), D (German)

This setting only affects the language of the SAPDBA online help. Errors, warnings, and menu texts are always displayed in English.

index_degree

This parameter can be used to create indexes in parallel.

Default value: 1

Possible values: $n \geq 1$. This value should be set depending on the CPUs available. The value `index_degree` defines the maximum possible number of parallel processes.

See also [Performing Processes in Parallel \[Page 627\]](#).

`max_file_size`

`max_file_size`

This parameter can be used to tell SAPDBA the maximum file size which the operating system supports. SAPDBA tries to limit the size of each export dump file to this value when it divides them into several dump directories. If the export dump files are larger than the value defined in `max_file_size`, you can use R3chop to split the files or to increase the number of target directories. If R3chop is used, `max_file_size` defines the maximum size of the split files.

Default value: 2G

See also: [chop_util_name \[Page 269\]](#).

message_buf

This parameter determines the size of the message buffer.

Default value: 100000

Possible values: Size of the buffer in bytes.

ops

ops

If SAPDBA is to be used for a database with Parallel Server Option (OPS), the `init<DBSID>.dba` parameter `ops` should be set.

Default value: `N`

Possible values: `Y/N` (Y(es) - SAPDBA displays OPS-specific information, N(o) - SAPDBA standard functionality)

The setting `Y(es)` causes for example `FREELIST GROUPS` to be included in the list of changeable storage parameters if there is a reorganization of the OPS-specific parameters (see [Overview of the Storage Parameters \[Page 669\]](#)). See also [ORACLE Databases and Parallel Server Option \[Page 769\]](#).

orapwd_always_used

This optional parameter controls the initial SAPDBA logon to the ORACLE database.

Default value: **N**

Possible values: **N(o)**, **Y(es)**.

Yes means that SAPDBA initial uses an identifier (user/password) which is stored in readable form in a protected file (see [orapwd_path \[Ext.\]](#)).

See also: [SAPDBA: Expert Mode \[Page 60\]](#)

parallel_instance

parallel_instance

If SAPDBA is used for a database with ORACLE Parallel Server option (OPS), all the instances running in parallel to the local instance must be defined in each of the local `init<DBSID>.dba` profiles.

Default value: none

Possible values: Description of the non-local instances. Enter each parallel instance individually (it is not possible to define a list of instances after parameter `parallel_instance`).

```
parallel_instance = <inst_descr>
```

```
<inst_descr>: <ORACLE_SID>:<ORACLE_HOME>@<connect_string>
```

```
<ORACLE_SID>: ORACLE system ID of the parallel instance
```

```
<ORACLE_HOME>: ORACLE_HOME directory of the parallel instance
```

```
<connect_string>: ORACLE Connect-String for the parallel instance
```



The local instance is the Dedicated Database Instance C11. Profile `initC11.dba` contains the following entries in order to identify the parallel instances C11_002 and C11_003:

```
parallel_instance = C11_002:/oracle/C11@T:hs0002:C11_002
```

```
parallel_instance = C11_003:/oracle/C11@T:hs0003:C11_003
```

Analog entries must exist in `initC11_002.dba` and `initC11_003.dba` in order to define the other non-local instances.

rawdev_dir

If this parameter is set, SAPDBA uses it and the value defined in `blkdev_dir` to display the raw device assignment (for example in the menu options of the tablespace administration).

Default value: `/dev` (also if the parameter is not set)

Possible values: Raw device directories (operating system specific). If this value is set, parameter `blkdev_dir` should also be set.



HP-UX: `/dev/rdsk`; AIX: `/dev`; SUN: `/dev/rdsk`

SAPDBA searches the directory `/dev` and its subdirectories for block device drivers and the corresponding raw device drivers. This specification can be made more precise by setting parameters `blkdev_dir` and `rawdev_dir`.



HP-UX (version earlier than 10.0): The block and raw device drivers are in separate subdirectories. Two precise entries in `init<DBSID>.dba` can accelerate the search for these drivers and in addition no exceptions such as floppy disks are displayed.

```
rawdev_dir  /dev/rdsk
blkdev_dir  /dev/dsk
```

recovery_degree

recovery_degree

This parameter is used to support parallel recovery.

Default value: 1 (even if the parameter is not set)

Possible values: n ($n \geq 1$). The value 1 specifies that parallel processing is not required. n should have a value between m and $2 * m$, depending on the number m of disks affected.

When doing a parallel recovery, the change entries are read from the redo log sequentially but are written to the data files in parallel. You should use this parameter only if more than one hard disk is involved in the recovery procedure.

sqldba_path

This parameter can be used to specify the directory and names for the ORACLE database administration tool.

Default value:	<ORACLE_HOME>/bin/sqldba (ORACLE version <7.3)
	<ORACLE_HOME>/bin/svrmgr1 (ORACLE version ≥7.3)

trace_level

trace_level

Default value: 2

Possible values: 1 (developer trace), 2 (customer trace)

Since the developer trace displays the passwords, the default value should always be 2 and the profile for the authorization concept of the operating system should be protected against manipulation.

tspadd_tspname

This parameter can be used to define the default values in percentage for a possible tablespace extension.

Default value: 30 (for all tablespaces)

Possible values: <tablespace name> <n>

<n> - Number that defines a suggested size (in %) for a new data file when the named tablespace (<tablespace name>) is extended. The tablespace is extended by <n>% of its original size (sum of the size of all the data files of the tablespace) when the default value is accepted.

Example: **PSAPBTABD 100**

When an extension is performed, the system will suggest adding a file to tablespace PSAPBTABD that is equal to the present size of that tablespace - that is, extend the tablespace by 100%.

Enter a value for every tablespace for which you want to set a default value (it is not possible to specify a list of tablespaces after parameter `tspadd_tspname`). For all tablespaces that are not contained in `tspadd_tspname` entries, SAPDBA assumes a default value of 30%.

Sample Initialization Profile

Sample Initialization Profile

An initialization file, `initC11.dba`, could have the following appearance (no OPS installation):

initC11.dba

<code>exireo_dumpdir</code>	<code>\$ORACLE_HOME/sapreorg</code>
<code>exireo_exp_imp_buf</code>	<code>3000000</code>
<code>tspadd_tspname</code>	<code>PSAPBTABD 100</code>
<code>tspadd_tspname</code>	<code>PSAPBTABI 100</code>
<code>tspadd_tspname</code>	<code>PSAPDDICD 30</code>
<code>tspadd_tspname</code>	<code>PSAPDDICI 30</code>
<code>tspadd_tspname</code>	<code>PSAPLOADD 30</code>
<code>tspadd_tspname</code>	<code>PSAPLOADI 30</code>
<code>tspadd_tspname</code>	<code>PSAPSTABD 60</code>
<code>message_buf</code>	<code>100000</code>
<code>language</code>	<code>D</code>
<code>check_initial_settings</code>	<code>Y</code>
<code>check_controlfile_mirror</code>	<code>Y</code>
<code>check_redolog_mirror</code>	<code>Y</code>
<code>backup_util_name</code>	<code>Networker</code>
<code>chop_util_name</code>	<code>R3chop</code>
<code>expir_period_sapdba_normal</code>	<code>10</code>
<code>expir_period_daily_check</code>	<code>5</code>
<code>expir_period_brbackup</code>	<code>30</code>
<code>expir_period_brarchive</code>	<code>30</code>
<code>expir_period_oracle_trace</code>	<code>1</code>
<code>recovery_degree</code>	<code>1</code>
<code>exp_imp_degree</code>	<code>1</code>
<code>max_file_size</code>	<code>2G</code>

Overview of Script Types

SAPDBA writes the SQL commands for reorganizing, exporting, and importing objects in SQL and shell scripts that are stored in the corresponding work directories. This chapter contains an overview of the script types, the export files, and the parameter files.

[SAPDBA Scripts and Files \[Page 291\]](#)

Starting SQL Scripts Manually with SQLPLUS

Starting SQL Scripts Manually with SQLPLUS

You can also use the ORACLE program SQLPLUS to start the SQL scripts. This is not essential, however, since all the scripts can also be started with the SAPDBA function *Additional functions* → *Execute SQL script*. The scripts `cssvg<DBSID>.sql`, `grants.sql` and `con<TSP>.sql` cannot be used with this function because these scripts can contain SAPDBA-specific commands.

Use the following command, if necessary, to start a script with SQLPLUS:

```
sqlplus <uid</passwd>> @<script[.sql]>
```

<uid</passwd>>: user name and password of the ORACLE DBA user. In the SAP standard system, this is `system` with password `manager`.

<script[.sql]>: Name of the SQL script you want to start.

SAPDBA Scripts and Files

SAPDBA automatically starts each required script. For more information on a script, refer to the chapter that describes the corresponding function. If an error occurs, you can use the Restart function to resume the reorganization or the export/import. If this attempt also fails, you can start the required scripts manually. See "Starting the Scripts".

SAPDBA stores all the scripts (and also the named pipes in case of compression) in subdirectories of the working directory. This is generally the directory `<ORACLE_HOME>/sapreorg` (see [exireo_dumpdir \[Page 271\]](#)) with the corresponding `<timestamp>` directories, which specify the date and time of the executed operation.

The export dump files (possibly compressed) are stored in `<timestamp>` subdirectories of the dump directory or on tape (as defined in *Export dump destination*). The default value for the dump directory is also the directory specified in [exireo_dumpdir \[Page 271\]](#).

If several export dump destinations were defined, there will be a corresponding number of export dump files, export/import shell scripts and parameter files. See "Information in the Export Dump Destination - Script Names".

Some script names change when you use the export/import tools SAPDBA unload/SAPDBA load or SAP unload/SQL*Loader for test purposes. See [SAP unload/load, SQL*Loader \[Page 665\]](#).

Overview of scripts and files

Name	Use
<code>alt<TSP>.sql</code>	Deletion of index analysis. Only before exporting tables and their data, reorganizing lists of objects and reorganizing tablespaces with or without data files.
<code>cssvg<DBSID>.sql</code>	Script for creating constraints, grants, synonyms, sequences, and views.
<code>con<TSP>.sql</code>	Script for creating constraints.
<code>drc<TSP>.sql</code>	Script used during reorganization of tablespaces that deletes the constraints.
<code>dro<TSP>.sql</code>	Script that contains the DROP commands required during a reorganization.
<code>grants.sql</code>	Script for defining grants; is only generated during the export of table and index definitions (structure export).
<code>ind<TSP>.sql</code>	Script for creating indexes. When an export with data is performed, this script is not used in certain cases.
<code>inx<TSP>.sql</code>	SQL script created during a test import. (It is not used in reorganizations).
<code>nam<TSP>.sql</code>	Script for temporarily renaming tables during a reorganization.
<code>rnm<TSP>.sql</code>	Script for temporarily renaming a table during the a reorganization, with delayed deletion of the data (deferred mode).

SAPDBA Scripts and Files

<code>ren<TSP>.sql</code>	Script containing the necessary rename commands if the data file assignment of a tablespace is changed (move/rename). It can also be used to reset temporary renaming of tables during reorganizations.
<code>swi<TSP>.sql</code>	Script used during a reorganization with data files that triggers certain security mechanisms (redo log file switch, backup of the control file).
<code>tab<TSP>.sql</code>	Script for creating tables.
<code>tsp<TSP>.sql</code>	Script used to create tablespaces during the reorganization of tablespaces with data files.
<code>exc<TSP>.sh</code>	Script for compressing (compress) or splitting (R3chop) export dump files.
<code>exp<TSP>.sh</code>	ORACLE export of tables and their data.
<code>exc<TSP>.par</code>	Parameter file for R3chop
<code>exp<TSP>.par</code>	Parameter file for export and import.
<code>exp<TSP>.dmp</code>	Export file or named pipe (if you compress data or split it with R/3 chop) that contains tables with data.
<code>exp<TSP>.dmp.Z</code>	Compressed export file.
<code>exp<TSP>.dmp.<m></code> <code>exp<TSP>.dmp.Z.<m></code>	Export dump files can be split into several smaller files <m> with R3chop.
<code>imc<TSP>.sh</code>	Script for decompressing or merging (R3chop) split export dump files for the ORACLE import.
<code>imp<TSP>.sh</code>	ORACLE import of tables and their data (incl. grants)
<code>imx<TSP>.sh</code>	Shellscript for test import (dump file check)
<code>ldr<TSP>.dmp</code>	Unloader dump directory
<code>ldr<TSP>.par</code>	Parameter file for unload and load
<code>ldr<TSP>.sh</code> <code>unl<TSP>.sh</code>	Script for unload or load; cannot be executed as a separate shell script, contains internal information for SAPDBA.
<code>del<TSP>.sh</code>	Script containing operating system commands for deleting old files, links, and directories and commands for creating new directories and links in case of a reorganization with data files or a change to the data file assignment of a tablespace.



The following scripts are not generated when only indexes are involved in an export or reorganization:

```
exc | exp<TSP>.sh, imp | imc<TSP>.sh, tab<TSP>.sql, grants.sql,
exp<TSP>.dmp, exp<TSP>.dmp.Z, exp<TSP>.par,
ldr | unl<TSP>.sh, imx<TSP>.sh, inx<TSP>.sql
```

Information in the Export Dump Destination - Script Names

- If you define a target for the export dump files with *Export dump destination*, the tables are exported into exactly one export dump file per tablespace.

Script names: `exc | exp | imp | imc | imx | un1 | ldr<TSP>.<ext>`

`TSP`: ID of the tablespace

`ext`: `dmp | dmp.z | sh | par | sql`

All export dump files are saved in the specified directory. If you export to tape, only one export dump file is generated for all the tables to be exported.

- If you define several targets for the export dump files with *Export dump destination*, the tables are distributed on exactly one export dump file per defined target according to their size (not according to tablespace).

Script names: `exc | exp | imp | imc | imx | un1 | ldr<TSP>_<n>.<ext>`

`TSP`: ID of the tablespace of the first table to be edited

`n`: Number of targets defined in *Export dump destination*

`ext`: `dmp | dmp.z | sh | par | sql`

Depending on the degree of parallelism (see [Parallel Export/Import \[Page 643\]](#)), both cases can be edited either sequentially or in parallel.

Starting the Scripts

When a reorganization, export or import process is terminated, you can use the restart functionality of the SAPDBA to repeat the process starting at the corresponding location. You only have to resort to alternative possibilities for starting the individual scripts if such an attempt fails.

If you start the scripts using one of the options specified below, you must adhere to the order in which you call the scripts! Refer to the relevant information in the log written during the corresponding process and in the restart file.



The options for starting the individual scripts described in this section should only be used in case of an emergency.

[Starting Individual SQL Scripts with SAPDBA \[Page 294\]](#)

[Starting SQL Scripts Manually with SQLPLUS \[Page 290\]](#)

[Starting Shell Scripts for ORACLE Export/Import \[Page 295\]](#)

Starting Individual SQL Scripts with SAPDBA**Starting Individual SQL Scripts with SAPDBA**

In the SAPDBA initial menu, select the function *Additional functions* → *Execute SQL script*. A menu appears in which you can enter the required SQL script name (and complete path).

Then start the script.

Starting Shell Scripts for ORACLE Export/Import

SAPDBA uses the ORACLE export/import utilities to export/import tables and their data.

SAPDBA creates shell scripts that can be used to call the export/import programs.

To manually perform an import of tables and their data, for example, enter the following command in the work directory at the prompt:

```
OS> imp<TSP>.sh or imp<TSP>_<n>.sh
```



The files for the unload/load option (`ldr<TSP>.sh,...`) are not genuine shellscripts but merely contain internal SAPDBA information. They cannot be started as separate shellscripts. Unload and load can only be performed with SAPDBA.

SAPDBA Logs

SAPDBA Logs

This chapter provides an overview of the logs that are written by SAPDBA.

These files are contained in the directories

`<ORACLE_HOME>/sapreorg` and `<ORACLE_HOME>/sapcheck`. You can display the logs with the menu option *Show/Cleanup* → *Show log files/profiles*.

[Overview of the Logs \[Page 297\]](#)

[Logging of Parallel Export/Import Processes \[Page 304\]](#)

[Structure Log \[Page 299\]](#)

[Main Log \[Page 300\]](#)

Log Files and Their Contents

Log file	Contents
<timestamp>.dba	Database startup and shutdown messages, deletion of logs, trace, and audit files, log of functions: restoring one tablespace (<i>Restore/Recovery</i> → <i>Restore one tablespace</i>) or restoring individual files (<i>Restore/Recovery</i> → <i>Restore individual file(s)</i>)
<timestamp>.rch	Log of the check function (in submenus of <i>Reorganization</i> → <i>Check extents and fragmentation</i>)
<timestamp>.ral	Log of function: configuration of storage parameters (<i>Reorganization</i> → <i>Alter/Show table or index storage parameters</i>)
<timestamp>.rsi	Log of function: reorganize a single table or index (<i>Reorganization</i> → <i>Reorganize single table or index</i>)
<timestamp>.rli	Log of function: Reorganize a list of objects (<i>Reorganization</i> → <i>Reorganize list of tables or indexes</i>)
<timestamp>.rtc	Log of function: Reorganization of a tablespace (<i>Reorganization</i> → <i>Reorganize tablespace</i>)
<timestamp>.rtd	Log of function: reorganize a tablespace with data files (<i>Reorganization</i> → <i>Reorganize tablespace and datafiles</i>)
<timestamp>.rmv	Log of function: change the data file assignment of a tablespace (<i>Reorganization</i> → <i>Move/rename data files of a tablespace</i>)
<timestamp>.rrs	Log of function: Resize data file (<i>Reorganization</i> → <i>Resize data files of a tablespace</i>)
<timestamp>.ext	Log of function: tablespace administration, including the data file statistics before and after the extension, and the messages displayed by the command ALTER TABLESPACE or CREATE TABLESPACE (<i>Tablespace administration</i>)
<timestamp>.exd	Log of function: export with data (<i>Export/import</i> → <i>Export tables and indexes including data</i>)
<timestamp>.exs	Log of function: export tables and index structures (<i>Export/import</i> → <i>Export structures of tables and indexes</i>)
<timestamp>.exg	Log of function: export other objects (<i>Export/import</i> → <i>Export constraints, synonyms, sequences, views, grants</i>)
<timestamp>.imd	Log of function: import with data (<i>Export/import</i> → <i>Import tables and indexes including data</i>)
<timestamp>.ims	Log of function: structure import (<i>Export/import</i> → <i>Import structures</i>)

Log Files and Their Contents

<timestamp>.chk	Statistics generated by using the SAPDBA command option <code>-check</code> (e.g. display using <i>SAPDBA daily check log files</i>)
<timestamp>.nxt	Log generated by using the SAPDBA command option <code>-next</code> (e.g. display using <i>SAPDBA daily check log files</i>)
<timestamp>.aly	Log generated by using the SAPDBA command option <code>-analyze</code> (e.g. display using <i>SAPDBA daily check log files</i>)
<timestamp>.opt	Log generated by using the SAPDBA command option <code>-checkopt</code> (requirements planning for the DB optimizer).
<timestamp>.brb	Log of the BRBACKUP call by SAPDBA (<i>Backup database</i>)
<timestamp>.bra	Log of the BRARCHIVE call by SAPDBA (<i>Backup archive log</i>)
<timestamp>.rcv	Log of function: recovery (<i>Check (and repair) Database</i>)
<timestamp>.rsn	Log of function: recovery to current time (now) (<i>Restore/Recovery → Full restore and recovery</i>)
<timestamp>.rsp	Log of function: point in time recovery (<i>Restore/Recovery → Full restore and recovery</i>)
<timestamp>.rst	Log of function: restore the database (<i>Restore/Recovery → Full restore</i>)
esc<TSP>.log esc<TSP>_<n>.log	R3chop logs (n=1,...,number of processes). Unlike the export, import, unload, load logs, these are not appended to the main log <timestamp>.<ext> but are generated separately in the subdirectory <timestamp> of the working directory.
ldr exp imp<TSP> _<n>.log	Depending on the number of parallel processes, from the second ORACLE Export/Import process onwards (n=2,...,number of processes) partial, temporary logs are written (see Logging of Parallel Export/Import Processes [Page 304] and SAP Unload/Load, SQL*Loader [Page 665]).
ldr<TSP> <n>.log	Temporary logs if the Export/Import tools SAP unload/SAP load or SAP unload/SQL*Loader are used for test purposes. (See SAP Unload/Load, SQL*Loader).
reorg<DBSID>.log	Main log with overview information on the operations performed (see Main Log [Page 300])
struct<DBSID>.log	Structure log containing information on the structure of the database (see Structure Log [Page 299]).

Structure Log

SAPDBA continuously writes data to the log `struct<DBSID>.log`, including details of the files belonging to the database and whether they were deleted or created. You can find the following information there:

	Explanation
S/D/C	S: File recognized by SAPDBA as belonging to the database. D: File was deleted C: File was created
Time stamp	When the log entry was made.
Tablespace	Name of the tablespace to which the file belongs
ID Filename	ORACLE file ID Full path of this file
Size Size	File size (ORACLE) File size (operating system)
Path Path	If a link exists for this file, this is where it is logged. First is the ORACLE name or partial name of the file, and then the point to which for this file has been set. This could be a link to a file or a link to a raw device. SAPDBA uses these entries to monitor the link structure of the database and to correct it where necessary (Database Check).

This information should help you keep as accurate an overview of the structure of the database as possible. For example, you can compare the ORACLE and operating system file sizes, you can discover the allocation of tablespaces to files and you can find out whether the “files” are really files or raw devices. If the log does not exist, then it will be created the next time that SAPDBA is started. This means that you can generate an up-to-date structure log at any time, for example if the old one has become too confusing.

Central Log reorg<DBSID>.log

Central Log reorg<DBSID>.log

Use

A central log, `reorg<DBSID>.log`, contains information on important operations such as creating, extending and reorganizing a tablespace, restarting an operation, and resetting, restoring or recovering a database. Each time one of these operations is performed, it is entered in the log. The time information (BEGIN or END) helps you to estimate the amount of time needed for operations on tables or tablespaces. This can help you to plan your reorganizations more efficiently, for example.

SAPDBA uses the central log to display the tablespaces where structure changes (reorganization of a tablespace with data files, extension or creation of a tablespace) have been made. This information is particularly important when you are recovering the database, since the recovery depends on any changes that have been made to the structure.

There is no entry in the central log `reorg<DBSID>.log` for the results of the analysis functions *Estimate/compute table/tablespace* and *Validate index* (see [Extent Check Functions \[Page 606\]](#)). However, detail logs are also written (`<timestamp>.rch`) and you can display the latest analysis by using the check functions of the reorganization menu (for example, *Reorganization* → *Check extents and fragmentation* → *Estimate/Compute table* → *Get last analysis*).

Features

`reorg<DBSID>.log` contains the following information:

- **BEGIN**
 `<timestamp>` of the form YYMMDDHHMM
 Time process started
- **END**
 `<timestamp>` of the form YYMMDDHHMM
 Time process ended
- **FID**
 Function ID; also in the extension of the corresponding process log

ext	Extension/creation of a tablespace
rli	Reorganization of a list of objects
rmv	Renaming and moving of data files of a tablespace
rrs	Resizing a data file
rsi	Reorganization of a single table or index
rsn	Restoration of the database and subsequent recovery to the current time (Recovery until now)
rsp	Restoration of the database and subsequent point in time recovery
rst	Resetting the database
rtc	Reorganization of a tablespace without data files
rtd	Reorganization of a tablespace with data files

- **LOG_TIME**
 <timestamp> of the form YYMMDDHHMM
 Logging time
 - **M**
 Mode of action
 Possible values: **N**(ormal), **B**(atch), **R**(estart)
 The operation was performed in normal mode (N) or in the background (B) or started or restarted at a later time (R).
 - **OBJECT**
 <table name>/<tablespace name>
 Name of the table or tablespace on which the operation was performed.
- | | |
|-------|--|
| RESET | if the database was reset |
| RES | if a point in time recovery was performed |
| REC | if a <i>Recovery until now</i> was performed |
- **RC**
[Return Codes in the Central Log reorg<DBSID>.log \[Page 302\]](#)
 - **SIZE_B**
 <n> (in KB)
 Size of the object before the start of the operation; it can be 0 if the tablespace is new, for restart functions, and when resetting, restoring or recovering the database.
 - **SIZE_A**
 <n> (in KB)
 Size of the object after reorganization or extension, or the size of the new tablespace; it can be 0 if the tablespace is new, for restart functions, and when resetting, restoring or recovering the database.
 - **SIZE_T**
 <n> (in KB)
 Size of the export dump file; it can be 0 if the tablespace is new, for restart functions, and when resetting, restoring or recovering the database, for index reorganization and tablespace extensions.

Integration

For more information, see [Connection to CCMS \[Page 305\]](#).

RC

RC

Return code

Reorganization or Restart

Possible values: <15 | 14 | 12 | 11 | ... 2 | 1 | 0>

The return code indicates the last script that was processed successfully during the operation.

Return code	Termination occurred in script
15	nam<TSP>.sql
14	alt<TSP>.sql
13	drc<TSP>.sql
12	exp<TSP>[_<n>].sh (also exc<TSP>[_<n>].sh) or unl<TSP>[_<n>].sh
11	imx<TSP>.sh
10	swi<TSP>.sql
9	dro<TSP>.sql
8	del<TSP>.sh
7	tsp<TSP>.sql
6	tab<TSP>.sql
5	imp<TSP>[_<n>].sql (also imc<TSP>[_<n>].sh) or ldr<TSP>[_<n>].sh
4	ind<TSP>.sql
3	con<TSP>.sql
2	grants.sql or cssv<DIS>.sql
1	ren<TSP>.sql
0	No termination; all scripts were processed successfully.

Extension/Creation of a Tablespace

Possible values: <1 | 0>

Return code	Meaning
1	Extension/creation terminated
0	Extension/creation successful

Resetting the Database

Possible values: <40 | 35 | 30 | 25 | 20 | 15 | 5 | 0>

RC	Explanation (when error occurred)	Database status
40	Start of operation. Search for offline backups.	unchanged
35	BRBACKUP run selected. Reading the header information of the logs.	unchanged
30	Control file and online redo log files saved.	unchanged
25	Setting and listing the restore parameters, deleting the data files, online redo log files and control files.	changed
20	Database was closed. The sapdata<n> directories were cleaned up.	changed
15	Restore of the whole backup (and re-creating the mirrors) was done.	changed
5	Database was opened.	consistent
0	All operations successful.	consistent

Restore and Recovery of the Database (Point in Time or Restore until Now)

Possible values: <40 | 35 | 30 | 25 | 20 | 15 | 10 | 0>

RC	Explanation (when error occurred)	Database status
40	Operation started. Control file and online redo log files were backed up. All tablespaces restored.	changed
35	Old data files deleted.	changed
30	Database was closed. If a control file had to be restored, this was done here.	changed
25	Restore parameters set and listed. Data files restored.	changed
20	Database mounted. Current control file restored (if necessary). New data files created.	changed
15	Archived redo logs were checked and found. Restore parameters were checked and set. The first archived redo log files were restored.	changed
10	Point-In-Time-Recovery. Backup recommended.	consistent
0	All operations successful. The database was opened (RESETLOGS or NORESETLOGS).	consistent

Logging of Parallel Export/Import Processes

Logging of Parallel Export/Import Processes

Export/import processes can run in parallel when the following functions are used:

SAPDBA Function	Main Log
<i>Reorganization</i> → <i>Reorganize list of tables or indexes</i>	<timestamp>.rli
<i>Reorganization</i> → <i>Reorganize tablespace</i>	<timestamp>.rtc
<i>Reorganization</i> → <i>Reorganize tablespace and datafiles</i>	<timestamp>.rtd
<i>Export/import</i> → <i>Export tables and indexes including data</i>	<timestamp>.exd
<i>Export/import</i> → <i>Import tables and indexes including data</i>	<timestamp>.imd

When several parallel export/import processes are started (see [Parallel Export/Import \[Page 643\]](#)) partial, temporary logs are written in addition to the main logs listed above.

They are stored in the directory:

<working directory>/<timestamp> and have the names:

unl | ldr | exp | imp<TSP>_<n>.log (n=2,...,number of processes)

exc<TSP>_<n>.log (n=1,...,number of processes)

The actions of the first process are recorded in the main log. From the second parallel import/export onwards, partial temporary logs are written. After the completion of the last export, the partial logs are appended to the main log and are then deleted. The temporary logs are not displayed on the screen during interactive processing.

If SAPDBA cancels an import or export in progress, the partial logs are kept and not appended to the main log.

Connection to CCMS

SAPDBA records changes to the database due to reorganization, extension, or creation of a tablespace, resetting or restoring and recovering a database not only in the main log `reorg<DBSID>.log`, but also in database tables SDBAH and SDBAD. These details can therefore also be shown in CCMS. Please use the online help of the *Computing Center Management System* for detailed information on using CCMS.

SDBAH and SDBAD are also used by the SAP utilities BRBACKUP and BRARCHIVE (see [BRBACKUP, BRARCHIVE, and BRRESTORE Logs \[Page 434\]](#)).

Furthermore, the results of the analyses (Compute/Estimate statistics, validate structure) are stored in the relevant database tables, so that these statistics can also be viewed from CCMS.

These database tables should only be supplemented or changed by the SAP utilities SAPDBA, BRBACKUP, and BRARCHIVE, or by the CCMS of the R/3 System.

The information from the main log, for example, are stored in the following fields of SDBAH and SDBAD (the following is background information of interest to the DBA and given here for the sake of completeness. CCMS presents this data in a more user-friendly way, so that exact knowledge of the structure of these tables is not required):

Database Tables SDBAH and SDBAD

Table	Fields	Contents of the fields when used by SAPDBA
SDBAH	BEG	BEGIN
	FUN	FID
	OBJ	OBJECT
	RC	RC
	ENDE	END
	ACTID	LOG_TIME
SDBAD	BEG	BEGIN
	FUN	FID
	POS	D(ata)/I(ndex)
	LINE	MODE SIZE_B SIZE_A SIZE_T Number_of_rows Error

See also [Computing Center Management System \[Page 24\]](#).

Initialization Profile `init<DBSID>.sap`

Initialization Profile `init<DBSID>.sap`

The SAP utilities BRBACKUP, BRARCHIVE, and BRRESTORE must be configured before they can be used. To do this, you must set the appropriate parameters in initialization profile `init<DBSID>.sap`. This chapter provides an overview of these parameters.

[Using the Profile `init<DBSID>.sap` \[Page 307\]](#)

[Parameter Format \[Page 308\]](#)

[Description of Parameters \[Page 309\]](#)

Using the Profile `init<DBSID>.sap`

BRBACKUP, BRARCHIVE, and BRRESTORE use a common initialization profile `init<DBSID>.sap` (usually in directory `<ORACLE_HOME>/dbs`) to find the required parameters. This file has TEXT format and can be edited with any system editor.

Before you use one of the SAP utilities, find out exactly which parameters you have to configure. Pay particular attention to parameters without default values and parameters that have device-specific information or require special platform-specific commands.

If you do not make any changes, the present default values for the parameters will be accepted. Many of these parameters can also be overridden by using a command option ([Effects of the Command Options \[Page 223\]](#)). See [Effects of the Command Options \[Page 223\]](#) and [Command Options for BRBACKUP, BRARCHIVE, and BRRESTORE \[Page 363\]](#).

Changes to parameter values do not take effect until you call the corresponding utility.

[Parameter Format \[Page 308\]](#)

Parameter Format

Parameter Format

Use the following format for entering parameters and values in profile `init<DBSID>.sap`:

```
<Parameter> = <value> | (<value1>, <value2>, ...)
```

The individual values in a value list are separated by commas, and the entire list is enclosed in parentheses. Blanks may be used between any symbols of such commands. If necessary, the parameter values can be continued on the next line (the line break is then treated as a blank).

When in doubt, compare your input format with the format in the sample profile, `/usr/sap/<SAPSID>/SYS/exe/run/initSID.sap`, or the initialization profile, `<ORACLE_HOME>/dbs/init<DBSID>.ora`.

Description of Parameters

The parameters and their uses are described below.

[archive_copy_dir \[Page 311\]](#)

[archive_function \[Page 312\]](#)

[backup_dev_type \[Page 314\]](#)

[backup_mode \[Page 316\]](#)

[backup_root_dir \[Page 318\]](#)

[backup_type \[Page 319\]](#)

[compress \[Page 320\]](#)

[compress_cmd \[Page 321\]](#)

[compress_dir \[Page 322\]](#)

[copy_in_cmd \[Page 323\]](#)

[copy_out_cmd \[Page 324\]](#)

[cpio_disk_flags \[Page 325\]](#)

[cpio_flags \[Page 326\]](#)

[cpio_in_flags \[Page 327\]](#)

[dd_flags \[Page 328\]](#)

[dd_in_flags \[Page 329\]](#)

[dismount_cmd \[Page 330\]](#)

[exec_parallel \[Page 331\]](#)

[expir_period \[Page 332\]](#)

[mount_cmd \[Page 333\]](#)

[mount_par_file \[Page 334\]](#)

[new_db_home \[Page 335\]](#)

[parallel_instances \[Page 336\]](#)

[primary_db \[Page 337\]](#)

[read_fifo_cmd \[Ext.\]](#)

[resync_cmd \[Page 340\]](#)

[rewind \[Page 341\]](#)

[rewind_offline \[Page 342\]](#)

[split_cmd \[Page 344\]](#)

[tape_address \[Page 347\]](#)

[tape_address_arch \[Page 348\]](#)

[tape_address_ctl \[Page 349\]](#)

Description of Parameters

[tape_address_ctl_arch \[Page 350\]](#)

[tape_address_rew \[Page 351\]](#)

[tape_address_rew_arch \[Page 352\]](#)

[tape_copy_cmd \[Page 353\]](#)

[tape_pos_cmd \[Page 355\]](#)

[tape_size \[Page 356\]](#)

[tape_size_arch \[Page 357\]](#)

[tape_use_count \[Page 358\]](#)

[uncompress_cmd \[Page 359\]](#)

[util_par_file \[Page 360\]](#)

[volume_archive \[Page 361\]](#)

[volume_backup \[Page 362\]](#)

archive_copy_dir

This parameter identifies the directory used by BRARCHIVE to back up the offline redo log files to a local disk.

Default: First value of the parameter [backup_root_dir \[Page 318\]](#)

Required value: A directory in which the offline redo log files should be archived.

This is generally only required in the following cases:

- You are working with the parallel server option. For the OPS configuration, it might be necessary for BRARCHIVE to provide the offline redo log files on disk. See [Special Features of BRARCHIVE \[Page 773\]](#).
- You are working with a two-phase archiving procedure (similar to a [Two-Phase Backup \[Page 124\]](#).) The offline redo log files are first collected in a directory on a disk and then written to tape using external means. The user is responsible for archiving to tape and for restoring from the tape to the disk (should this become necessary in connection with a recovery).



BRARCHIVE archiving on disk should only be used in the situations mentioned above. In all other cases: The offline redo log files must always be archived on tape. Archiving on disk is not a substitute for archiving the offline redo log files on tape.



Archiving to disk and tape should not be combined during a BRARCHIVE run.



When you archive the offline redo log files to disk, you may only use the BRARCHIVE options `-s`, `-sd` and `-ds`.

archive_function

archive_function

This parameter defines the type of archiving with BRARCHIVE.

Default value: **save**

Possible values:

- **save**: Archive the offline redo log files.
- **second_copy**: Create a second copy of offline redo log files which were already archived.
- **delete_saved**: Delete offline redo log files which were archived once.
- **delete_copied**: Delete offline redo log files which were copied a second time.
- **save_delete**: Archive the offline redo log files and then delete these files.
- **second_copy_delete**: Create a second copy of offline redo log files which were already archived and then delete these files.
- **double_save**: Archive the offline redo log files on two backup devices (tape devices) in parallel.
- **double_save_delete**: Archive the offline redo log files on two backup devices (tape devices) in parallel and then delete the files.
- **copy_save**: Create a second copy of offline redo log files which were already archived and then archive the offline redo log files which have been created in the meantime.
- **copy_delete_save**: Create a second copy of offline redo log files which were already archived. These are then deleted and archiving of the offline redo log files which have been created in the meantime is started.

If there is only one tape device, you can use parameter `archive_function = copy_save` or `copy_delete_save` to ensure that BRARCHIVE creates a second copy of the offline redo log files in one run, deletes it if necessary and continues archiving immediately. This can also be done by first calling BRARCHIVE with `archive_function = save` and then with `archive_function = second_copy` or `second_copy_delete`. However, two BRARCHIVE calls with modified profiles are necessary in this case. In practice, the second possibility is not of great interest and should only be used with the corresponding BRARCHIVE command option call if it must be used at all.

archive_stage_dir

This parameter identifies the directory used by BRARCHIVE to back up the offline redo log files to a remote disk.

This parameter corresponds to [archive_copy_dir \[Page 311\]](#) for a backup to a local disk.

Default: First value of the parameter [stage_root_dir \[Page 346\]](#)

Required value: A directory in which the offline redo log files should be backed up.



```
archive_stage_dir = $SAPDATA_HOME/sapbackup
```

See: [Backup to a Remote Disk \[Page 120\]](#)

backup_dev_type**backup_dev_type**

Determines the backup medium that you want to use.

Default: tape

Possible values:

- **disk**: Database backups/archiving to disk.
The directory(-ies) that should be used for the backups/archiving is(are) defined in parameter **backup_root_dir** (possibly in **archive_copy_dir**).
- **tape**: Use one or more local tape devices.
- **pipe**: Backup to a remote system using the commands entered in the profile parameters [remote_host \[Page 338\]](#), [remote_user \[Page 339\]](#), [copy_in_cmd \[Page 323\]](#), and [copy_out_cmd \[Page 324\]](#). All the functions that are available for a local backup on tape are also available here.
- **tape_auto**: Use of a local tape device with a tape changer. The prompts for changing the volumes are suppressed.
- **pipe_auto**: Use of a tape device with a tape changer on a remote system. The prompts for changing the volumes are suppressed. When you use this parameter specification, do not forget to set the parameters **copy_in_cmd**, **copy_out_cmd**, **remote_host** and **remote_user** appropriately.
The parameters **tape_auto** and **pipe_auto** generally have no effect on BRARCHIVE, since that program does not support continuation tapes.
- **tape_box**: Use of jukeboxes and autoloader. The accompanying tape devices must be locally accessible.
- **pipe_box**: Use of jukeboxes and autoloader. The accompanying tape devices must be remotely accessible (**remote_host**, **remote_user**)
In both cases the drivers defined in the parameter **tape_address** or **tape_address_arch** are used for the data transfer (**cpio**, **dd**). For the rewind those defined in the parameter **tape_address_rew** or **tape_address_rew_** are used.
- **disk_copy**: Copying of database files to a disk with an identical directory structure. The name of the new Oracle_Home directory is defined in the parameter **new_db_home**. See [Structure Retaining Database Copy \[Page 130\]](#).
- **disk_standby**: Copying of database files to a disk with an identical directory structure (compare **disk_copy**). To let you construct a standby database, a standby control file is generated and copied. See [Standby Database Configuration \[Page 186\]](#).
- **util_file**: Backup or restore is carried out file-by-file using the backup program specified by the BACKINT interface program
- **util_file_online**: (if supported by the manufacturer of the external backup program) backup or restoration is made file-by-file using an external backup program addressed by the BACKINT interface. The backup status is also set and ended dynamically for the tablespaces to be saved in an online backup.

backup_dev_type

The parameter `util_file` allows you to use other storage media (e.g. optical media), provided the supplier provides corresponding backup programs and a BACKINT interface. In such cases, the SAP utilities can call the external backup program for the physical backup or restore of the corresponding files. If parameter `util_file_online` is used, the volume of offline redo log files is also drastically reduced during an online backup. See [Using External Backup Programs \[Page 241\]](#).

Ask the supplier of the non-SAP backup programs and the interface BACKINT for any additionally required parameters for the backup program call. If necessary, store this information in an appropriately maintained parameter file, which you should also enter in the parameter `util_par_file` or the command option `-r | -parfile` of the SAP utilities.


- **stage:** Backup to a remote disk. This can be used for standard backups with BRBACKUP, for incremental backups while using the RMAN functions. See [Backup to a Remote Disk \[Page 120\]](#)
- **stage_copy:** Copying of database files to a remote disk with an identical directory structure. The name of the new Oracle_Home directory is defined in the parameter `new_db_home`.
- **stage_standby:** Copying of database files to a remote disk with an identical directory structure (compare `disk_copy`). To let you construct a standby database, a standby control file is generated and copied.
- **rman_prep:** The best distribution of the files to savesets is determined before a RMAN backup to tape with file multiplexing.
See: [Saveset Grouping and Tape Layout \[Page 167\]](#)
- **rman_util:** RMAN backup in combination with a backup library and the backup tool of another manufacturer. BACKINT integrates the external backup tool.
See: [Backups with External Backup Libraries \[Page 162\]](#)

backup_mode**backup_mode**

This parameter is used by BRBACKUP and BRRESTORE to determine the scope of the backup/restore activity.

Default: **a11**

Possible values:

- **a11**: Back up the entire database using BRBACKUP or restore all tablespaces (without control files or redo log files) using BRRESTORE.
- **full1**: Restore a complete backup using BRRESTORE. See [Incremental Backup \[Page 108\]](#)
- **<tablespace name>**: The files of the specified tablespaces will be processed by the appropriate SAP utility.
- **a11_data**: Save the files of all tablespaces which are not pure index tablespaces.
- **incr**: Incremental backup with RMAN. See: [Incremental Backup \[Page 108\]](#).
- **<file_ID>**: The file with this file ID will be processed. For data files, this is the ORACLE file ID. Control files can be addressed with the file ID 0. Online redo log files can be addressed using the file ID 0<n>, <n> is the redo log group number. Specify file ID 00 to process all existing online redo log files.
- **<file_ID>-<file_ID2>**: The files in this interval will be processed.
- **<generic_path>**: By entering a full path, you can process database files, non-database files, or the specified directory. By entering a generic path, you can process database data files whose name starts with that path. In this case, the path must contain at least the ORACLE_HOME directory and an additional generic specification (for example, **sapdata<n>**) in the path.
If the generic path is a directory you can also restore one or more files by the additional specification of one or more file names:
<directory_name>:<file_name1> :<file_name2>...]
- **sap_dir**: With this option, you can automatically determine and save all the files of the SAP environment. This means that the following directory trees are saved:
/sapmnt/<SAPSID>, **/usr/sap/<SAPSID>**, **/usr/sap/trans** This option could, for example be used after an SAP upgrade. Saving with the **sap_dir** option should not replace regular backups of the file systems using operating system commands.

You can only use this option when saving to tape and when not carrying out a verification of the backup.
- **ora_dir**: With this option, you can automatically determine and save all the non-database files of the ORACLE environment. This means that the directory trees under **<ORACLE_HOME>** (except for the directories **sapdata<n>** and **saplog-** or **origlog/mirrlog**) are saved. You could, for example, use this option after an ORACLE migration. Saving with the option **ora_dir** should not replace regular backups of the file systems with operating system commands.



You can only use this option when saving to tape and when not carrying out a verification of the backup.

- **<object list>**: You can enter an object list (see [Parameter Format \[Page 308\]](#)). This list can also include the key word **all**. However, SAP recommends processing database files and non-database files separately.



For UNIX systems: Start BRBACKUP to save the SAP/ORACLE environment (`backup_mode= sap_dir|ora_dir`) under user `root`, as otherwise you will not have the authorizations required for the directory to be saved.

Saving and restoring under `root` also has the advantage that you can be sure that the settings for the user and authorizations for the files and directories will be kept after restoring.

`backup_root_dir`

backup_root_dir

This is the parameter that is used only by BRBACKUP to identify the directory(-ies) in which database backups are carried out on hard disk. In exceptions (for example, no definition of the parameter `archive_copy_dir`), BRARCHIVE also uses the directories defined in this parameter if disk archiving is required. If you enter more than one directory, you must enclose the names in parentheses and separate them by commas.

Default: `<ORACLE_HOME>/sapbackup`

Required value: Directory(-ies) to which the backups carried out on hard disk should be written.

If you do not have enough storage space in one of the directories on your hard disk, you should provide directories on other disks and add their names in the parameter `backup_root_dir`. BRBACKUP then uses these directories for database backup.

See also:

[Backup on Several Disks \[Page 119\]](#)

backup_type

Identifies the default type of the database backup. This parameter is only used by BRBACKUP.

Default: offline

Possible values:

- **online**: Database backup in online mode, in other words, with the database running.
- **online_cons**: Database backup in online mode. As well as the database files, the offline redo log files generated during the backup are copied to the same volume. You then have a logically consistent dataset available. This backup of the offline redo log files with BRBACKUP runs completely independently of other BRARCHIVE backups.
- **offline**: Database backup in offline mode, in other words, the database is shut down during backup. When you select this parameter, BRBACKUP checks that no SAP system users are connected to the database. If an SAP System is active, the database is not shut down and BRBACKUP terminates the process with an error message (message number BR068E).
- **offline_force**: Database backup in offline mode, in other words, the database is shut down during backup. BRBACKUP always shuts down the database, even if the SAP System is active.
- **offline_standby**: Data backup of a standby database in offline mode; in other words, the standby database is shut down during the backup. This backup mode is only relevant for the disaster recovery configuration. See [Standby Database Scenario \[Page 185\]](#).
- **online_split**: The mirror disks are split and backed up while the database is running. The tablespaces to be backed up are only placed in BACKUP status during the split. This backup mode is only relevant for the 'split mirror' configuration. See: [Split Mirror Disks Backup \[Page 192\]](#).
- **offline_split**: The database is only shut down for the mirror disks split. The backup of the mirror disks can take place whilst the database is running. The R/3 System is available during the entire split mirror backup. Restriction: No transactions can be carried out during the short period of time that the database is shut down. This backup mode is only relevant for the 'split mirror' configuration. See: [Split Mirror Disks Backup \[Page 192\]](#).
- **offline_stop**: Database backup in offline mode without a consequent startup of the database. After its backup the database can be transferred directly into the mount standby status. This type of backup is only relevant in the following case: The productive database is saved and then takes over the role of a standby database. The backup itself becomes a productive system. See: [Standby Database Scenario: \[Page 185\]](#)

See also:

[Backup Mode \[Page 127\]](#).

compress

compress

Define whether files are to be compressed (software or hardware compression).

Default: no

Possible values:

- no: No file compression.
- yes: Software compression of files during backup.
- hardware: Should be set when tape units that support hardware compression are used. BRRESTORE handles this setting as **compress = no**. Just setting this parameter will not activate hardware compression. It is merely information for BRBACKUP/BRARCHIVE. You will also have to configure your backup device accordingly.
- only: Software compression, but no backup of files is started. This setting is not used by BRRESTORE.

Notes on Software Compression (Without Starting a Backup) with **compress = only**:

If you use tape units that support hardware compression, SAP recommends that you perform software compression of the entire database with parameter `compress = only` at least once a month, so that the current compression rate can be determined for each of the data files. When you use this parameter, backup is not started.

If you do not want to compress the entire database, you should at least compress those database files where a lot of changes take place. To determine these files, compare the compression rates of all files in two subsequent compression operations. The compression rates for files that have not changed will probably remain constant in the future, as well.

After extreme database changes (reorganization, release upgrade, transfer of large quantities of data), you should always start the software compression in order to determine the compression rates for the entire database.

The setting of parameter `backup_dev_type` is not relevant for this activity, since no backup is started.

Check the setting of option -b 12 in command [compress_cmd \[Page 321\]](#).

compress_cmd

Provides the command to be used for software file compression (if activated using the compression parameter such as: `compress = yes`).

Default value: None

If you want to use the software compression, you must enter a command.

The command must contain two \$ characters which stand for the file to be compressed and the compressed file.

You can enter the following command, for example:

```
compress_cmd = "compress -c $ > $"
```

Make sure that you adhere to the syntax rules when you enter parameters composed of several values - in particular, the double quotes " are essential.

Compression is performed on disk. You can specify the directory in which compression is to be performed in parameter `compress_dir`.

BRBACKUP replaces the first variable in the command with the source file which you want compressed. The second variable is replaced with the name of the compressed file. The second file name is assigned the extension ".Z".



Ensure that the compression command you enter does not delete the original of the compressed file. SAP therefore recommends that you **always use the option -c of the compress command** (when available) so that the original is not deleted.



If you use compression with the parameter `compress = only` or option `-k only`, SAP recommends using option `-b 12` of the compress command.

```
compress_cmd = "compress -b 12 -c $ > $"
```

The compression rates determined in this manner correspond much more closely to the actual hardware compression rates, and therefore enable optimized file distribution and volume load balancing.

`compress_dir`

`compress_dir`

Directory(-ies) in which file compression will be performed.

Default value: The directory(-ies) specified in parameter `backup_root_dir`

Because file compression is performed on disk, you must provide enough space to compress the largest file in your database.



If you want a parallel backup with software compression on local or remote tape devices, `<n>` copy processes are generally started in parallel, where `<n>` is normally the number of defined backup devices which are connected. In order to be able to use this parallel copy effectively also for the compression, you should define in directory `compress_dir` as many directories as there are copy processes. See also [exec_parallel \[Page 331\]](#).



If you check the readability and completeness of the backup using `-w|-verify` [\[Page 389\]](#), the files are restored back to the directory(-ies) defined in `compress_dir`. In order to carry out this process in parallel as effectively as possible, you should define as many directories as there are copy processes (the number of copy processes generally corresponds to the number of connected tape devices, see [exec_parallel \[Page 331\]](#)).

copy_in_cmd

Default value: None

You must enter a value before you switch to "remote piping" (`backup_dev_type = pipe|pipe_auto`).

This parameter specification is only useful in combination with the parameter `read_fifo_cmd` for reading from remote systems.

SAP recommends entering the command

```
copy_in_cmd = "dd bs=5k conv=block if=$",
```

to enable volume reading with the `dd` command. By using parameter `conv=block`, you can also import the backup locally, provided you remain with the default blocking of 5KB. You can also use higher blocking to improve performance, however, `conv=block` does not work as described above. The `$` character is replaced by the device address. Do not forget to enclose the parameter specifications composed of several values in double quotes.

`copy_out_cmd`

`copy_out_cmd`

Default value: None

You must enter a value before you switch to “remote piping”.

This parameter specification is only useful in combination with the parameter `read_fifo_cmd` for performing backups on remote systems.

SAP recommends entering the command

```
copy_out_cmd = "dd bs=5k conv=block of=$",
```

to enable volume reading with the `dd` command. By using parameter `conv=block`, you can also import the backup locally, provided you remain with the default blocking of 5KB. You can also use higher blocking to improve performance, however, `conv=block` does not work as described above. The `$` character is replaced by the device address. Do not forget to enclose the parameter specifications composed of several values in double quotes.

cpio_disk_flags

Default value: `-pdcu`

Required value: Flags for cpio commands which are used by the SAP utility programs if directories are copied to disk.



The call `brbackup -d disk -m /sapmnt/C11/bin` makes a backup copy of directory `/sapmnt/C11/bin` on disk. Parameter `cpio_disk_flags` must be set here.

cpio_flags**cpio_flags**

Default value: `-ovB`

Required value: Flags for cpio commands that the SAP utilities use.

The option `B` causes a block size of 5KB to be used. You can also use a higher block size (if `cpio` permits this for your hardware platform) to increase performance when writing to tape.

The block size in `cpio_flags` and `cpio_in_flags` has to be the same.

cpio_in_flags

Default value: `-iuvB`

Required value: Flags for cpio commands that the SAP utilities use for reading files from a volume.

The option `v` causes a block size of 5KB to be used. You can also use a higher block size (if `cpio` permits this for your hardware platform), to increase performance when writing to tape.

The block size in `cpio_flags` and `cpio_in_flags` must match.

dd_flags**dd_flags**

This parameter is only required if you are working with raw devices or if you are using the Parallel Server option. In order to access raw devices, use the dd command. See [Supporting the Raw Device Configuration with BRBACKUP and BRRESTORE \[Page 768\]](#).

Default value: "obs=16k"

Required value: Definition of the dd options required for writing to tape.

If you define the default value, the data is written to tape in blocks of 16 Kbyte.

dd_in_flags

This parameter is only required if you are working with raw devices or if you are using the Parallel Server option. In order to access raw devices, use the dd command. See [Supporting the Raw Device Configuration with BRBACKUP and BRRESTORE \[Page 768\]](#).

Default value: `"ibs=16k"`

Required value: Definition of the dd options required for reading from tape.

If you define the default value, the data is read from tape in blocks of 16 Kbyte.

dismount_cmd

dismount_cmd

In this parameter the dismount command for the automatic dismounting of a tape is defined. For `backup_dev_type = tape_box` the command is called locally; for `backup_dev_type = pipe_box` it is called on a machine defined in the parameter `read_fifo_cmd`.

Default value: none (for `backup_dev_type = tape_box | pipe_box` this parameter *must* be defined in the initialization profile).

```
dismount_cmd = "<dismount_cmd> $ $ [$]"
```

<dismount_cmd>: Command name, for example `dismount.csh`

Do not forget to enclose the parameter specifications composed of several values in double quotes. The \$ characters stand, in the following order, for:

1. the name of the database to be backed up
2. the address(es) of the tape device(s)
3. optional: name of a file for additional configuration parameters (parameter: `mount_par_file`).



The dismount or mount command must be created by the user in the form of a program, a shell script or a batch file.

If the dismount command has been performed successfully the exit code 0 is displayed. No other output is displayed. The only possible outputs not recognized as errors are messages beginning with the characters '#INFO'.

See also:

[Backup with Jukebox/Autoloader \[Page 197\]](#)

[Mount / Dismount Command \[Page 198\]](#).

[mount_cmd \[Page 333\]](#)

[mount_par_file \[Page 334\]](#)

exec_parallel

This parameter specifies the maximum number of parallel copy processes.

Default value: 0

The number of parallel copy processes corresponds to the number of backup devices available (tape devices/disks) in this case. If you use option `-k only` to determine the compression rates, the number of parallel copy processes corresponds to the number of disks (or logical volumes) on which the database files reside.

Required value: n

- Backup on tape

The value n should be less than or equal to the number of backup devices. If you define a value n less than the number of tape devices, this means that you can only use n of the available tape devices in parallel. If a tape change should then become necessary in one of the tape devices used in parallel, there is an automatic change to the next free backup device and the backup continues there.
- Backup to disk

The number of parallel copy processes can be greater than the number of disks defined in `backup_root_dir` (but not greater than 255). One or more disks are then written by several processes at the same time. If you choose the number of copy processes n to be less than the number of disks, this means that you can only use n of the available disks in parallel. If one of the disks used in parallel is full, there is an automatic change to the next disk which has not been used and backup continues there.
- Restoring

The maximum number of parallel copy processes which BRRESTORE can restore is the number of parallel copy processes used in the backup. The number of copy processes can be reduced.

`expir_period`

`expir_period`

This parameter specifies the expiration period for the volumes.

Default value: 30

Required values: Indicate the period (in days) for which the tapes to be used for the backup should be locked by entering a whole number.

Before BRBACKUP or BRARCHIVE start the backup to a volume, the system checks whether the expiration period set by `expir_period` for this volume has expired.

See also [Volume Expiration Period \[Page 147\]](#).

mount_cmd

In this parameter the dismount command for the automatic demounting of a tape is defined. The command is called locally for `backup_dev_type = tape_box`; for `backup_dev_type = pipe_box` it is called on a machine defined in parameter `read_fifo_cmd`.

Default value: none (in case of `backup_dev_type = tape_box | pipe_box` this parameter must be defined in the initialization profile).

```
mount_cmd = "<mount_cmd> $ $ $ [$]"
```

<mount_cmd>: Command name, for example `mount.csh`

Do not forget to enclose the parameter specifications composed of several values in double quotes.

The \$ characters stand, in the following order, for:

1. the name of the database to be backed up
2. the address(es) of the tape device(s)
3. the tape name(s)
4. optional: Name of a file for additional configuration parameters (parameter: `mount_par_file`)



The dismount or mount command must be created by the user in the form of a program, a shell script or a batch file.

If the mount command has been performed successfully the exit code 0 is displayed. No other output is displayed. The only possible outputs not recognized as errors are messages beginning with the characters **#INFO** .

See also:

[Backup with Jukebox/Autoloader \[Page 197\]](#)

[Mount / Dismount Command \[Page 198\]](#)

[dismount_cmd \[Page 330\]](#)

[mount_par_file \[Page 334\]](#)

mount_par_file

mount_par_file

This parameter is used in conjunction with the mount or dismount commands for automatic mounting or dismounting of tapes during backups with jukeboxes or autoloaders. The name of a parameter file is defined in `mount_par_file` or by the corresponding command option `-r|parfile`. This file contains additional configuration parameters for the mount or dismount commands.

See also:

[-r|-parfile \[Page 384\]](#)

[Mount / Dismount Command \[Page 198\]](#)

new_db_home

This parameter must be set if you want to make a database copy using BRBACKUP ([backup_dev_type \[Page 314\]](#) = `disk_copy`). The name of the home directory of the database copy is defined in `new_db_home`.

Default: none

`new_db_home` = `<dir>`

UNIX: `<dir>` is the new SAP database directory (`SAPDATA_HOME`)

NT : `<dir>` is the new SAP database directory: `<drive>\oracle\<SID>`

This directory must also contain the `sapbackup` directory.



Under Windows NT, the `sapdata` directories can be distributed across several drives. When making a database copy, a target drive can be specified for each drive (see [m|-mode \[Page 378\]](#)). If you do not specify a target drive, the files will be copied to the directory defined in the parameter.

See also:

[Structure Retaining Database Copy \[Page 130\]](#)

[The SAP Utilities under Windows NT \[Page 245\]](#)

parallel_instances

parallel_instances

This parameter is only relevant if you are using the Parallel Server option. See [ORACLE Databases and Parallel Server Option \[Page 769\]](#).

The instances running in parallel to the dedicated database instance are defined by this parameter.

Default value: none.

```
parallel_instances = <inst_beschr>|(<inst_beschr1>,<inst_beschr2>,...)
```

```
<inst_descr>: <ORACLE_SID>:<ORACLE_HOME>@<connect_string>
```

```
<ORACLE_SID>: ORACLE System ID of the parallel instance
```

```
<ORACLE_HOME>: ORACLE_HOME directory of the parallel instance
```

```
<connect_string>: ORACLE Connect-String to the parallel instance
```



```
parallel_instances =  
(C11_002:/oracle/C11@T:hs0002:C11_002,C11_003:/oracle/C11@T:hs  
0003:C11_003)
```


primary_db

This parameter is only relevant if the disaster recovery configuration is being used. See: [Standby Database Scenario \[Page 185\]](#), [Split Mirror Disks Backup \[Page 192\]](#).

The connect string to the primary database instance is defined with this parameter so that BRBACKUP can log onto the primary host.

Default value: None

```
primary_db = <connect_string>
```

```
<connect_string>: Connect string ('SQL*Net database specification  
string') from the standby host to the primary (production) database.
```



```
primary_db = C11
```



As for the OPS configuration, certain requirements must be satisfied for the connection to a remote host.

`remote_host`

remote_host

Use this parameter to specify the name of the remote host, if you want to make a backup to a remote disk.

Default: none

`remote_host = <host_name>`

The parameters `remote_host` and `remote_user` replace the parameter `read_fifo_cmd` which will no longer be used as of Release 4.5A.

See also:

[remote_user \[Page 339\]](#)

[Backup to a Remote Disk \[Page 120\]](#)

remote_user

Use this parameter to specify the user for a remote host, if you want to make a backup to a remote disk.

Default: none

```
remote_user = <user_name>
```

If you use the ftp network to transfer data (see [stage_copy_cmd \[Page 345\]](#)) you can also specify the password which BRBACKUP uses to log on to the remote host.

```
remote_user = "<user_name> <user_password>"
```

If you do not specify a password, BRBACKUP uses the password of the database user. If this is the case, the two user passwords must be the same.

The parameters `remote_user` and `remote_host` replace the parameter `read_fifo_cmd` which will no longer be used as of Release 4.5A.

See also:

[Backup to a Remote Disk \[Page 120\]](#)

[remote_host \[Page 338\]](#)

resync_cmd

resync_cmd

In this parameter the command for the synchronization of the mirror disks with the original disks is defined. If this parameter is not set the synchronization process is not executed under the control of BRBACKUP.

```
resync_cmd = "<resync_cmd> [$]"
```

<resync_cmd>: Program or shell script for synchronizing the mirror disks.

The '\$' character is optional. If it is set it replaces BRBACKUP in runtime with the name of a text file containing all file names to be backed up.



The resync command must be created by the user in the form of a program, a shell script or a batch file.

If the resync command has been performed successfully the exit code 0 is displayed. No other output is displayed. The only possible outputs not recognized as errors are messages beginning with the characters '#INFO'.

See also:

[Split Mirror Disks Backup \[Page 192\]](#)

[split_cmd \[Page 344\]](#)

rewind

Default value: None

Used by the SAP utilities to rewind a tape when a tape backup is performed. This parameter is also used when you use `pipe` or `pipe_auto` to back up to a remote system or restore file from that system.

The rewind command for your host operating system.

For example, you might enter:

```
rewind = "mt -t $ rew"
```



Rewind commands:

```
HP-UX: "mt -t $ rew"; AIX: "tctl -f $ rewind" ; OSF/1, SINIX, SUN, NT:  
"mt -f $ rewind"
```

Make sure that you do not forget the double quotes when you enter parameters composed of several values.

Make sure that you replace the device address with the character \$, as shown in the example. The SAP utilities replace the \$ character with the address of the device you use.

rewind_offline

rewind_offline

Default value: The value of the rewind parameter is used.

Used by BRBACKUP and BRARCHIVE to rewind a tape after a backup and set it to offline. This means that the volume is automatically ejected from the tape device (even when you are backing up to a remote system using `pipe` or `pipe_auto`).

Required value: The rewind/set offline command for your host operating system.

You could make the following entry, for example:

```
rewind_offline = "mt -t $ offl"
```



Rewind commands:

```
HP-UX: "mt -t $ offl"; AIX: "tctl -f $ offline"; OSF/1, SINIX, SUN,  
NT: "mt -f $ offline";
```

Make sure that you do not forget the double quotes when you enter parameters composed of several values.

Make sure that you replace the device address with the character \$, as shown in the example. The SAP utilities replace the \$ character with the address of the device you use.

See also:

[Automatic Tape Changing \[Page 222\]](#).

saveset_members

If you make a backup to tape with the Oracle Recovery Manager and want to use the file multiplexing option, then you can use this parameter to control the grouping of the savesets.

By setting the number of files to be saved in a single saveset, you can ensure that the speed of your tape device is optimal.

Default: 1 (no file multiplexing)

Possible values: `saveset_members = 1, 2, 3, 4, tsp, all`

See also:

[Saveset Grouping and Tape Layout \[Page 167\]](#)

split_cmd

split_cmd

This parameter is only relevant when using the split mirror disk configuration. See [Split Mirror Disks Backup \[Page 192\]](#)

This parameter contains a program or shell script (with or without options) called by BRBACKUP to split the mirror disks.

Default value: none

```
split_cmd = "<split_cmd> [$]"
```

```
<split_cmd>: Program or shell script to split the mirror disks.
```

The '\$' character is optional. If it is set it replaces BRBACKUP in runtime with the name of a text file containing all file names to be backed up.



The split command must be created by the user in the form of a program, a shell script or a batch file.

If the split has been performed successfully the exit code 0 is displayed. No other output is displayed. The only possible outputs not recognized as errors are messages beginning with the characters '#INFO'.

See also:

[Split Mirror Disks Backup \[Page 192\]](#)

stage_copy_cmd

Use this parameter to choose which transfer program will copy the data across the network to another host. This parameter is used when you make backups to remote disks.

Default: `rcp`

Possible values: `rcp`, `ftp`

An incremental backup to a remote disk is always made with `ftp`. If this is the case, this parameter is ignored.

See: [Backup to a Remote Disk \[Page 120\]](#)

`stage_root_dir`

`stage_root_dir`

This parameter identifies the directory(ies) used by BRBACKUP to back up database backups to a remote disk(s). In exceptions (for example, no definition of the parameter `archive_stage_dir`), BRARCHIVE also uses the directories defined in this parameter if disk archiving is required. If you enter more than one directory, you must enclose the names in parentheses and separate them by commas.

Default: The directory defined in the parameter [backup_root_dir \[Page 318\]](#).

Required value: Directory(-ies) to which the remote disk backups are to be written.

If necessary, you can make other directories on other remote disks available, and add their names to the parameter `stage_root_dir`. BRBACKUP then uses these directories for database backup.

See also:

[Backup to a Remote Disk \[Page 120\]](#)

tape_address

This parameter is used by the SAP utilities to identify the device addresses that will be used to write to a volume (tape). This parameter is also used when you back up to a remote system using `pipe` or `pipe_auto`.

Default value: None

Required values: Addresses of backup devices (tape devices, no rewind) that are to be used for backing up or restoring backups. If you specify more than one device, you must separate the names with commas and enclose the list in parentheses. Pay special attention to the differences in the device address names between tape devices with rewind and those with no rewind. Often, the only difference is that no-rewind addresses have an additional "n" in their name.



Specification of backup device:

HP-UX, SUN: `/dev/rmt/0mn`; AIX: `/dev/rmt0.l`; OSF/1: `/dev/nrmt0h`;
SINIX: `/dev/ios0/rstape005n`; NT: `/dev/nmt0`

When more than one address is specified, BRBACKUP carries out the backup on several backup devices simultaneously (see [Parallel Backup \[Page 119\]](#)). BRARCHIVE only uses the first of the specified device addresses (exception: the first two device addresses are used when the `-ss` or `-ssd` option is used). If parameters `tape_address_arch` and `tape_address_rew_arch` were defined, BRARCHIVE uses the devices defined there.

Under certain circumstances, BRRESTORE can also use several backup devices in parallel. See [Restoring Files \[Page 237\]](#).

The number of device addresses in `tape_address_rewind` must agree with the number of device addresses in `tape_address`.

`tape_address_arch`

`tape_address_arch`

BRARCHIVE uses this parameter to identify device addresses for writing to a volume (tape).

Default value: none

Possible values: Addresses of backup devices (tape devices, no rewind) to be used by BRARCHIVE. If this parameter is not set, BRARCHIVE uses the devices defined in parameter [tape_address \[Page 347\]](#).

The number of device addresses in `tape_address_rew_arch` must agree with the number of device addresses in parameter `tape_address_arch`.

tape_address_ctl

This parameter is used by SAP utilities to define the control drivers for the mount or dismount commands. These commands cause the tapes to be mounted or dismounted when using a jukebox or autoloader (`backup_dev_type = tape_box|pipe_box`).

Default value: none

```
tape_address_ctl = <tape_driver>|(<tape_driver1>,<tape_driver2>,...)
```

<tape_driver>: Control driver address

See also:

[Backup with Jukebox/Autoloader \[Page 197\]](#)

tape_address_ctl_arch

tape_address_ctl_arch

This parameter is used by BRARCHIVE to define the control drivers for the mount or dismount commands. These commands cause the tapes to be mounted or dismounted when using a jukebox or autoloader (backup_dev_type = tape_box|pipe_box).

Default value: none

If this parameter is not set BRARCHIVE uses the drivers specified in the parameter [tape_address_ctl](#) [Page 349].

```
tape_address_ctl_arch=  
<tape_driver>|(<tape_driver1>,<tape_driver2>,...)
```

<tape_driver>: Control driver address

[Backup with Jukebox/Autoloader](#) [Page 197]

tape_address_rew

This parameter is used by the SAP utilities to identify the device addresses that will be used to write to a volume (tape). This parameter is also used when you back up to a remote system using `pipe` or `pipe_auto`.

Default value: None

Required values: Addresses of backup devices (tape devices, no rewind) that are to be used for backing up or restoring backups. If you specify more than one device, you must separate the names with commas and enclose the list in parentheses.



Specification of backup device:

HP-UX, SUN: `/dev/rmt/0m`; AIX: `/dev/rmt0`; OSF/1: `/dev/rmt0h`; SINIX:
`/dev/ios0/rstape005`; NT: `/dev/mt0`

Also see [tape_address \[Page 347\]](#).

The number of device addresses in `tape_address_rewind` must agree with the number of device addresses in `tape_address`.

`tape_address_rew_arch`

`tape_address_rew_arch`

BRARCHIVE uses this parameter to identify device addresses for writing to a volume (tape).

Default value: none

Possible values: Addresses of backup devices (tape devices, rewind) to be used by BRARCHIVE. If this parameter is not set, BRARCHIVE uses the devices defined in parameter [tape_address_rew](#) [Page 351].

See also [tape_address_arch](#) [Page 348].

The number of device addresses in `tape_address_rew_arch` must agree with the number of device addresses in parameter `tape_address_arch`.

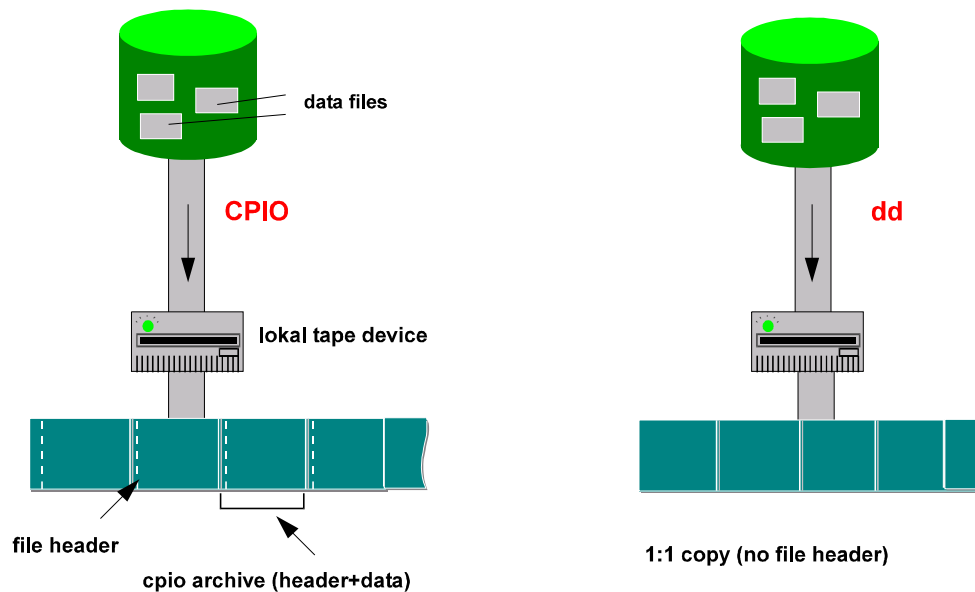
tape_copy_cmd

Command used to copy database files (not to raw devices) and non-database files from disk to tape.

Default: `cpio`

Possible values: `cpio`, `dd`

This parameter does not affect either raw devices (they are always copied with "`dd`") or directories (they are always copied with "`cpio`"). Tape header files (tape file label, `init_ora`, `init_sap`) and tape end files (`reorg_log`, `det_log`, `sum_log`) are always written with "`cpio`".



The use of `dd` commands to backup the database can prove advantageous to the performance and thus reduce the backup time.



The `dd` command reports an I/O-Error on UNIX platforms when it reaches the physical end of the tape. This message should not be confused with the same message when it is output for hardware problems. Check whether the end of the tape was really reached, taking the tape capacity into consideration. Reduce the value of parameter `tape_size` in this case. If a hardware problem is suspected, however, start the same backup to the same tape with `tape_copy_cmd = cpio` to identify the situation.



If you set the parameter `tape_copy_cmd = dd`, we recommend that you set parameters `dd_flags` and `dd_in_flags` as follows:

Unix systems:

```
dd_flags = "obs=xk bs=xk" dd_in_flags = "ibs=xk bs=xk"
```

tape_copy_cmd

Windows NT:

```
dd_flags = "bs=xk" dd_in_flags = "bs=xk" (with x >=16)
```

The `dd` options `obs` and `ibs` are not supported on Windows NT.

tape_pos_cmd

Command for positioning the read head on the tape.

Default value: None

This parameter is used by BRRESTORE and by BRBACKUP/BRARCHIVE with option `-w|verify`. For a file to be read from the volume, the tape must be fast-forwarded to the required position.

For example, you could make the following entry:

```
tape_pos_cmd = "mt -t $ fsf $"
```



Positioning commands:

```
HP-UX: "mt -t $ fsf $"; AIX: "tctl -f $ fsf $"; OSF/1, SINIX, SUN, NT:  
"mt -f $ fsf $";
```

Make sure that you do not forget the double quotes when you enter parameters composed of several values. The SAP utilities replace the first \$ character with the address of the device you use. The second \$ character is replaced by the number of files that are to be skipped.

`tape_size`

`tape_size`

Storage size in gigabytes (G), megabytes (M) or kilobytes (K) for the tapes that will be used for backups and for archiving redo log files.

Default value: **1200M**

The physical tape size (tape length * write density) is configured by this parameter. If you use software or hardware compression for files, this parameter specifies the total size of the files that will fit on the tape **after** compression. The largest file for backup should not be larger than the value specified in `tape_size` (after compression, when applicable).

See [Parameter `tape_size` \[Ext.\]](#).

tape_size_arch

This is the tape size in gigabytes (G), megabytes (M) or kilobytes (K) used by BRARCHIVE. This parameter should be used if different tape stations are addressed with parameters `tape_address_arch` and `tape_address_rew_arch` than with `tape_address` and `tape_address_rew`. Different tape station types can then be used for BRBACKUP and BRARCHIVE without risking a tape overflow.

Default value: the value set in [tape_size \[Page 356\]](#)

tape_use_count

tape_use_count

This parameter specifies the maximum number of times a volume can be written to. This is only a recommendation, and can be changed when the volume is still in a condition suitable for backup.

Default value: 100

See [Parameter tape_use_count \[Ext.\]](#).

uncompress_cmd

Provides the command used to decompress files that were backed up with software compression.

Default value: None

The command must contain two \$ characters, one for the file before decompression and one for after.

For example, you could enter the following command:

```
uncompress_cmd = "uncompress -c $ > $"
```

Make sure that you adhere to the syntax rules - in particular, the double quotes " are essential.

util_par_file

util_par_file

With this parameter, you can enter where the parameter file with the specifications that might be required for a backup with an external backup program is located.

Default value: None

Required value: Enter the directory and the name of the parameter file. If you have not entered a path, the system searches for the parameter file in directory `<ORACLE_HOME>/dbs`.

If you select the device type `backup_dev_type = util_file` or `util_file_online` (if supported), you initiate backup by a non-SAP backup program that is addressed through the BACKINT interface. Contact the supplier of these programs for the required specifications. To transfer additional parameters to the external backup program when you call one of the SAP utilities, you may have to create a parameter file with these values.

If possible, use parameter `util_file_online` for an online backup, since in this case the backup status of the individual tablespaces is set dynamically, which means that the volume of redo log entries is much smaller than if you use `util_file`.

See [Using External Backup Programs \[Page 241\]](#).

volume_archive

This parameter is used by BRARCHIVE to identify the volume/directory to be used for the archive of the offline redo log files.

Default value: None

The length of the volume name is limited to 10 characters.

Required values: Names of volumes (tapes) that are to be used for an archiving run. If you specify more than one volume, you must separate the names with commas and enclose the list in parentheses. You can also enter SCRATCH to deactivate automatic volume management.

Volume list:

When BRARCHIVE is started, the automatic volume management will check all the volumes in the sequence of their names in `volume_archive` (only volumes whose expiration period has expired will be suggested for archiving). The volumes will be suggested cyclically.



A valid volume name was found based on parameter `volume_archive` (that is, the expiration period for this volume has expired). The program assumes that you have mounted this volume. Once several check mechanisms have run, archiving is started on that volume.

See [Automatic Volume Management \[Page 153\]](#).

The expiration period can be configured using parameter `expir_period`. The automatic volume management ensures that only free volumes are offered for backup, e.g. those whose expiration period has expired. As a result, you will have to supply a quantity of volumes that is large enough for your selected expiration period.



You perform archives once a day, one volume (tape) is required; parameter `expir_period` is set to 14 (each volume is locked for 14 days at a time). In this case, you must specify at least 14 volumes to ensure that a volume is always available. Even better is to add a buffer of around 25% - e.g. specifying 18 or more volume names in this case.

SCRATCH

To deactivate the automatic volume management, use the command `volume_backup = SCRATCH`. In this case, you can mount any volume whose expiration period has expired. This name will then also be recorded during the archive.

volume_backup

volume_backup

This parameter is used by BRBACKUP to identify the volume/directory to be used for the backup of the database or non-database files.

Default value: None

The length of the volume name is limited to 10 characters.

Required values: Names of volumes that are to be used for database or other backups. If you specify more than one volume, you must separate the names with commas and enclose the list in parentheses. You can also enter SCRATCH to deactivate automatic volume management.

Volume list:

When BRBACKUP is started, the automatic volume management will check all the volumes in the sequence of their names in `volume_backup` (only volumes whose expiration period has expired will be suggested for backup). The volumes will be suggested cyclically.



You have two backup devices available, and three valid volumes were found in parameter `volume_backup` (the expiration period for these volumes has expired). The program assumes that you have mounted two of the selected volumes in the devices. If a third volume is required to complete the backup, the program will prompt you to mount the third selected volume at the appropriate time.

See [Automatic Volume Management \[Page 153\]](#).

The expiration period can be configured using parameter `expir_period`. The automatic volume management ensures that only free volumes are offered for backup, e.g. those whose expiration period has expired. As a result, you will have to supply a quantity of volumes that is large enough for your selected expiration period.



You perform a backup once a day, two volumes (tapes) are required; parameter `expir_period` is set to 14 (each volume is locked for 14 days at a time). In this case, you must specify at least 28 volumes to ensure that a volume is always available. Even better is to add a buffer of around 25% - e.g. specifying 35 or more volumes in this case.

SCRATCH

To deactivate the automatic volume management, use the command `volume_backup = SCRATCH`. In this case, you can mount any volume whose expiration period has expired. This name will then also be recorded during the backup.

Command Options for BRBACKUP, BRARCHIVE, and BRRESTORE

The SAP utilities BRBACKUP, BRARCHIVE, and BRRESTORE are configured by the initialization program `init<DBSID>.sap`. You can override many of these specifications by calling the individual programs with the appropriate command option (see [Effects of the Command Options \[Page 223\]](#)). All command options are described in this chapter.

[Command Options for BRBACKUP \[Page 364\]](#)

[Command Options for BRARCHIVE \[Page 390\]](#)

[Command Options for BRRESTORE \[Page 413\]](#)

Command Options for BRBACKUP

Command Options for BRBACKUP

You can start the SAP utility BRBACKUP without command options. If no values to the contrary have been configured in initialization profile **init<DBSID>.sap**, BRBACKUP will start an offline complete backup to a local tape device, with a storage capacity of 1200 MB and no file compression.

See [Initialization Profile init<DBSID>.sap \[Page 306\]](#).

For more information on the SAP utility BRBACKUP, refer to [Using BRBACKUP and BRARCHIVE \[Page 219\]](#).

If you use BRBACKUP with command options, these override the corresponding values in the initialization profile. To use the options, you can specify either letter indicated or the complete word.

```
brbackup
[-a|-archive [<bra_options>]]
[-b|-backup <log_name>|last]
[-c|-confirm [force]]
[-d|-device tape|disk|disk_copy|disk_standby|pipe|tape_auto|pipe_auto|
tape_box|pipe_box|util_file|util_file_online]
[-e|-execute <number>]
[-f|-fillup <log name1>[<log name2>,...]|
<No. of days>|last]
[-h|-help [version]]
[-i|-initialize [force|show]]
[-k|-compress no|yes|hardware|only]
[-l|-language E|D]
[-m|-mode all|<tablespace_name>|all_data|<file_ID>|<file_ID1>-
<file_ID2>|
<generic_path>|<object list>|sap_dir|ora_dir]
[-n|-number <number of tapes>]
[-o|-output dist|time[,time|dist]]
[-p|-profile <profile>]
[-q|-query [check]]
[-r|-parfile <parameter_file>]
[-t|-type online|online_cons|offline|offline_force|
offline_standby|offline_stop|online_split|offline_split]
[-u|-user [<user>[/<password>]]]
[-v|-volume <volume>|<volume list>|SCRATCH]
[-w|-verify]
```

Command Options for BRBACKUP

[-a|-archive \[Page 366\]](#)

[-b|-backup \[Page 367\]](#)

[-c|-confirm \[Page 369\]](#)

[-d|-device \[Page 371\]](#)

[-e|-execute \[Page 372\]](#)

[-f|-fillup \[Page 373\]](#)

[-h|-help \[Page 374\]](#)

[-i|-initialize \[Page 375\]](#)

[-k|-compress \[Page 376\]](#)

[-l|-language \[Page 377\]](#)

[m|-mode \[Page 378\]](#)

[-n|-number \[Page 380\]](#)

[-o|-output \[Page 381\]](#)

[-p|-profile \[Page 382\]](#)

[-q|-query \[Page 383\]](#)

[-r|-parfile \[Page 384\]](#)

[-t|-type \[Page 386\]](#)

[-u|-user \[Page 387\]](#)

[-v|-volume \[Page 388\]](#)

[-w|-verify \[Page 389\]](#)

-a|-archive

-a|-archive

Input syntax: `-a [<bra_options>]`

Default: BRBACKUP and BRARCHIVE are called separately, the backup of the data files and the offline redo log files takes place in separate backups on separate tapes.

When you specify the option `-a` BRARCHIVE is automatically started after the BRBACKUP backup of the data files. The offline redo log files, as well as all logs, are then copied by BRARCHIVE to the same tape(s) immediately after the backed up database files (as of Release 4.5A, this is also possible to disk, and with BACKINT). The main advantage of this procedure is that you only have to start or schedule one program (usually BRBACKUP). The second program is started automatically after the first program has ended. If the backup is made to tape, you can also use the tape capacity optimally.

This procedure enables both an unattended backup and an effective usage of tape capacity.

You can also specify other options for BRARCHIVE backups. If not the appropriate defaults are used. The option `-a` (including the additional options) must always be placed at the end of the BRBACKUP command line call.

See: [Command Options for BRARCHIVE \[Page 390\]](#) and [Initialization Profile init<DBSID>.sap \[Page 306\]](#)



Unattended online backup of a database in parallel on two tape devices followed by a startup of BRARCHIVE, in order to create two copies of the offline redo log files in parallel on the same tapes.

```
brbackup -m all -t online -c -a -ss -c
```

See also:

[BRBACKUP and BRARCHIVE Backups in One Run \[Page 136\]](#)

-b|-backup

Input syntax: `-b [<log_name> | last]`

Default: `last`

- `<log_name>`: The BRBACKUP disk backup with the log name entered in `<log_name>` is copied to the current backup tape.
- `last`: BRBACKUP selects the last successful disk backup and copies this to the current backup tape.



Backup of the last disk backup with the same parameters as defined in
`init<DBSID>.sap:`

```
brbackup -b
```

See also:

[Two-Phase Backup \[Page 124\]](#)

-bd|-backup_delete

-bd|-backup_delete

Input syntax: **-bd|-backup_delete** [**<log_name>** | **last**]

Default: **last**

- **<log_name>**: The BRBACKUP backup to disk with the log name entered in **<log_name>** is copied to the current backup tape.
- **last**: BRBACKUP selects the last successful backup to disk and copies this to the current backup tape.

After the backup the data is automatically deleted from the disk.



Backup of the last disk backup with the same parameters as defined in `init<DBSID>.sap` followed by deletion:

```
brbackup -bd
```

See also:

[Two-Phase Backup \[Page 124\]](#)

-c|-confirm

Input syntax: `-c [force]`

Default value: Confirmation messages are issued and user entry is expected.

When you specify the `-c` option, you suppress most of the confirmation messages that would normally be displayed during the execution of a backup. This enables you to run the program in unattended mode.

The following confirmation messages are not suppressed:

- Interactive password entry. To schedule a backup using CRON, use an appropriate script for entering the password. See [Unattended Backup \[Page 139\]](#).
- Prompt to mount the next volume, when more volumes are required than there are backup devices available. When a new volume must be mounted in the backup device during the backup, the utility responds as follows:
 - When a console is available, you can mount the next volume, make the entries required by the utility, and continue the backup.
 - When no console is available (the backup was scheduled using CRON or the CCMS, for example), the backup terminates.



A backup is scheduled using CRON. Three tapes are required for the backup. Two tape devices are available and you have mounted two of the requested tapes. The backup is started, but is terminated once the two tapes have been written.

- Prompt for mounting the next tape when a cpio continuation tape is requested (see [cpio Continuation Tape \[Page 224\]](#)). The backup is started. When the physical end of tape is reached, the utility responds as follows:
 - When a console is available, you can mount the continuation tape, make the entries required by the utility, and continue the backup.
 - When no console is available (the backup was scheduled using CRON or the CCMS, for example), the backup terminates.

To avoid terminations for the above reasons, use the option `-c force`.

force:

When the `-c force` option is used, the following confirmation messages are also suppressed:

- Prompt to mount the next volume. When more volumes are required than there are backup devices available, the backup is not even started.
- Prompt to mount a cpio continuation tape. If a continuation tape is required, the backup is terminated at this point, and an appropriate cpio error message is displayed.

The following confirmation prompt is not suppressed:

- Interactive password entry. To schedule a backup using CRON, use an appropriate script for entering the password. See [Unattended Backup \[Page 139\]](#).

If you want to run the SAP utility with, for example, the UNIX utility CRON, you should use the option `-c force`.

-c]-confirm

-d|-device

Input syntax:

```
-d tape|disk|pipe|disk_copy|disk_standby|tape_auto|pipe_auto|  
tape_box|pipe_box|util_file|util_file_online|stage|stage_copy|  
stage_standby|rman_prep|rman_util
```

Default: tape

The following backup media are supported:

- **disk**: Disk.
- **tape**: Tape in local tape device.
- **pipe**: Backup to a tape device of a remote system.
- **disk_copy**: Copying of database files to a disk with an identical directory structure. The name of the new Oracle_Home directory is defined in the parameter **new_db_home**.
- **disk_standby**: Copying of database files to a disk with an identical directory structure (compare **disk_copy**). To let you construct a standby database, a standby control file is generated and copied.
- **tape_auto** or **pipe_auto**: Prompts for changing the tape will be suppressed. This is only useful when you use a tape device with automatic tape changing (tape changing device).
- **tape_box** or **pipe_box**: Jukeboxes or autoloader tape devices which can be addressed locally or remotely. The drivers for the data transfer (**cpio**, **dd**) are defined in the parameters **tape_address** or **tape_address_arch**, the drivers for spooling are defined in the parameters **tape_address_rew** or **tape_address_rew_arch** and the drivers for mounting and dismounting the tapes are defined in the parameters **tape_address_ctl** or **tape_address_ctl_arch**.
- **util_file** or **util_file_online**: Use external backup programs for file-by-file backup. If you want to use BRRESTORE, you can also restore the files one by one. If you use this parameter, you may have to create a file containing the parameters required for that type of backup. If a parameter file of this type is required, you must specify the name of the file in the profile parameter **util_par_file** or with the option **-r**. Use parameter **util_file_online** for an online backup (if it is supported by the external backup program). This dynamically sets and ends the backup status of the tablespaces to be saved and thus greatly reduces the volume of offline redo log files during the backup.
- **stage|stage_copy|stage_standby**: See [backup_dev_type \[Page 314\]](#)
- **rman_util|rman_prep**: See [backup_dev_type \[Page 314\]](#)

See also:

Parameters in **init<DBSID>.sap**: [backup_dev_type \[Page 314\]](#).

-e|-execute

-e|-execute

Input syntax: `-e <n>`

Default value: 0

This parameter defines the maximum number of parallel copy processes. If the default value is set to 0, the number of parallel copy processes corresponds to the number of backup devices available (tape devices/disks). If the option `-k only` is used to determine the compression rates, the number of parallel copy processes corresponds to the number of disks (or logical volumes) on which the database files reside.

You can also define a different number `n` of copy processes. This will cause the following to happen:

- Backup to tape

The value `n` should be less than or equal to the number of backup devices. If you define a value `n` less than the number of tape devices, this means that you can only use `n` of the available tape devices in parallel. Should a tape change to one of the tape devices used in parallel be necessary, there is an automatic change to the next free backup device and the backup continues there.

- Backup to disk

The number of parallel copy processes can be greater than the number of disks defined in `backup_root_dir` (but not greater than 255). In this case, one or more disks will be written simultaneously by several processes. If you choose the number of copy processes `n` to be less than the number of disks, this means that you can only use `n` of the available disks in parallel. If one of the disks used in parallel is full, there is an automatic change to the next disk which is not used and the backup continues there.

Parameters in `init<DBSID>.sap`: [exec_parallel \[Page 331\]](#).

-f|-fillup

Input syntax: `-f <log_name1>[,<log_name2,...>] | <no._days> | last`

Default value: `last`

The completion of backups can refer to one or more defined backup(s) `<log name>`, to all backups in a defined number of preceding days `<no. days>` or to the last backup to be carried out.

The BRRESTORE option `-fillup` completes canceled BRRESTORE runs. See [-f|-fillup \[Page 422\]](#)

-h|-help

-h|-help

Input syntax: **-h** [**version**]

Default value: No help

Using this option, you can obtain an overview of all applicable options for BRBACKUP.

- **version**: To display detailed information on the versions of the program modules.

-ij-initialize

Input syntax: `-i [force | show]`

Default value: No initialization.

Use this option to initialize SAP volumes (tapes) non-SAP volumes. Only tapes of this type can be used for backup with BRBACKUP. See [Volume Management \[Page 142\]](#).

- **-i force:** This option is used to initialize new or non-SAP volumes. The expiration period is not checked. It is also possible to initialize SAP volumes with `-i force`. You should only use this option when absolutely essential, however, since the expiration period is not checked and the `tape_use_count` stored in the volume label is reset to one.
- **-i:** This option can only be used for SAP volumes. It is used mainly to rename volumes. BRBACKUP suggests the volume name specified in `volume_backup`. Mount the volumes with the appropriate names or other SAP volumes with labels. BRBACKUP checks whether the expiration period has expired. If so, BRBACKUP initializes the volume and assigns it the specified name.

For both the above options, you can also use in addition the option `-v`. The system then initializes the tapes with the names selected in `-v`.

You receive an error message from BRBACKUP and one of the following cpio messages if you want to start the backup program and want to use a non-initialized tape as a backup medium. Initialize this tape using the option `-i force` and then repeat the procedure.

The cpio message is platform-dependent and has, for example, the following texts

Operating System	Message
SINIX	cpio: Bad header - checksum error. cpio: Not a cpio file.
AIX	Cannot read from the specified input. Out of phase.
HP-UX	Out of phase -- get help. End of volume.
OSF1	Not a cpio archive.
SUN	cpio: Bad header - checksum error. cpio: Not a cpio file, bad header.
NT	cpio: Read error on file...: No more data on tape. cpio: Bad magic number in archive.

- **-i show:** Used to display the volume label information. No initialization is executed: the volume remains unaltered.

-k|-compress

-k|-compress

Input syntax: `-k no|yes|hardware|only`

Default value: `no` (no compression)

If you specify this option, you can compress your data files before the backup is performed.

- **yes**: Software compression.
- **hardware**: Hardware compression. Prerequisite: the tape device must support hardware compression and hardware compression must be active. See [Tape Devices with Hardware Compression \[Page 115\]](#)
- **only**: Determine the current compression rate of the individual data files. In this case, no backup is started. If you use tape devices with hardware compression, SAP recommends repeating this procedure about once a month. [Hardware Compression \[Page 232\]](#)

Also see [compress_cmd \[Page 321\]](#).

Parameters in `init<DBSID>.sap`: [compress \[Page 320\]](#).

-l|-language

Input syntax: **-l** **E** | **D**

Default value: **E**

You can determine whether you want the BRARCHIVE messages to appear in English (E) or in German (D).

The default becomes invalid if you specify another value by setting the environment variable `BR_LANG` (language variable). If you set option **-l**, the value specified with this option applies.

-m|-mode

-m|-mode

Input syntax:

```
-m all | <tablespace_name> | all_data | incr | full | <file_ID> | <file_ID1>-<file_ID2> | <generic_path> | <object_list> | sap_dir | ora_dir
```

Default: all

You can perform a full database backup or back up specific tablespaces or files (whether part of the database or not). You can create object lists.

You can specify what you want to back up:

- **all**: Back up the complete database

In a [Structure Retaining Database Copy \[Page 130\]](#) (`backup_dev_type = disk_copy` or `disk_standby`) you can retain the distribution of the `sapdata` directories to different drives (only for Windows NT).



The files of the drive `d` are copied to drive `k`, the files of the drive `e` are copied to the drive `l` and the files of the drive `f` are copied to the drive `m`.

```
brbackup -m all,d:=k:,e:=l:,f:=m:
```

If you do not specify a target drive, the files will be copied to the directory defined in the parameter [new_db_home \[Page 335\]](#).

- **<tablespace name>**: Back up the files of one tablespace.
- **all_data**: Back up the files of all tablespaces, except for pure index tablespaces.
- **incr**: Incremental backup with RMAN. See [Incremental Backup \[Page 108\]](#)
- **full**: See [Incremental Backup \[Page 108\]](#)
- **<file_ID>**: Back up a data file with the specified ORACLE file ID as file ID. Control files can be addressed with the file ID 0. Online redo log files can be addressed using the file ID 0<n>, <n> is the redo log group number. To address all the online redo log files, use file ID 00.
- **<file_ID1>-<file_ID2>**: Back up the files specified in the file ID interval. The specified file IDs must be known in the database.
- **<generic_path>**: Enter a complete path to back up the required database file, non-database file, or directory. Specify a generic path to back up all the database data files whose name starts with that path. In this case, the path must contain at least the ORACLE_HOME directory and an additional generic specification (for example, `sapdata<n>`) in the path.



When you specify a directory to be backed up its contents and the names of the subdirectories are backed up. However the directory structure and the content of the subdirectories are not backed up.

-m|-mode

- **<object list>**: You can specify a list of tablespaces or files, or combine the key word **all** with an object list. The individual objects are separated by commas (commas only, no blanks!).
- **sap_dir**: With this option, you can automatically determine and save all the files of the SAP environment. This means that the following directory trees are saved: `/sapmnt/<SAPSID>`, `/usr/sap/<SAPSID>` and `/usr/sap/trans`. If possible, these directories should be backed separately. You can only use this option when saving to tape without verifying the backup.
- **ora_dir**: With this option, you can automatically determine and save all the non-database files of the ORACLE environment. This means that the directory trees are saved in `<ORACLE_HOME>` (except for the `sapdata<n>` and `saplog-` or `origlog/mirrlog` directories). If possible, save these directories in a separate backup run. You can only use this option when saving to tape without verifying the backup.



For UNIX systems: Start BRBACKUP to save the SAP/ORACLE environment (`brbackup -m sap_dir|ora_dir`) under user `root`, as otherwise you will not have the authorizations required for the directory to be saved.

Saving and restoring under `root` also has the advantage that you can be sure that the settings for the user and authorizations for the files and directories will be kept after restoring.

Parameters in `init<DBSID>.sap`: [backup_mode \[Page 316\]](#).

If you want to repeatedly back up several tablespaces and/or files, it may be more effective to configure parameter `backup_mode` of the initialization profile accordingly.

-n|-number

-n|-number

Input syntax: `-n <number of tapes>`

Default value: 10,000

Define the number of volumes tapes you want initialized (you can use this option only together with one of the options `-i`, `-i force`, or `-i show`).

The default value is 10,000. The program processes all existing tapes. (i.e. in the `init<DBSID>.sap` parameter `volume_backup` or those defined with the option `-v`). To initialize only a specific number of tapes, change this value to meet your requirements.

-o|-output

Input syntax: `-o dist|time[,time|dist]`

Default value: The BRBACKUP detail log is written in normal form. See [Detail Log \[Page 440\]](#).

Using this option causes the detail log to be extended as follows:

- **dist**: The SAP utility generates information about the distribution of the files for backup among the volumes (tapes or disks) to be used.
- **time**: The SAP utility generates additional time stamps that enable you to determine the time required for the individual operations. Among other things, you can then determine the pure backup time for a file. After a successful backup, this information is included in list form for all backed up files.

See [Log Supplements \[Page 442\]](#).

-p|-profile

-p|-profile

Input syntax: `-p <profile>`

Default value: `init<DBSID>.sap`

This profile is contained in directory `<ORACLE_HOME>/dbs`.

If you want to use a different profile, specify the name of the profile file here. If this file is not in the standard directory `<ORACLE_HOME>/dbs`, specify the complete path.

`-q|-query`

Input syntax: `-q [check]`

Default value: The backup is started.

When you select the `-q` option, you find out which volumes (e.g. tapes) must be mounted for the backup process. In this case, backup is not started. Before you start a backup request with CRON, you should use this option to find out which volumes are required.

- **check:** You can check whether the proper volumes have really been mounted in the backup devices. The backup is not started.

The preparation for an unattended backup to tapes could be performed as follows:

- `brbackup -q check` to query the required tapes.
- Mount the required tapes in the tape devices.
- Enter `cont` to start the check of the mounted tapes.

Once you have made these preparations, you can start an unattended backup on the same day, since you have already checked the validity of the tapes.

-r|-parfile

-r|-parfile

Input syntax: `-r <parameter_file>`

Default value: No parameter file

If you want to carry out a BRBACKUP backup using backup devices such as jukeboxes or autoloaders, you can define the name of a parameter file with the option `-r` or in the initialization profile, which contains the configuration parameters for the mount or dismount command.

When using non-SAP backup programs for the backup, you might have to store the additional information required in a parameter file by specifying the option `-d util_file|util_file_online`. You can enter the complete names of this file in the initialization profile or using the option `-r`. The contents of this file depend on the external backup program used. The SAP utilities only pass on the information about the parameter file that the external program can use to obtain the required information. To find out which parameters must appear in this file and the syntax of those parameters, please contact the supplier of the external backup program.

Parameters in `init<DBSID>.sap`:

[mount_par_file \[Page 334\]](#) (for BRBACKUP backups)

[util_par_file \[Page 360\]](#) (for external backup programs)

-s|-saveaset

Input syntax: `-s|-saveaset_members|saveaset_members = 1|2|3|4|tsp|all`

Default: `-saveaset_members = 1`

The SAP backup library helps to optimize the utilization of quick tape drives by combining multiple data files in saveasets. Multiple file access (file multiplexing) maximizes the flow of data (streaming mode).

The command option is `brbackup -s|-saveasets`.

A saveaset can contain individual data files, all files of a tablespace, or the complete data backup. The size of the saveasets for the backup must be selected according to the tape device. A fast data flow with a minimum saveaset size is the optimum.



Large saveasets are not recommended, since in a restoration the complete saveaset has to be imported, even if only one data file is required.

`saveaset_members = all` is set as standard for an incremental backup with the SAP backup library so that only one "incremental saveaset" is created including all changed blocks.

For more information, refer to [Saveaset Grouping \[Page 167\]](#).

-t|-type

-t|-type

Input syntax: `-t online|online_cons|offline|offline_force|offline_standby|offline_stop|online_split|offline_split`

Default value: `offline`

Specify the backup mode.

- **online**: The backup is carried out for the active database.
- **online_cons**: The backup is carried out whilst the database is active. As well as the database files the offline redo log files generated during the backup are also copied to the same volume. This means that you have a logically consistent dataset available. This backup of the offline redo log files using BRBACKUP runs completely independently of other BRARCHIVE backups.
- **offline**: The database is shut down for the backup when the SAP System has also been shut down. Otherwise, the database is not shut down, and BRBACKUP terminates with an appropriate error message (BR068E).
- **offline_force**: The system does not check whether an SAP System user is active. The database is shut down and an offline backup is performed.
- **offline_standby**: The standby database is stopped for the backup. This option is only relevant for the disaster recovery configuration. See [Standby Database Scenario \[Page 185\]](#)
- **offline_stop**: Database backup in offline mode, followed by the migration of the saved database into mount standby status. This type of backup is only relevant in the following case: The productive database is saved and then takes over the role of a standby database. The backup itself becomes a productive system. See: [Standby Database: Backup with BRBACKUP \[Page 190\]](#)
- **online_split**: The splitting and saving of the mirror disks is carried out whilst the database is active. This option is only relevant for the split mirror configuration. See: [Split-Mirror-Online-Sicherung \[Page 193\]](#).
- **offline_split**: The database is stopped for the splitting of the mirror disks. The backup of the mirror disks is then either done directly afterwards or at a later date, whilst the database is active. This option is only relevant for the split mirror configuration. See: [Split-Mirror-Offline-Sicherung \[Page 194\]](#).

Parameters in `init<DBSID>.sap:backup_type [Page 319]`

-u|-user

Input syntax: `-u [<user>[/<password>]]`

Default value: `system/manager`

User name and password used by the ORACLE database administrator to log on to the database system.

If you only enter `-u`, an interactive query of the user ID and the password is carried out by the SAP utility. You can enter the user name and the password separately (only enter the user name or the option `-u <user>`). The utility then prompts entry of the password. In this case, the password is not displayed during entry, and does not appear in the process list.

These measures are taken to protect the DBA password.

In shell scripts, you can structure the call as follows (example: BRBACKUP):

```
brbackup -c -u <<END  
<user>/<password>  
END
```

However, use this command only if the option `-c` is active and you are sure that the tape does not need to be changed.



If you are working with an OPS\$ user, use the following call:

```
BS> brbackup -u /.
```

In this case, BRBACKUP tries to log on to the database as OPS\$ user (see ORACLE documentation and information in the SAPNet Notes database. The OPS\$ user must be defined in the database and have at least SYSOPER authorization. With this method, it is not necessary to specify the password when calling BRBACKUP.

-v|-volume

-v|-volume

Input syntax: `-v <volume>|<volume list>|SCRATCH`

Default value: defined in `init<DBSID>.sap`

The length of the volume name is limited to 10 characters.

- **volume:** Enter the volume(s) you want to use for backup. BRBACKUP checks whether the mounted volume (tape) has that name and whether the expiration period has expired. The individual objects in a volume list are separated by commas (only commas - no blanks!).
- **SCRATCH:** By entering the reserved name `SCRATCH`, you can mount any volume (whose expiration period has expired) for the backup.

If you want to use the standard volume(s) specified in parameter `volume_backup` of the initialization profile, do not use the parameter `SCRATCH`. In this case, the automatic volume management will cyclically select the volumes named in `volume_backup` and check whether their expiration period has expired. If a free volume (e.g. with expired expiration period) is found, you are prompted to mount it in the backup device (tape device). BRBACKUP checks the name of the mounted volume and makes sure the expiration period really has expired. The expiration period is configured with parameter `expir_period` in profile `init<DBSID>.sap`.

By using the reserved name `SCRATCH`, you can deactivate the automatic volume management. You can then mount any SAP volume. The program still makes sure that the expiration period has expired.

See [Volume Management \[Page 142\]](#).

Parameters in `init<DBSID>.sap`: [volume_archive \[Page 361\]](#) or [volume_backup \[Page 362\]](#).

-w|-verify

Input syntax: `-w [use_dbv|only_dbv]`

Default: No checks.

This option can be used to make sure that the backup is readable and complete. Once the backup phase is complete, all the saved files are restored from the volume (for example, tape) in sequence (provided they were saved on a backup device), decompressed (when **compress = yes** was used), read by the check program, and compared with the originals. See also [compress_dir \[Page 322\]](#).

During an offline backup, the file contents are compared in binary form. During an online backup, the sizes of the saved files are determined and checked.

See [Log Supplements \[Page 442\]](#)

use_dbv: Database backup followed by a restoration to a temporary directory and a check of the ORACLE block structure with the tool DB_VERIFY from Oracle.

only_dbv: Online block structure check on the database files with DB_VERIFY without backup:

Note that verification will approximately double the backup time required.

For security reasons, we recommend using this option at least once within the volume expiration period for your complete backups. It enables you to detect any possible hardware problems and undertake the appropriate measures.

See also:

[Backup Verifications \[Page 155\]](#)

Command Options for BRARCHIVE

Command Options for BRARCHIVE

You can start the SAP utility BRBACKUP without command options. If no values to the contrary have been configured in initialization profile `init<DBSID>.sap`, BRARCHIVE will start archiving all the offline redo log files to tape, with a storage capacity of 1200 MB and no file compression.

See [Initialization Profile `init<DBSID>.sap` \[Page 306\]](#).

For more information on the SAP utility BRARCHIVE, refer to the sections [Using BRBACKUP and BRARCHIVE \[Page 219\]](#).

If you use BRARCHIVE with command options, these override the corresponding values in the initialization profile. To use the options, you can specify either letter indicated or the complete word.

brarchive

```
[ -a | -archive [[<DBSID>], <log_nr> | [<DBSID>], <log_nr1> - <log_nr2> |
<log_nr_list> ]
[ -b | -backup [<brb_options> ] ]
[ -c | -confirm [<force> ] ]
[ -d | -device tape | disk | pipe | tape_auto | pipe_auto |
tape_box | pipe_box | util_file ]
[ -f | -fill [<number> | stop ] ]
[ -h | -help [<version> ] ]
[ -i | -initialize [<force> | show ] ]
[ -k | -compress no | yes | hardware | only ]
[ -l | -language E | D ]
[ -m | -modify [<delay> ] ]
[ -n | -number <number of logs> | <number of tapes> ]
[ -o | -output dist | time [, time | dist ] ]
[ -p | -profile <profile> ]
[ -q | -query [check] ]
[ -r | -parfile <parameter_file> ]
[ -s | -save | -sc | -second_copy | -ds | -delete_saved | -dc |
-delete_copied | -sd | save_delete | -scd | -second_copy_delete | -ss |
-double_save | -ssd | -double_save_delete | -cs | -copy_save | -cgs | -
copy_delete_save ]
[ -u | -user [<user> [ / <password> ] ] ]
[ -v | -volume <volume> | <volume list> | SCRATCH ]
[ -w | -verify ]
```

[-a|-archive \[Page 392\]](#)

[-b|-backup \[Page 393\]](#)

[-c|-confirm \[Page 394\]](#)

[-d|-device \[Page 395\]](#)

[-f|-fill \[Page 396\]](#)

[-h|-help \[Page 397\]](#)

[-i|-initialize \[Page 398\]](#)

[-k|-compress \[Page 399\]](#)

[-l|-language \[Page 400\]](#)

Command Options for BRARCHIVE

[-m|-modify \[Page 401\]](#)

[-n|-number \[Page 402\]](#)

[-o|-output \[Page 403\]](#)

[-p|-profile \[Page 404\]](#)

[-q|-query \[Page 405\]](#)

[-r|-parfile \[Page 406\]](#)

[-s|-sc|-ds|-dc|-sd|-scd|-ss|-ssd|-cs|-cbs \[Page 407\]](#)

[-u|-user \[Page 410\]](#)

[-v|-volume \[Page 411\]](#)

[-w|-verify \[Page 412\]](#)

-a|-archive

-a|-archive

Input syntax:

```
-a|-archive [[<SID>],<log_nr>| [<SID>],<log_nr1>-<log_nr2>|  
<log_nr_list>]
```

Default value: BRARCHIVE saves all offline redo log files to tape, which have not already been saved to tape. Both the copies saved directly to tape as well as those written to tape via a disk backup are taken into account. At the most two copies can be written to tape.

Specifying the offline redo log files to be backed up via the entry <log_nr> is only sensible in exceptional cases.

The option -a|-archive can only be used in combination with the following BRARCHIVE-options:

```
brarchive -s| -sc| -ss.
```



A copy of the offline redo log files which were copied to disk in a previous backup is created in parallel on each of two different tapes:

```
brarchive -ss -a
```

See also:

[Two-Phase Backup \[Page 124\]](#)

-b|-backup

Input syntax: `-b [<brb_options>]`

Default: BRBACKUP and BRARCHIVE are called separately, the backup of the data files and the offline redo log files takes place in separate backups on separate tapes.

When you enter the option `-b` BRBACKUP is automatically started after the BRARCHIVE backup of the offline redo log files. After the offline redo log files, BRBACKUP copies the data files, as well as all logs to the same tape(s) (as of Release 4.5A this is also possible to disk, and with BACKINT). The main advantage of this procedure is that you only have to start or schedule one program (usually BRBACKUP). The second program is started automatically after the first program has ended.

If the backup is made to tape, you can also use the tape capacity optimally.

You can also specify other options for BRBACKUP backup. If you do not specify any options, the relevant default is used. The option `-b` (including the additional options) must always be placed at the end of the BRARCHIVE command line call.

See: [Command Options for BRBACKUP \[Page 364\]](#)



If you want to execute the BRBACKUP and BRARCHIVE backup procedure in one run, then we recommend that you carry out the tape management under control of BRBACKUP (`brbackup -a|-archive [Page 366]`).



Unattended backup of the offline redo log files on a tape followed by a startup of an offline backup on the same tape.

```
brarchive -sd -c -b -m all -t offline -c
```

See: [BRBACKUP and BRARCHIVE Backups in One Run \[Page 136\]](#)

`-c|-confirm`

`-c|-confirm`

Input syntax: `-c [force]`

Default value: Confirmation messages are issued and user entry is expected.

See [-c|-confirm \[Page 369\]](#).

-d|-device

Input syntax:

`-d tape|disk|pipe|tape_auto|pipe_auto|tape_box|pipe_box|util_file`

Default value: `tape`

The following backup media are supported:

- **tape**: Tape in local tape device.
- **disk**: Disk. You should normally only use the option for archiving to disk in exceptional cases.

Examples:

Requirements in connection with a parallel server configuration. See [Mixed Scenario \[Page 780\]](#).

Two-step archiving. See [archive_copy_dir \[Page 311\]](#)

Offline redo log files should always be archived to tape in the end, if possible twice.

- **pipe**: Archiving on a tape device of a remote system.
- **tape_auto** or **pipe_auto**: The BRARCHIVE program generally only uses one tape for archiving (exception: archiving with `-ss` or `-ssd`). No continuation tapes are created. Therefore, specifying `tape_auto` or `pipe_auto` has no effect.
- **tape_box** or **pipe_box**: Jukeboxes or autoloader tape devices which can be addressed locally or remotely. See: [Backup with Jukebox/Autoloader \[Page 197\]](#)
- **util_file**: see the corresponding option in [-d|-device \[Page 371\]](#).

See also:

Parameters in `init<DBSID>.sap`: [backup_dev_type \[Page 314\]](#).

-f|-fill

-f|-fill

Input syntax: `-f [<number> | stop]`

Default value: Do not wait for the next offline redo log files to write them to the volume (tape). BRARCHIVE must be started afresh for the backup of the next offline redo log files.

When you select the `-f` option, BRARCHIVE waits for the next offline redo log files copied by ORACLE to the archiving directory, and archives them on the volume as soon as they are created. This process is not stopped until the volume is full or the specified number of offline redo log files for processing has been reached.

- **<number>**: BRARCHIVE does not archive each offline redo log file individually, but waits until a certain number (<number>) of files have accumulated in the archiving directory. These offline redo log files are then saved to tape as a group. See [Grouping Offline Redo Log Files with brarchive -fill \[Page 217\]](#).
- **stop**: A BRARCHIVE archiving run that was started with `-f` can be stopped properly using `-f stop`. This operation may take a while (some seconds/ a few minutes). Inspect the log written by BRARCHIVE. See [Names of the Detail Logs \[Page 446\]](#).

-h|-help

Input syntax: **-h** [**version**]

Default value: No help

See [-h|-help \[Page 374\]](#).

-ij-initialize

-ij-initialize

Input syntax: `-i [force | show]`

Default value: no initialization.

Use this option to initialize SAP volumes (tapes) or non-SAP volumes. Only tapes of this type can be used for archiving with BRARCHIVE. See [Volume Management \[Page 142\]](#).

If you specify the option `-i` to initialize volumes (and `-v` has not been specified), BRARCHIVE initializes the volumes specified in `volume_archive`.

See [-ij-initialize \[Page 375\]](#).

-k|-compress

Input syntax: `-k no|yes|hardware|only`

Default value: `no` (no compression)

See [-k|-compress \[Page 376\]](#).

The option, `-k only`, has practically no meaning for BRARCHIVE as the information is not saved in the database. See [Hardware Compression \[Page 232\]](#)

See [-k|-compress \[Page 376\]](#)

-l|-language

-l|-language

Input syntax: -l **E** | **D**

Default value: **E**

See [-l|-language \[Page 377\]](#).

-m|-modify

Input syntax: `-m [<delay>]`

Default value: No delay

Delay: The offline redo log files which are created are sent to the standby database before they are processed. This could happen with a delay time of <delay> minutes after creating the ORACLE offline redo log file.

If there is a standby database, `brarchive -m` *must* be called in order to transfer the offline redo log files.

See: [Standby Database Scenario \[Page 185\]](#)

-n|-number

-n|-number

Input syntax: `-n <number of logs>|<number of volumes>`

Default value: 10,000

- **number of logs:** Specify the number of offline redo log files to archive and/or delete.
- **number of volumes:** Together with the option `-i`, `-i force` or `-i show`, you can use `-n` to specify the number of volumes (tapes) to initialize.

The default value is 10,000. The program processes practically all the existing offline redo log files (or volumes for volume initialization). In other words, the files in the `init<DBSID>.sap` parameter `volume_archive` or those defined with the option `-v`.

You can also use the option `-c` to enable BRARCHIVE to perform unattended archiving of the offline redo log files.

-o|-output

Input syntax: `-o dist|time[,time|dist]`

Default value: The BRARCHIVE detail log is written in normal form. See [Detail Log \[Page 447\]](#).

See [-o|-output \[Page 381\]](#).

-p|-profile

-p|-profile

Input syntax: -p <profile>

Default value: `init<DBSID>.sap`

See [-p|-profile \[Page 382\]](#).

-q|-query

Input syntax: `-q [check]`

Default value: Archiving is started.

The preparation for an unattended archiving to tapes could be performed as follows (archiving is not started):

- `brarchive -q check` to query the required tapes.
- Mount the required tape.
- Enter `cont` to start the check of the mounted tape.

See [-q|-query \[Page 383\]](#).

-r|-parfile

-r|-parfile

Input syntax: `-r <parameter_file>`

Default value: No parameter file

See [-r|-parfile \[Page 384\]](#).

-s|-sc|-ds|-dc|-sd|-scd|-ss|-ssd|-cs|-cbs

-s|-sc|-ds|-dc|-sd|-scd|-ss|-ssd|-cs|-cbs

Possible options:

`-s|-save|-sc|-second_copy|-ds|-delete_saved|-dc|-delete_copied|-sd|save_delete|-scd|-second_copy_delete|-ss|-double_save|-ssd|-double_save_delete|-cs|-copy_save|-cbs|-copy_delete_save`

Default value: `-s`

You can start the following operations:

- `-s|-save`
Archive the offline redo log files.
- `-sc|-second_copy`
Create a second copy of the offline redo log files which were already archived.
- `-ds|-delete_saved`
Delete offline redo log files that have been archived once.
- `-dc|-delete_copied`
Delete offline redo log files that have been copied twice.
- `-sd|-save_delete`
Archive offline redo log files and then delete these files.
- `-scd|-second_copy_delete`
Create a second copy of the offline redo log files that have already been archived and then delete these files.
- `-ss|-double_save`
Archive the offline redo logs to two backup devices (tape devices) in parallel.
- `-ssd|-double_save_delete`
Archive the offline redo logs to two backup devices (tape devices) in parallel and then delete the files.
- `-cs|-copy_save`
Create a second copy of offline redo log files that have already been archived and then archive the newly created offline redo log files.
- `-cbs|-copy_delete_save`
Create a second copy of offline redo log files which were already archived. These are then deleted and archiving of the newly created offline redo log files is begun.

BRARCHIVE will only perform the selected operation for the number of offline redo log files that you selected with the option `-n`. The program will only delete offline redo log files if they have already been archived.

If only one tape device exists, you can use the option `-cs` or `-cbs` to ensure that BRARCHIVE creates a second copy of the offline redo log files in a run, deletes them if necessary and immediately continues with archiving. You can also obtain this effect by calling BRARCHIVE first with the option `-s` and then with the option `-sc` or `-scd`. However, in this case two BRARCHIVE calls are required.

Parameters in `init<DBSID>.sap`: [archive function \[Page 312\]](#).

-s|-sc|-ds|-dc|-sd|-scd|-ss|-ssd|-cs|-cbs

-sd|-ds|-dc|-scd|-cde|-ssd

Input syntax:

```
-sd|-save_delete|-ds|-delete_saved|-dc|-delete_copied|  
-scd|-second_copy_delete|-cde|-copy_delete_save|  
-ssd|-double_save_delete
```

You can perform the following operations with automatic deletion:

- **-sd|-save_delete**
Archive offline redo log files and then delete these files.
- **-ds|-delete_saved**
Delete offline redo log files that have been archived once.
- **-dc|-delete_copied**
Delete offline redo log files that have been copied again.
- **-scd|-second_copy_delete**
Create a second copy of the offline redo log files that have already been archived and then delete these files.
- **-cde|-copy_delete_save**
Create a second copy of offline redo log files which have already been archived. These are then deleted and archiving of the offline redo log files which have been created in the meantime is started.
- **-ssd|-double_save_delete**
Archive the offline redo logs to two backup devices (tape devices) in parallel and then delete the files.

Use the option **-n** to set the number of offline redo log files for which you want BRARCHIVE to perform your selected operation. The program will only delete offline redo log files if they have already been archived.

If only one tape device exists, you can use the option **-cde** to make sure that BRARCHIVE creates a second copy of the offline redo log files, deletes them if necessary and immediately continues with archiving in a single run. You can also achieve this by calling BRARCHIVE first with the option **-s** and then with the option **-scd**. However, in this case you need to make two BRARCHIVE calls.

Parameters in `init<DBSID>.sap`: [archive_function \[Page 312\]](#)

For more information, see [-s|-sc|-ss|-cs \[Page 407\]](#).

-u|-user

-u|-user

Input syntax: -u [<user>[/<password>]]

Default value: `system/manager`

See [-u|-user \[Page 387\]](#).

-v|-volume

Input syntax: `-v <volume>|<volume list>|SCRATCH`

Default value: defined in `init<DBSID>.sap`

If the option `-v` is not used, BRARCHIVE will use the volumes (tapes) specified in parameter `volume_archive`.

See [-v|-volume \[Page 388\]](#).

-w|-verify

-w|-verify

Default value: No checks.

This option can be used to make sure that the backup is readable and complete. Once the archiving phase is complete, all the saved files are read from the volume (e.g. tape) in sequence (provided they were saved on a backup device), decompressed (when `compress = yes` was used), read by the check program, and compared with their originals. When archiving to disk, verification is carried out immediately after archiving the individual offline redo log files.

If the option `-s`, `-sc`, `-ss` or `-cs` is used, the file contents are compared in binary form. If the option `-sd`, `-scd`, `-ssd` or `-c ds` is used, the sizes of the archived files are determined and checked.

See [Log Supplements \[Page 442\]](#).

Please note that verification will approximately double the archiving time required.

For security reasons, we recommend using this option at least once within the volume expiration period for your archiving with `-s`, `-sc`, `-ss` or `-cs`. It enables you to detect any possible hardware problems and undertake the appropriate measures.

Command Options for BRRESTORE

You can start the SAP utility BRRESTORE without command options. If no values to the contrary have been configured in initialization profile `init<DBSID>.sap`, BRRESTORE will restore the files of all tablespaces from the last successful backup of the database.

See [Initialization Profile `init<DBSID>.sap` \[Page 306\]](#).

For more information on the SAP utility BRRESTORE, refer to the sections [Using BRRESTORE \[Page 236\]](#).

If you use BRRESTORE with command options, these override the corresponding values in the initialization profile. To use the options, you can specify either letter indicated or the complete word.

```
brrestore
[-a|-archive|-a1|-archive1 [<DBSID>,<log_no>[=<rest_dir>]|
<DBSID>,<log_no1>-<log_no2>[=<rest_dir>]| [<DBSID>,<log_no_list>]
|-a2|-archive2 [<DBSID>,<log_no>[=<rest_dir>]|
<DBSID>,<log_no1>-<log_no2>[=<rest_dir>]| [<DBSID>,<log_no_list>]
|-b|-backup <log name>|last]
|-b2|-backup2 <util_backup_id>
|-c|-confirm [force]]
|-d|-device tape|disk|pipe|tape_auto|pipe_auto|
tape_box|pipe_box|util_file]
|-e|-execute <number>]
|-f|-fillup <log_name1>[,<log_name2>,..]]
|-h|-help [version]]
|-k|-compress no|yes|hardware]
|-l|-language E|D]
|-m|-mode all|all_data|full|<tablespace name>[=<rest_dir>]|
<file_ID>[=<rest_dir>]|<file_ID1>-<file_ID2>[=<rest_dir>]|
<generic_path>[=<rest_dir>]|<object list>|archive_logs]
|-n|-number <file_pos>|init_ora|init_sap|reorg_log|det_log|
sum_log[=<rest_dir>]]
|-o|-output dist|time[,time|dist]]
|-p|-profile <profile>]
|-q|-query [check]]
|-r|-parfile <parameter_file>]
|-w|-verify]
```

[-a|-archive|a1|archive1 \[Page 415\]](#)

[-a2|-archive2 \[Page 416\]](#)

[-b|-backup \[Page 417\]](#)

[-b2|-backup2 \[Page 418\]](#)

[-c|-confirm \[Page 419\]](#)

[-d|-device \[Page 420\]](#)

[-e|-execute \[Page 421\]](#)

[-h|-help \[Page 423\]](#)

Command Options for BRRESTORE

[-k|-compress \[Page 425\]](#)

[-l|-language \[Page 426\]](#)

[-m|-mode \[Page 427\]](#)

[-n|-number \[Page 428\]](#)

[-o|-output \[Page 429\]](#)

[-p|-profile \[Page 430\]](#)

[-q|-query \[Page 431\]](#)

[-r|-parfile \[Page 432\]](#)

[-w|-verify \[Page 433\]](#)

-a|-archive|a1|archive1

Input syntax: `-a1 [<DBSID>,]<log_no>[=<rest_dir>] | [<DBSID>,]<log_no1>-<log_no2>[=<rest_dir>] | [<DBSID>,]<log_no_list>`

Default value: No restore of archived redo log files.

If you use this option, BRRESTORE checks the BRARCHIVE summary log to see which volume contains the required archived redo log files (in this case, the first copy of the offline redo log files). Mount the requested volume in the backup device and enter `cont` to confirm that you want to start the restore.

You can also use this option when you restore archived redo log files from a disk.

- **<DBSID>**: Only required if the Parallel Server option is used. In this case, the name of the ORACLE instance **<DBSID>** is included in the BRRESTORE call. Several instance identifiers may occur in one call. All the log sequence numbers listed after a **<DBSID>** refer to the ORACLE instance with this **<DBSID>**. If no instance SID appears before the first log sequence number, the log sequence numbers of the redo log files of the local instance are meant.
- **<log_no>**: Log sequence number to specify the requested first copy of the offline redo log files.
- **<log_no1>-<log_no2>**: Log sequence number interval to specify the requested first copies of the offline redo log files.
- **<rest_dir>**: The restore directory where the archived redo log files will be restored. If you do not specify a directory, the archiving directory (**<ORACLE_HOME>/saparch**) is selected.
- **<log_no_list>**: You can combine any specifications for the log sequence intervals. Separate the individual names with commas (commas only - no blanks!).

Under certain circumstances, you can restore archived redo log files from several volumes simultaneously, in parallel. See [Restoring Files \[Page 237\]](#).

-a2|-archive2

-a2|-archive2

Input syntax: -a2 [<DBSID>,]<log_no>[=<rest_dir>] | [<DBSID>,]<log_no1>-<log_no2>[=<rest_dir>] | [<DBSID>,]<log_no_list>

Default value: No restore of archived redo log files.

If you use this option, BRRESTORE checks the BRARCHIVE summary log to see which volume contains the required archived redo log files (in this case, the second copy of the offline redo log files). Mount the requested volume in the backup device and enter `cont` to confirm that you want to start the restore.

You **cannot** use this option to restore archived redo log files from a disk.

- **<DBSID>**: Only required if the Parallel Server option is used. In this case, the name of the ORACLE instance **<DBSID>** is included in the BRRESTORE call. Several instance identifiers may occur in one call. All the log sequence numbers listed after a **<DBSID>** refer to the ORACLE instance with this **<DBSID>**. If no instance SID appears before the first log sequence number, the log sequence numbers of the redo log files of the local instance are meant.
- **<log_no>**: Log sequence number to specify the requested second copy of the offline redo log files.
- **<log_no1>-<log_no2>**: Log sequence number interval to specify the requested second copies of the offline redo log files.
- **<rest_dir>, <log_no_list>**: See [-a|-archive|a1|archive1 \[Page 415\]](#).

-b|-backup

Input syntax: **-b <log name>|last**

Default value: The last successful backup of the database is used (**last**).

- **<log name>**: Enter the name of the detail log file `b<encoded timestamp>.<ext>` from a BRBACKUP backup. The requested objects will then be restored from that database backup.
- **last**: The last successful database backup is used to restore the requested objects.

-b2|-backup2

-b2|-backup2

Input syntax: **-b2** <util_backup_id>

<util_backup_id>: Backup ID of backup with an external tool

With this option you can reload backups which were executed with an external backup tool via the interface BACKINT.

See also:

[Using External Backup Programs \[Page 241\]](#)

-c|-confirm

Input syntax: `-c [force]`

Default value: Confirmation messages are issued and user entry is expected.

If you specify the option `-c`, confirmation messages that are output when the volume (e.g. tape) is mounted are suppressed. In this case, BRRESTORE assumes that the correct volume has been mounted in the backup device (e.g. tape device). All other BRRESTORE confirmation messages must be responded to.

force:

When you specify the option `-c force` all confirmation messages are suppressed. You can use this option when you regularly make database copies to have a current test system available, or when carrying out similar actions. See: [Structure Retaining Database Copy \[Page 130\]](#)



Do not use the option `-c force` when recovering a database. Follow the BRRESTORE confirmation messages in this case.

-d|-device

-d|-device

Input syntax:

-d `tape|disk|pipe|tape_auto|pipe_auto|util_file|tape_box|pipe_box|util_file`

Default value: `tape`

Depending on which backup you want to restore from, you can use this option to specify the backup media that was used.

- **disk**: Disk.
- **tape**: Tape in local tape device.
- **pipe**: Backup was performed on a tape device of a remote system.
- **tape_auto** or **pipe_auto**: Prompts for changing the tape will be suppressed. This is only useful when you use a tape device with automatic tape changing.
- **tape_box** or **pipe_box**: Jukeboxes or autoloader tape devices which can be addressed locally or remotely.
- **util_file**: Use this option when you performed the backup file by file, using external backup programs. If a parameter file is required, specify its name in profile parameter `util_par_file` or with the option `-r`.

See also:

Parameters in `init<DBSID>.sap`: [backup_dev_type \[Page 314\]](#).

-e|-execute

Input syntax: **-e** <n>

Default value: 0

See [-e|-execute \[Page 372\]](#).

When restoring, the maximum number of copy processes used corresponds to the number used for the backup. The number of parallel copy processes can therefore only be reduced by setting this option.

-f|-fillup

-f|-fillup

Input syntax: -f <log_name1>[,<log_name2>,...]

Default value: last BRRESTORE run

Using the successfully restored files entered in the BRRESTORE log <log_name> BRRESTORE determines which files still have to be restored.

See also:

[Completion of BRRESTORE Runs \[Page 238\]](#)

-h|-help

Input syntax: **-h** [**version**]

Default value: No help

See [-h|-help \[Page 374\]](#).

-incr

-incr

Input syntax:

```
-incr | -incr_only | -incr_full | -incr_only, <object_name> | <object_list>
```

When you want to restore incremental backups you can specify how many files you want to restore. You can choose either:

1. Restoration of the last full backup (level 0)
2. Restoration of the last incremental backup (level 1)

Default: **-incr**

If you only want to restore the changes to old files, enter:

```
-incr_only
```

If you only want to restore the new files, enter:

```
-incr_full
```

If you only want to restore individual files, enter:

```
-incr_only, <object_name> | <object_list>
```

See also:

Parameters in `init<DBSID>.sap: brrestore_incr`.

For more information, see [Incremental Backups \[Page 108\]](#).

-k|-compress

Input syntax: **-k no|yes|hardware**

Default value: **no** (no compression)

If the value for decompression does not agree with the compression type from the backup being used when you start BRRESTORE, an appropriate warning is issued. However, BRRESTORE always decompresses the files that were saved with software compression.

- **yes**: If you performed the backup with software compression and `compress = yes` is not set in the initialization profile, you should use this option.
- **hardware**: BRRESTORE does not distinguish internally between `no` and `hardware`.

Parameters in `init<DBSID>.sap`: [compress \[Page 320\]](#).

-l|-language

-l|-language

Input syntax: -l **E** | **D**

Default value: **E**

See [-l|-language \[Page 377\]](#).

-m|-mode

Input syntax:

```
-m all|all_data|full|<tablespace_name>[=<rest_dir>]
|<file_ID>[=<rest_dir>]|
<file_ID1>-<file_ID2> [=<rest_dir>]|
<generic_path>[=<rest_dir>]|<object list>|archive_logs]
```

Default value: **all**

Specify the objects you want to restore with BRRESTORE:

- **all**: The files in all tablespaces, but not the control files and online redo log files.
- **all_data**: The files in all tablespaces which are not pure index tablespaces.
- **full**: A complete backup, including any non-database files, directories, control files, online redo log files (complete offline backup) and offline redo log files (consistent online backup). The corresponding mirror copies of the control files and online redo log files are restored.
- **<tablespace_name>**: The files of this tablespace.
- **<file_ID>**: Data file with the specified ORACLE file ID. Control files have the file ID 0. Online redo log files have the file ID 0<n>, <n> is the redo log group number. To address all the online redo log files, use file ID 00.
- **<file_ID1>-<file_ID2>**: The files specified by the file ID interval.
- **<generic_path>**: Enter a complete path to restore the required database file, non-database file, or directory. Specify a generic path to restore all the database data files whose name starts with that path. In this case, the path must contain at least the ORACLE_HOME directory and an additional generic specification (e.g. `sapdata<n>`) in the path.
If the generic path is a directory, you can also restore one or more file(s) from this directory by additionally specifying one or more file names:
`<directory_name>:<file_name1>[:<file_name2>...]`
- **<rest_dir>**: The restore directory to which the requested files will be restored. If no directory is specified, the original directory from the backup log is selected. See also [new db home \[Page 335\]](#)
- **<object list>**: You can specify a list of tablespaces or files, or combine the key words `all` and `full` with an object list. When possible, always restore database files and non-database files in separate runs. The individual objects are separated by commas (commas only, no blanks!).
- **archive_logs**: The offline redo log files of a consistent BRBACKUP online backup (`backup_type=online_cons`).

Parameters in `init<DBSID>.sap`: [backup_mode \[Page 316\]](#).

-n|-number

-n|-number

Input syntax:

```
-n <file_Pos>|init_ora|init_sap|reorg_log|det_log| sum_log[=<rest_dir>]
```

Default value: No specification for position of the file on the volume (tape).

With this option, you can restore a file directly from the volume without having to specify the name of the backup log or the log sequence number.

- **<file_Pos>**: Specify the position of the file on the tape.
- **init_ora**: The profiles `init<DBSID>.ora` and `init<DBSID>.dba` are restored (OPS: the `init.ora/init.dba` profiles of all database instances are restored). These files have the position 2.
- **init_sap**: The initialization profile `init<DBSID>.sap` is restored. This file is in position 3.
- **reorg_log**: The main log `reorg<DBSID>.log` and the structure log `struct<DBSID>.log` are restored. These files are in the third-to-last position. (the script `create_raw_links` is also restored, if it exists).
- **det_log**: The detail BRBACKUP/BRARCHIVE log is restored. This file is in the second-to-last position.
- **sum_log**: The summary BRBACKUP/BRARCHIVE log is restored. This file is in the last position. (OPS: the collective BRARCHIVE logs of all database instances are restored).
- **<rest_dir>**: Specifies the restore directory into which the requested files will be restored. If no directory is specified, the files will be restored to the current directory where BRRESTORE was started.

-o|-output

Input syntax: `-o dist|time[,time|dist]`

Default value: The BRRESTORE detail log is written normally. See [Detail Log \[Page 453\]](#).

See [-o|-output \[Page 381\]](#).

-p|-profile

-p|-profile

Input syntax: -p <profile>

Default value: `init<DBSID>.sap`

See [-p|-profile \[Page 382\]](#).

-q|-query

Input syntax: **-q** [**check**]

Default value: The restore process is started.

When you select the **-q** option, you find out which volumes (tapes) must be mounted for the restore process and which additional sources the program needs. In this case, restore is not started.

- **check**: You can check whether the proper volumes have really been mounted in the backup devices. The restore is not started.

-r|-parfile

-r|-parfile

Input syntax: `-r <parameter_file>`

Default value: No parameter file

See [-r|-parfile \[Page 384\]](#).

-w|-verify

Input syntax: `-w [use_dbv]`

Default value: No check



Restoring the last check, followed by a check of the ORACLE block structure using DB_VERIFY:

```
brrestore -b -w use_dbv
```

Restoring the archived offline redo log files nos. 112- 250 and checking them for readability:

```
brrestore -a 112-250 -w
```

See: [Backup Verifications \[Page 155\]](#)

BRBACKUP, BRARCHIVE, and BRRESTORE Logs

BRBACKUP, BRARCHIVE, and BRRESTORE Logs

Logs are created when calling the SAP programs BRBACKUP, BRARCHIVE or BRRESTORE. These are a source of information about performed operations and are particularly useful for error analysis.

They are also a prerequisite for the smooth execution of the operations that you can perform using the SAP programs (for example, for the restore and recovery process).



You should always include logs in a problem message to provide a basis for error analysis.

[Log Types \[Page 435\]](#)

[BRBACKUP Logs \[Page 438\]](#)

[BRARCHIVE Logs \[Page 445\]](#)

[BRRESTORE Logs \[Page 451\]](#)

[Sample Logs \[Page 454\]](#)

Log Types

The following types of logs are written:

- [File System Logs \[Page 436\]](#)
- [Database Logs \[Page 437\]](#) in tables SDBAH and SDBAD

File system logs

File system logs

BRBACKUP, BRARCHIVE, BRRESTORE write the following file system logs:

- A detail log that is generated anew each time a process is started.
- A summary log that is appended after every run of the utilities (only BRBACKUP, BRARCHIVE).

Displaying the File System Logs

To display the logs, you can use *SAPDBA Show/Cleanup* → *Show log files/profiles*.

The logs are contained in the following UNIX directories:

- `<ORACLE_HOME>/sapbackup` - for BRBACKUP, BRRESTORE
- `<ORACLE_HOME>/saparch` - for BRARCHIVE

Database Logs

The database logs of the SAP utilities BRBACKUP and BRARCHIVE are stored in tables SDBAH and SDBAD. The SAP utility SAPDBA also uses these tables. SAPDBA saves information from the main log `reorg<DBSID>.log` in these database tables (see [Main Log \[Page 300\]](#) **Error! No bookmark name given.**).



You should never manually change tables SDBAH and SDBAD. Only the appropriate SAP utilities and the R/3 System CCMS may append or change these tables.

As a result, this manual will not describe these tables in detail, but instead only list an overview of which information BRBACKUP/BRARCHIVE stores in which tables. This information is by no means complete.

SDBAH

This table contains information that refers to the backup as a whole:

- Starting time of the backup
- End time of the backup
- BRBACKUP/BRARCHIVE return code
- BRBACKUP/BRARCHIVE action ID (encoded timestamp of the file system log names)
- BRBACKUP/BRARCHIVE function ID (extension of the file system log names)

SDBAD

This table contains information that refers to the backup of one file:

- File name
- ORACLE file ID or log group number
- End time of the backup of the file
- Name of the volume where the file was saved
- Position of the file on the volume
- Backup ID of the external backup program
- Compression rate of the software compression

In addition, table SDBAD contains internal BRBACKUP information about compression rates and backup durations for the individual database files.

BRBACKUP Logs

BRBACKUP Logs

[Names of the Detail Logs \[Page 439\]](#)

[Detail Log \[Page 440\]](#)

[Log Supplements \[Page 442\]](#)

[Summary Log \[Page 444\]](#)

Names of the Detail Logs

Every detail log contains a name with the following format: **b**<encoded timestamp>.<ext>

The first characters indicate the encoded time the archiving was performed (action ID). The extension (function ID) indicates the type of archiving.

Possible values for **e**:

- **a**: The entire database was backed up (`backup_mode = all|all_data`).
- **p**: One or more tablespaces or files were backed up.

Possible values for **x**:

- **n**: The backup was performed online (`backup_type = online|online_cons|online_split`).
- **f**: The backup was performed offline (`backup_type = offline|offline_force|offline_standby|offline_split`).

Possible values for **t** (specification of backup devices):

- **t**: Tape device (`backup_device_type = tape|tape_auto|tape_box`).
- **d**: Disk (`backup_device_type = disk|disk_copy|disk_standby`).
- **p**: Tape device on a remote system (`backup_device_type = pipe|pipe_auto|pipe_box`).
- **f**: External backup program used; backup performed file by file (`backup_device_type = util_file|util_file_online`).

Other function IDs:

- **tib**: One of the BRBACKUP options `-i`, `-i force` or `-i show` was used to initialize a volume or display the information in the label.
- **qub**: One of the BRBACKUP options `-q` or `-q check` was used to display which volumes are to be used for the backup, or to make sure that these volumes were actually mounted; no backup was started.
- **cmb**: The BRBACKUP option `-k only` was used to only perform a software compression, but no backup was started. This can be used to determine the current compression rate of all files.

Detail Log

Detail Log

This section describes the information contained in a detail log (a<encoded timestamp>.<ext>, see [Names of the Detail Logs \[Page 439\]](#) **Error! No bookmark name given.**).

The detail log file contains information about the status of the ORACLE database at the time of the backup, and about the actions that were performed in the course of the backup.

- Displays the relevant parameters of initialization profile `init<DBSID>.sap` that were set during the BRBACKUP run. Information about ORACLE archiving before the BRBACKUP run.
- Information on ORACLE archiving before starting BRBACKUP: Database mode (ARCHIVELOG, NOARCHIVELOG), status of the archiving process (enabled, disabled), archiving directory, oldest log sequence number of the online redo log files, next redo log files for archiving by ORACLE, log sequence number of the current online redo log files, start SCN (system change number) of the current online redo log files
- Listing of data files of the tablespaces, the redo log files, and the control files:
 - Tablespaces and data files:
Tablespace name, ORACLE tablespace status ('*' means: data tablespace), ORACLE file status, file name, file size, ORACLE file ID, disk volume ID, link directory at subdirectory level (usually for subdirectory <tablespace name>_<file number>) or key word `NOLINK`, when no soft links were defined, file type `FILE | RAW`
 - Online redo log files:
File name, file size, redo log group number, disk volume ID, ORACLE file status, link directory or key word `NOLINK` when no soft links were defined, file type `FILE | RAW`
 - Control files:
File name, file size, 0 (default for file ID), disk volume ID, link directory or key word `NOLINK` when no soft links were defined, file type `FILE | RAW`
- Listing of non-database files, if non-database files were saved:
Disk volume ID, file size, file name
- Listing of directories and their contents, if directories were saved using this option:
Directory name, disk volume ID, file category (`file` - file, `link` - soft link, `pipe` - named pipe, `dir` - directory, `spec` - special file), file size, file name
- The backup flow.

Backup Flow

The log will contain additional information when you start BRBACKUP with the option `-o dist|time` or/and `-w`. See [Log Supplements \[Page 442\]](#), [-o|-output \[Page 381\]](#) and [-w|-verify \[Page 389\]](#).

This information indicates which database file, non-database file, or directory, or which profile and log were saved where.

- **#FILE** The data file name as it appears in the control file. If `compress = only` is used, the compression rate is also displayed.
- **#NDBF** Name of a non-database file.
- **#DIR** Name of a directory.
- **#ARCHIVE** Name of the offline redo log files (only for `backup_type=online_cons`)
- **#INLOG** Name of the relevant profile/log (only when archiving with `BACKINT`).
- **#SAVED** This entry differs depending on the type of backup:
 - Backup on tape
#SAVED file name, name of tape, position of the file on the tape, compression rate and the size of the compressed file and size of the compressed file (if compressed)
 - Backup on disk
#SAVED file name, symbolic volume name and file position (only important for `BRESTORE`), compression rate and the size of the compressed file (only if compressed)
 - Backup using an external backup program
#SAVED Backup ID returned by the external backup program

This is followed by an archive log list that is created (and possibly updated) after the backup, as well as several closing comments.



See: [Detail Log of an Offline Backup \[Page 457\]](#)
[Volume Initialization \[Page 455\]](#)

Log Supplements

Log Supplements

Using the option `-o dist|time[,time|dist]` and/or the option `-w|-verify` causes the detail log to be supplemented (the effects are similar for all the SAP utilities). See [-o|output \[Page 381\]](#) or [-w|-verify \[Page 389\]](#) (for BRBACKUP), [-w|-verify \[Page 412\]](#) (for BRARCHIVE).

Log Supplements When Using: -o dist

Before the backup flow is logged, the SAP utility records information about the distribution of the files for backup among the volumes (tapes or disks) used. The following information is listed:

```
position size rate compressed duration speed name
```

- **position** Position of the file on the volume.
- **size** Size of the file (in bytes).
- **rate** Compression rate when hardware or software compression is used. The compression rate is specified as: `<value>:1,value=<size (file size before compression)>/<compressed (file size after compression)>`.

If no compression rate is available yet for this file, the default value 3:1 is taken as the compression rate for most files (indicated by an asterisk "*" after the compression rate). Exceptions: the files of tablespaces PSAPSOURCED, PSAPLOADD, PSAPPOOLD, PSAPDOCUD, PSAPCLUD and the online redo log files; the default value 1:1 is used for the compression rate.

- **compressed** Size of the file (in bytes) after their compression.
- **duration** Indicates how long the last backup of this file took (`<minutes>:<seconds>`). This enables you to approximately estimate the duration of the backup.
- **speed** Throughput of the backup (calculation: `speed=<size>/<duration>` in MB/h)
- **name** File name

This information is listed for each volume (tape or disk). The last line (**total**) shows the totals or averages of the individual columns. An asterisk "*" in column `duration` indicates that the SAP utility optimized distribution of the files among the backup devices for time (only relevant for parallel backups). The goal of this is to balance the load of all the BRBACKUP/BRARCHIVE backup devices. When optimization by time was not possible BRBACKUP/BRARCHIVE will attempt to equally distribute the data volume.

Log Supplements When Using: -o time

During the backup, BRBACKUP issues a timestamp (BR284I Time stamp YYYY-MM-DD hh.mm.ss, elapsed time: `<minutes>:<seconds>`) after every important event, e.g. more often than normal. This information can be used, for example, to track how much time was actually required for a backup (`<timestamp after operation complete> minus <timestamp at operation start>`); in the case of a parallel backup, these two timestamps must not appear one after another.

After the log entries on the backup flow, a list appears with information about the saved files.

```
position duration size speed compressed rate name
```

This information was described above with the use of option `-o dist`. Of course, the values in column `duration` have now been updated. If software compression is used, the current compression rate can also be determined.

Log Supplements When Using: `-w|-verify`

- Indicates that verification is active and that twice the backup time is required as a result.
- Information about which volume was used for the verification run, which files will be restored where, and the result of the check (e.g. BR363I Verification of backup of <file name> successful).

Please note that after successful backup to tape, the files are actually restored to disk, the check mechanisms are activated, and the files are deleted. This approximately doubles the backup time required.



See: [Supplements to a Detail Log \[Page 464\]](#)

Summary Log

Summary Log

You can display a brief entry for each backup in the summary log `back<DBSID>.log`. The entries in the file provide the following information about each backup using BRBACKUP:

- Action ID (encoded timestamp of the log name).
- Function ID (extension of the log name)
 - Timestamp (date, time) specifying the start of the backup
 - Timestamp (date, time) specifying the end of the backup
 - Return code **Error! No bookmark name given.**
 - Total compression rate (appears only when compression was used)
 - Number of files backed up:
 - `all[*]`: all database files
 - `all_data[*]`: all pure data tablespaces
 - `partial[*]`: partial backup
 - `non_db`: all non-database files (e.g. `sap_dir`, `ora_dir`)
- Value of the parameter [backup_type \[Page 319\]](#)
- Value of the parameter [backup_dev_type \[Page 314\]](#)
- Flag for the BRBACKUP command options:
a(-archive), b(-backup), f(-fillup), k(-compress), w(-verify)



See: [Summary BRBACKUP Log \[Page 476\]](#)

BRARCHIVE Logs

[Names of the Detail Logs \[Page 446\]](#)

[Detail Log \[Page 447\]](#)

[Summary Log \[Page 449\]](#)

Names of the Detail Logs

Names of the Detail Logs

Every detail log contains a name with the following format: `a<encoded timestamp>.<ext>`

The first characters indicate the encoded time the archiving was performed (action ID). The extension (function ID) indicates the type of archiving.

Possible function IDs:

- **sve**: Offline redo log files archived for the first time (option `-s`).
- **cpy**: Offline redo log files archived a second time (option `-sc`).
- **svd**: Offline redo log files were archived and then deleted from the archiving directory (option `-sd`).
- **cpd**: Offline redo log files were archived a second time and then deleted from the archiving directory (option `-scd`).
- **dsv**: Archived redo log files were deleted (option `-ds`).
- **dcp**: Offline redo log files that had been archived twice were deleted (option `-dc`).
- **ssv**: Offline redo log files were archived twice. Archiving was performed on two tape devices in parallel (option `-ss`).
- **ssd**: Offline redo log files were archived twice and then deleted. Archiving was performed on two tape devices in parallel (option `-ssd`).
- **cps**: Archived offline redo log files were archived for a second time. Then continues with the archiving of the files newly included in the archiving directory (option `-cs`).
- **cds**: Archived offline redo log files were archived for a second time and deleted. Then continues with the archiving of the files newly included in the archiving directory (option `-c ds`).
- **tia**: The BRARCHIVE option `-i`, `-i force` or `-i show` was used to initialize a volume or display the information in the label.
- **qua**: The BRARCHIVE option `-q` or `-q check` was used to display which volumes are to be used for archiving or make sure that those volumes were actually mounted; no archiving was started.
- **cma**: The BRARCHIVE option `-k only` was used to perform a software compression, but no archiving was started. This can be used to determine the current compression rate of all files.
- **fst**: The BRARCHIVE option `-f stop` was used to stop a BRARCHIVE run.

Detail Log

This section describes the information contained in a detail log (a<encoded timestamp>.<ext>, see [Names of the Detail Logs \[Page 446\]](#) **Error! No bookmark name given.**).

The detail log file contains information about the actions that were performed in the course of the archiving.

- Displays the relevant parameters of initialization profile `init<DBSID>.sap` that were set during the BRARCHIVE run.
- The archiving flow.

Archiving Flow

The log will contain additional information when you start BRARCHIVE with the option `-o dist|time or/and -w|-verify`. See [Log Supplements \[Page 442\]](#), [-o|-output \[Page 403\]](#) and [-w|-verify \[Page 412\]](#).

This information indicates which file was saved where.

- **#ARCHIVE:** Name of the relevant offline redo log files.
- **#INLOG:** Name of the relevant profile/log (only when archiving with BACKINT).
- **#SAVED:** This entry varies depending on the type of archiving:
 - Archiving on Tape
#SAVED file name, name of tape, position of the file on the tape, compression rate and the size of the compressed file (if compressed)
 - Archiving on disk
#SAVED file name, symbolic volume name and file position (only important for BRRESTORE), compression rate and the size of the compressed file (only if compressed)
 - Archiving using an external backup program
#SAVED Backup ID returned by the external backup program

This is followed by several closing comments (e.g. the total number of processed offline redo log files).



See: [Detail BRARCHIVE Log \[Page 478\]](#)

Log Supplements

Using the option `-o dist|time [,time|dist]` and/or the option `-w` causes the detail log to be supplemented.

Detail Log

When you select `-o dist` and no compression rate is available yet, the compression rate `1:1` is selected for the offline redo log files (column `rate` in the display). Note that BRARCHIVE will only carry out archiving in parallel when started with one of the options `-ss` or `-ssd`.

See: [Log Supplements \[Page 442\]](#)

Summary Log

You can display a brief entry for every archived offline redo log file in the summary log `arch<DBSID>.log`. The entries in the file provide the following information about each archiving run with BRARCHIVE:

- **#ARCHIVE** Log sequence number, name of the offline redo log files, creation time, file size, start SCN (system change number) of the offline redo log file, thread number
- **#SAVED** This entry depends on the type of archiving for the offline redo log files:
 - Archiving on tape
 - Action ID (encoded timestamp of the log name).
 - Function ID (extension of the log name)
 - Name of tape/position of the file on tape (starts with '#' for `tape_copy_cmd=cpio` or with '\$' for `tape_copy_cmd=dd`)
 - Timestamp (date, time) specifying the end of the archiving process
 - Compression rate and the size of the compressed file (if it was compressed)
 - Archiving using an external backup program
 - Action ID (encoded timestamp of the log name).
 - Function ID (extension of the log name)
 - Backup ID returned by the external backup program (if the `util_file` option was used, an asterisk '*' or '@' appears before the backup ID)
 - Timestamp (date, time) specifying the end of the backup
- **#COPIED** Information like under #SAVED when the offline redo log files were archived a second time.
- **#DELETED** Information (action ID, function ID, timestamp) about when the file was deleted.
 - ****** Information on the status of a BRARCHIVE run: ORACLE_SID, device type (values like in `backup_dev_type` or `null` when offline redo log files were only deleted), action ID, function ID, timestamps (date, time) specifying the start and the end of the BRARCHIVE run, return code **Error! No bookmark name given.**, the total compression rate (if compressed), flag for the BRARCHIVE command options.
`a(-archive), b(-backup), f(-fill), k(-compress), w(-verify), m(-modify).`
- **#DISK** This information appears if you used the BRARCHIVE feature for archiving to disk:
 - Log sequence number, name of the offline redo log file, creation time, file size, start SCN (system change number) of the offline redo log file, thread number.
- **#DISKSAV** Archiving to disk: action ID, function ID, name of the file on the disk, timestamp, compression rate and the size of the compressed file (if it was compressed).
- **#DISKDEL** Archiving to disk: information (action ID, function ID, timestamp), if the file was deleted.
 - **##** Comments entered by the user.

Summary Log



See: [Summary BRARCHIVE Log \[Page 483\]](#)

BRRESTORE Logs

[Names of the Detail Logs \[Page 452\]](#)

[Detail Log \[Page 453\]](#)

Names of the Detail Logs

Names of the Detail Logs

Every detail log contains a name with the following format: `r<encoded timestamp>.<ext>`

The first characters indicate the encoded time the restore was performed (action ID). The extension (function ID) indicates the type of the process.

Possible function IDs:

- **rsb**: Restore from a BRBACKUP backup (option `-b` | `-backup` | `-b2`).
- **rsa**: Restore the redo log files specified by the log sequence numbers (option `-a` | `-archive` | `-a1` | `-a2`).
- **rsf**: Restore a file characterized by its position on the volume (option `-n` | `-number`).

Detail Log

The detail log file contains information about the actions that were performed during the restore process.

- Displays the relevant parameters of initialization profile `init<DBSID>.sap` that were set during the BRRESTORE run.
- Restore flow. You can precisely monitor which backup was used to restore the files, which volumes were mounted, etc.
- **#FILE**: Indicates the full path and the name of the restored file.
- **#NDBF**: Indicates the full path and the name of the restored non-database file.
- **#DIR**: Indicates the full name of the restored directory.
- **#ARCHIVE**: Indicates the full path and the name of the restored archived redo log file.
- **#RESTORED**: This item varies depending on which backup medium was used to restore the file:
 - Restore from tape
#RESTORED Name of the file on tape, file name/position on the tape (this information was used to save the file on tape; the new file name does not necessarily agree with this name.).
 - Restore from disk
#RESTORED Complete name of the file on disk, as it was saved, and the symbolic volume name and file position.
 - Restore using an external backup program
#RESTORED Backup ID returned by the external backup program when the file was backed up.



See: [BRRESTORE Log: Restoring a Complete Backup \[Page 487\]](#)

Log Supplements

Using the option `-o dist|time` to start BRRESTORE causes the detail log to be supplemented. The information about the distribution of files on the volumes (if you use `-o dist`) refers to the time when the files were saved. See [Log Supplements \[Page 442\]](#) and [-o|-output \[Page 429\]](#). The details of the backup times (`-o time`) refer to the recovery process.

Sample Logs

Sample Logs

This section contains several sample logs for operations performed with BRBACKUP, BRARCHIVE, or BRRESTORE.

[Volume Initialization \[Page 455\]](#)

[Offline Backup \[Page 457\]](#)

[Supplements to a Detail Log \[Page 464\]](#)

[Summary BRBACKUP Log \[Page 476\]](#)

[Detail BRARCHIVE Log \[Page 478\]](#)

[Summary BRARCHIVE Log \[Page 483\]](#)

[Restoring a Complete Backup \[Page 487\]](#)

Volume Initialization

The following log shows the flow of the initialization of two tapes with labels C11B05 and C11B06. The initialization was performed on two tape devices in parallel, using the option `-i force` in unattended mode.

Call: `brbackup -i force -v C11B05,C11B06 -c`

```
BR051I BRBACKUP 4.0A
BR055I Start of database backup: bcvojsmi tib 1997-09-10 18.26.24

BR101I Parameters

oracle_sid          C11
oracle_home         /oracle/C11
oracle_profile      /oracle/C11/dbs/initC11.ora
sap_profile         /oracle/C11/dbs/initC11.sap
backup_dev_type     tape
tape_copy_cmd       cpio
cpio_flags          -ovB
cpio_in_flags       -iuvB
cpio_disk_flags     -pdcu
dd_flags            obs=16k bs=16k
dd_in_flags         ibs=16k bs=16k
rewind              mt -t $ rew
rewind_offline     mt -t $ offl
tape_address        /dev/rmt/0mn,/dev/rmt/1mn
tape_address_rew    /dev/rmt/0m,/dev/rmt/1m
volume_backup       C11B05,C11B06
expir_period        30
tape_use_count      100
exec_parallel       0

BR205I Backup volume initialization
BR058I BRBACKUP action ID: bcvojsmi
BR059I BRBACKUP function ID: tib
BR102I Following backup devices will be used: /dev/rmt/0mn,/dev/rmt/1mn
BR144I Following backup volume names will be used: C11B05,C11B06
BR129W Force option set - no label checking during initialization
BR126I Unattended mode active - no confirmation required

BR207I Volumes in devices: /dev/rmt/0mn,/dev/rmt/1mn
will be initialized with names: C11B05,C11B06

BR280I Time stamp 1997-09-10 18.26.24
BR226I Rewinding tape volume in device /dev/rmt/0m...

BR202I Saving /oracle/C11/sapbackup/.tape.hdr0
BR203I to /dev/rmt/0mn...

BR209I Volume in device /dev/rmt/0mn has name C11B05

BR280I Time stamp 1997-09-10 18.26.41
BR226I Rewinding tape volume in device /dev/rmt/0m...

BR280I Time stamp 1997-09-10 18.26.42
BR226I Rewinding tape volume in device /dev/rmt/1m...
```

Volume Initialization

```
BR202I Saving /oracle/C11/sapbackup/.tape.hdr0
BR203I to /dev/rmt/1mn...

BR209I Volume in device /dev/rmt/1mn has name C11B06

BR280I Time stamp 1997-09-10 18.26.58
BR226I Rewinding tape volume in device /dev/rmt/1m...

BR206I End of backup volume initialization.
BR053I BRBACKUP terminated successfully with warnings
```


Detail Log of an Offline Backup

The following excerpts come from the log of an offline backup performed in parallel on two tape devices in unattended mode.

Call: **brbackup -t offline -c**

```
BR051I BRBACKUP 4.0A
BR055I Start of database backup: bcvojtgt aft 1997-09-10 18.35.15
```

BR101I Parameters

```
oracle_sid          C11
oracle_home         /oracle/C11
oracle_profile      /oracle/C11/dbs/initC11.ora
sap_profile         /oracle/C11/dbs/initC11.sap
backup_mode        ALL
backup_type        offline
backup_dev_type    tape
compress           no
tape_copy_cmd      cpio
cpio_flags         -ovB
cpio_in_flags      -iuvB
cpio_disk_flags    -pdcu
dd_flags           obs=16k bs=16k
dd_in_flags        ibs=16k bs=16k
rewind             mt -t $ rew
rewind_offline     mt -t $ offl
tape_size          1200.000 MB
tape_address       /dev/rmt/0mn,/dev/rmt/1mn
tape_address_rew   /dev/rmt/0m,/dev/rmt/1m
volume_backup      C11B01,C11B02,C11B03,C11B04,C11B05,C11B06,C11B07,C11B08,C11B09,C11B10,
                  C11B11,C11B12,C11B13,C11B14,C11B15,C11B16,C11B17,C11B18,C11B19,C11B20,
                  C11B21,C11B22,C11B23,C11B24,C11B25,C11B26,C11B27,C11B28,C11B29,C11B30
expir_period       30
tape_use_count     100
exec_parallel      0
```

BR116I ARCHIVE LOG LIST before backup for database instance C11

```
Database log mode          Archive Mode
Automatic archival        Enabled
Archive destination       /oracle/C11/saparch/C11arch
Oldest online log sequence 232
Next log sequence to archive 235
Current log sequence      235      SCN: 9364
Database block size      8192
```

BR118I Tablespaces and data files

```
PSAPRAWD ONLINE ONLINE /oracle/C11/sapraw/rawd_1 10493952 8
1073741829 /dev/vg00/rlvol5 RAW
PSAPROLL ONLINE* ONLINE /oracle/C11/sapdata2/roll_1/roll.data1
20979712 2 1073807361 NOLINK FILE
PSAPSTABD ONLINE* ONLINE /oracle/C11/sapdata3/stabd_1/stabd.data1
10493952 4 1073807361 NOLINK FILE
```

Detail Log of an Offline Backup

```

PSAPSTABI ONLINE ONLINE /oracle/C11/sapdata4/stabi_1/stabi.data1
5251072 5 1073807361 NOLINK FILE
PSAPTEMP ONLINE ONLINE /oracle/C11/sapdata2/temp_1/temp.data1
5251072 3 1073807361 NOLINK FILE
PSAPTESTD ONLINE* ONLINE /oracle/C11/sapdata3/testd_1/testd.data1
4202496 6 1073807361 NOLINK FILE
PSAPTESTI ONLINE ONLINE /oracle/C11/sapdata4/testi_1/testi.data1
2105344 7 1073807361 NOLINK FILE
SYSTEM ONLINE* SYSTEM /oracle/C11/sapdata1/system_1/system.data1
52436992 1 1073807361 NOLINK FILE

```

BR119I Redo log files

```

/oracle/C11/origlogA/log_g11m1.dbf 205824 1 1073807361 INUSE NOLINK
FILE
/oracle/C11/mirrlogA/log_g11m2.dbf 205824 1 1073807361 INUSE NOLINK
FILE
/oracle/C11/origlogB/log_g12m1.dbf 205824 2 1073807361 INUSE NOLINK
FILE
/oracle/C11/mirrlogB/log_g12m2.dbf 205824 2 1073807361 INUSE NOLINK
FILE
/oracle/C11/origlogA/log_g13m1.dbf 205824 3 1073807361 INUSE NOLINK
FILE
/oracle/C11/mirrlogA/log_g13m2.dbf 205824 3 1073807361 INUSE NOLINK
FILE
/oracle/C11/origlogB/log_g14m1.dbf 205824 4 1073807361 INUSE NOLINK
FILE
/oracle/C11/mirrlogB/log_g14m2.dbf 205824 4 1073807361 INUSE NOLINK
FILE

```

BR120I Control files

```

/oracle/C11/sapdata1/cntrl/cntrlC11.dbf 336896 0 1073807361 NOLINK
FILE
/oracle/C11/sapdata2/cntrl/cntrlC11.dbf 336896 0 1073807361 NOLINK
FILE
/oracle/C11/sapdata3/cntrl/cntrlC11.dbf 336896 0 1073807361 NOLINK
FILE
/oracle/C11/sapdata4/cntrl/cntrlC11.dbf 336896 0 1073807361 NOLINK
FILE

```

```

BR280I Time stamp 1997-09-10 18.35.19
BR057I Backup of database: C11
BR058I BRBACKUP action ID: bcvojtgt
BR059I BRBACKUP function ID: aft
BR060I Tablespace for backup: ALL
BR077I Database files for backup:
/oracle/C11/origlogA/log_g11m1.dbf
/oracle/C11/origlogB/log_g12m1.dbf
/oracle/C11/origlogA/log_g13m1.dbf
/oracle/C11/origlogB/log_g14m1.dbf
/oracle/C11/sapdata1/cntrl/cntrlC11.dbf
BR061I 13 files found for backup, total size 107.169 MB.
BR143I Backup type: offline
BR112I Files will not be compressed.
BR130I Backup device type: tape
BR102I Following backup devices will be used:
/dev/rmt/0mn,/dev/rmt/1mn
BR103I Following backup volumes will be used:

```

Detail Log of an Offline Backup

```
C11B05,C11B06
BR140I Number of parallel copy processes: 2.
BR126I Unattended mode active - no confirmation required.

BR307I Shutting down the database instance C11...
BR308I Shutdown of database instance C11 successful.

BR280I Time stamp 1997-09-10 18.35.29
BR226I Rewinding tape volume in device /dev/rmt/0m...

BR280I Time stamp 1997-09-10 18.35.29
BR226I Rewinding tape volume in device /dev/rmt/1m...

BR351I Restoring /oracle/C11/sapbackup/brlab0/.tape.hdr0
BR355I from /dev/rmt/0mn...

BR351I Restoring /oracle/C11/sapbackup/brlab1/.tape.hdr0
BR355I from /dev/rmt/1mn...

BR280I Time stamp 1997-09-10 18.35.32
BR226I Rewinding tape volume in device /dev/rmt/0m...

BR280I Time stamp 1997-09-10 18.35.32
BR226I Rewinding tape volume in device /dev/rmt/1m...

BR208I Volume with name:
C11B05|C11B06 required in device /dev/rmt/0mn

BR241I Checking label on volume in device /dev/rmt/0mn

BR208I Volume with name C11B06 required in device /dev/rmt/1mn

BR241I Checking label on volume in device /dev/rmt/1mn

BR202I Saving /oracle/C11/sapbackup/brlab0/.tape.hdr0
BR203I to /dev/rmt/0mn...

BR202I Saving /oracle/C11/sapbackup/brlab1/.tape.hdr0
BR203I to /dev/rmt/1mn...

BR209I Volume in device /dev/rmt/0mn has name C11B05

BR209I Volume in device /dev/rmt/1mn has name C11B06

BR202I Saving init_ora
BR203I to /dev/rmt/0mn...

BR202I Saving init_ora
BR203I to /dev/rmt/1mn...

BR202I Saving /oracle/C11/dbs/initC11.sap
BR203I to /dev/rmt/0mn...

BR202I Saving /oracle/C11/dbs/initC11.sap
BR203I to /dev/rmt/1mn...

BR280I Time stamp 1997-09-10 18.36.20

BR202I Saving /oracle/C11/sapdata1/system_1/system.data1
```

Detail Log of an Offline Backup

```
BR203I to /dev/rmt/0mn...

BR202I Saving /oracle/C11/sapraw/rawd_1
BR203I to /dev/rmt/1mn...

#FILE.... /oracle/C11/sapraw/rawd_1
#SAVED... rawd_1 C11B06/4

BR280I Time stamp 1997-09-10 18.37.25
BR063I 1 of 13 files processed - 10.008 MB of 107.169 MB done.
BR204I Percentage done: 9.34%, estimated end time: 18:47
BR001I *****

BR202I Saving /oracle/C11/sapdata3/stabd_1/stabd.data1
BR203I to /dev/rmt/1mn...

#FILE.... /oracle/C11/sapdata3/stabd_1/stabd.data1
#SAVED... stabd.data1 C11B06/5

BR280I Time stamp 1997-09-10 18.38.50
BR063I 2 of 13 files processed - 20.016 MB of 107.169 MB done.
BR204I Percentage done: 18.68%, estimated end time: 18:49
BR001I *****

BR202I Saving /oracle/C11/sapdata4/stabi_1/stabi.data1
BR203I to /dev/rmt/1mn...

#FILE.... /oracle/C11/sapdata4/stabi_1/stabi.data1
#SAVED... stabi.data1 C11B06/6

BR280I Time stamp 1997-09-10 18.39.46
BR063I 3 of 13 files processed - 25.023 MB of 107.169 MB done.
BR204I Percentage done: 23.35%, estimated end time: 18:51
BR001I *****

BR202I Saving /oracle/C11/sapdata2/temp_1/temp.data1
BR203I to /dev/rmt/1mn...

#FILE.... /oracle/C11/sapdata1/system_1/system.data1
#SAVED... system.data1 C11B05/4

BR280I Time stamp 1997-09-10 18.39.52
BR063I 4 of 13 files processed - 75.031 MB of 107.169 MB done.
BR204I Percentage done: 70.01%, estimated end time: 18:41
BR001I *****

BR202I Saving /oracle/C11/sapdata2/roll_1/roll.data1
BR203I to /dev/rmt/0mn...

#FILE.... /oracle/C11/sapdata2/temp_1/temp.data1
#SAVED... temp.data1 C11B06/7

BR280I Time stamp 1997-09-10 18.40.09
BR063I 5 of 13 files processed - 80.039 MB of 107.169 MB done.
BR204I Percentage done: 74.68%, estimated end time: 18:41
BR001I *****

BR202I Saving /oracle/C11/sapdata3/testd_1/testd.data1
BR203I to /dev/rmt/1mn...
```

Detail Log of an Offline Backup

```
#FILE.... /oracle/C11/sapdata3/testd_1/testd.data1
#SAVED.... testd.data1 C11B06/8

BR280I Time stamp 1997-09-10 18.40.44
BR063I 6 of 13 files processed - 84.047 MB of 107.169 MB done.
BR204I Percentage done: 78.42%, estimated end time: 18:41
BR001I *****

BR202I Saving /oracle/C11/sapdata4/testi_1/testi.data1
BR203I to /dev/rmt/lmn...

#FILE.... /oracle/C11/sapdata4/testi_1/testi.data1
#SAVED.... testi.data1 C11B06/9

BR280I Time stamp 1997-09-10 18.41.14
BR063I 7 of 13 files processed - 86.055 MB of 107.169 MB done.
BR204I Percentage done: 80.30%, estimated end time: 18:42
BR001I *****

BR202I Saving /oracle/C11/sapdata1/cntrl/cntrlC11.dbf
BR203I to /dev/rmt/lmn...

#FILE.... /oracle/C11/sapdata2/roll_1/roll.data1
#SAVED.... roll.data1 C11B05/5

BR280I Time stamp 1997-09-10 18.41.22
BR063I 8 of 13 files processed - 106.062 MB of 107.169 MB done.
BR204I Percentage done: 98.97%, estimated end time: 18:41
BR001I *****

#FILE.... /oracle/C11/sapdata1/cntrl/cntrlC11.dbf
#SAVED.... cntrlC11.dbf C11B06/10

BR280I Time stamp 1997-09-10 18.41.31
BR063I 9 of 13 files processed - 106.384 MB of 107.169 MB done.
BR204I Percentage done: 99.27%, estimated end time: 18:41
BR001I *****

BR202I Saving /oracle/C11/origlogA/log_g11m1.dbf
BR203I to /dev/rmt/lmn...

#FILE.... /oracle/C11/origlogA/log_g11m1.dbf
#SAVED.... log_g11m1.dbf C11B06/11

BR280I Time stamp 1997-09-10 18.41.47
BR063I 10 of 13 files processed - 106.580 MB of 107.169 MB done.
BR204I Percentage done: 99.45%, estimated end time: 18:41
BR001I *****

BR202I Saving /oracle/C11/origlogB/log_g12m1.dbf
BR203I to /dev/rmt/lmn...

#FILE.... /oracle/C11/origlogB/log_g12m1.dbf
#SAVED.... log_g12m1.dbf C11B06/12

BR280I Time stamp 1997-09-10 18.42.03
BR063I 11 of 13 files processed - 106.776 MB of 107.169 MB done.
BR204I Percentage done: 99.63%, estimated end time: 18:42
```

Detail Log of an Offline Backup

```

BR001I *****

BR202I Saving /oracle/C11/origlogA/log_g13m1.dbf
BR203I to /dev/rmt/lmn...

#FILE.... /oracle/C11/origlogA/log_g13m1.dbf
#SAVED... log_g13m1.dbf C11B06/13

BR280I Time stamp 1997-09-10 18.42.19
BR063I 12 of 13 files processed - 106.973 MB of 107.169 MB done.
BR204I Percentage done: 99.82%, estimated end time: 18:42
BR001I *****

BR202I Saving /oracle/C11/origlogB/log_g14m1.dbf
BR203I to /dev/rmt/lmn...

#FILE.... /oracle/C11/origlogB/log_g14m1.dbf
#SAVED... log_g14m1.dbf C11B06/14

BR280I Time stamp 1997-09-10 18.42.35
BR063I 13 of 13 files processed - 107.169 MB of 107.169 MB done.
BR204I Percentage done: 100.00%, estimated end time: 18:42
BR001I *****

BR304I Starting the database instance C11...
BR305I Startup of database instance C11 successful.

BR117I ARCHIVE LOG LIST after backup for database instance C11

Database log mode           Archive Mode
Automatic archival         Enabled
Archive destination        /oracle/C11/saparch/C11arch
Oldest online log sequence 232
Next log sequence to archive 235
Current log sequence        235      SCN: 9364
Database block size        8192

BR202I Saving reorg_log
BR203I to /dev/rmt/0mn...

BR202I Saving reorg_log
BR203I to /dev/rmt/lmn...

BR202I Saving /oracle/C11/sapbackup/bcvojtgt.aft
BR203I to /dev/rmt/0mn...

BR202I Saving /oracle/C11/sapbackup/bcvojtgt.aft
BR203I to /dev/rmt/lmn...

BR202I Saving /oracle/C11/sapbackup/backC11.log
BR203I to /dev/rmt/0mn...

BR202I Saving /oracle/C11/sapbackup/backC11.log
BR203I to /dev/rmt/lmn...

BR280I Time stamp 1997-09-10 18.44.02
BR226I Rewinding tape volume in device /dev/rmt/0m...

```

Detail Log of an Offline Backup

```
BR280I Time stamp 1997-09-10 18.44.02
BR226I Rewinding tape volume in device /dev/rmt/lm...

BR351I Restoring /oracle/C11/sapbackup/brlab0/.tape.hdr0
BR355I from /dev/rmt/0mn...

BR351I Restoring /oracle/C11/sapbackup/brlab1/.tape.hdr0
BR355I from /dev/rmt/1mn...

BR241I Checking label on volume in device /dev/rmt/0mn

BR241I Checking label on volume in device /dev/rmt/1mn

BR280I Time stamp 1997-09-10 18.44.20
BR226I Rewinding tape volume in device /dev/rmt/0m...

BR280I Time stamp 1997-09-10 18.44.20
BR226I Rewinding tape volume in device /dev/rmt/lm...

BR056I End of database backup: bcvojtgt aft 1997-09-10 18.45.14
BR052I BRBACKUP terminated successfully.
```

Supplements to a Detail Log

Supplements to a Detail Log

The following example shows how a supplement to the detail log might appear. The BRBACKUP backup was not started in this example.

Call: `brbackup -o dist,time -q`

```
BR051I BRBACKUP 4.0A
BR055I Start of database backup: bcvojuqu qub 1997-09-10 18.50.52

BR101I Parameters

oracle_sid          C11
oracle_home         /oracle/C11
oracle_profile      /oracle/C11/dbs/initC11.ora
sap_profile         /oracle/C11/dbs/initC11.sap
backup_mode        ALL
backup_type         offline
backup_dev_type     tape
compress           no
tape_copy_cmd       cpio
cpio_flags          -ovB
cpio_in_flags       -iuvB
cpio_disk_flags     -pdcu
dd_flags            obs=16k bs=16k
dd_in_flags         ibs=16k bs=16k
rewind              mt -t $ rew
rewind_offline      mt -t $ offl
tape_size           1200.000 MB
tape_address        /dev/rmt/0mn,/dev/rmt/1mn
tape_address_rew    /dev/rmt/0m,/dev/rmt/1m
volume_backup       C11B01,C11B02,C11B03,C11B04,C11B05,C11B06,C11B07,C11B08,C11B09,C11B10,
C11B11,C11B12,C11B13,C11B14,C11B15,C11B16,C11B17,C11B18,C11B19,C11B20,
C11B21,C11B22,C11B23,C11B24,C11B25,C11B26,C11B27,C11B28,C11B29,C11B30
expir_period        30
tape_use_count      100
exec_parallel       0

BR116I ARCHIVE LOG LIST before backup for database instance C11

Database log mode          Archive Mode
Automatic archival        Enabled
Archive destination       /oracle/C11/saparch/C11arch
Oldest online log sequence 232
Next log sequence to archive 235
Current log sequence       235      SCN: 9364
Database block size       8192

BR118I Tablespaces and data files

PSAPRAWD  ONLINE  ONLINE  /oracle/C11/sapraw/rawd_1 10493952 8
1073741829 /dev/vg00/rlvol5 RAW
PSAPROLL  ONLINE*  ONLINE  /oracle/C11/sapdata2/roll_1/roll.data1
20979712  2 1073807361 NOLINK FILE
PSAPSTABD ONLINE*  ONLINE  /oracle/C11/sapdata3/stabd_1/stabd.data1
10493952  4 1073807361 NOLINK FILE
```


Supplements to a Detail Log

```

PSAPSTABI ONLINE ONLINE /oracle/C11/sapdata4/stabi_1/stabi.data1
5251072 5 1073807361 NOLINK FILE
PSAPTEMP ONLINE ONLINE /oracle/C11/sapdata2/temp_1/temp.data1
5251072 3 1073807361 NOLINK FILE
PSAPTESTD ONLINE* ONLINE /oracle/C11/sapdata3/testd_1/testd.data1
4202496 6 1073807361 NOLINK FILE
PSAPTESTI ONLINE ONLINE /oracle/C11/sapdata4/testi_1/testi.data1
2105344 7 1073807361 NOLINK FILE
SYSTEM ONLINE* SYSTEM /oracle/C11/sapdata1/system_1/system.data1
52436992 1 1073807361 NOLINK FILE
    
```

BR119I Redo log files

```

/oracle/C11/origlogA/log_g11m1.dbf 205824 1 1073807361 INUSE NOLINK
FILE
/oracle/C11/mirrlogA/log_g11m2.dbf 205824 1 1073807361 INUSE NOLINK
FILE
/oracle/C11/origlogB/log_g12m1.dbf 205824 2 1073807361 INUSE NOLINK
FILE
/oracle/C11/mirrlogB/log_g12m2.dbf 205824 2 1073807361 INUSE NOLINK
FILE
/oracle/C11/origlogA/log_g13m1.dbf 205824 3 1073807361 INUSE NOLINK
FILE
/oracle/C11/mirrlogA/log_g13m2.dbf 205824 3 1073807361 INUSE NOLINK
FILE
/oracle/C11/origlogB/log_g14m1.dbf 205824 4 1073807361 INUSE NOLINK
FILE
/oracle/C11/mirrlogB/log_g14m2.dbf 205824 4 1073807361 INUSE NOLINK
FILE
    
```

BR120I Control files

```

/oracle/C11/sapdata1/cntrl/cntrlC11.dbf 336896 0 1073807361 NOLINK
FILE
/oracle/C11/sapdata2/cntrl/cntrlC11.dbf 336896 0 1073807361 NOLINK
FILE
/oracle/C11/sapdata3/cntrl/cntrlC11.dbf 336896 0 1073807361 NOLINK
FILE
/oracle/C11/sapdata4/cntrl/cntrlC11.dbf 336896 0 1073807361 NOLINK
FILE
    
```

BR379I Distribution of files on volume #1:

position	size	rate	compressed	duration [min:sec]	speed [MB/h]	name
4	52436992			3:32	849	/oracle/C11/sapdata1/system_1/system.data1
5	20979712			1:30	800	/oracle/C11/sapdata2/roll_1/roll.data1
6	10493952			1:25	424	/oracle/C11/sapdata3/stabd_1/stabd.data1
total:	83910656			6:27*	744	

BR379I Distribution of files on volume #2:

position	size	rate	compressed	duration	speed	name
----------	------	------	------------	----------	-------	------

Supplements to a Detail Log

		[min:sec]	[MB/h]
4	10493952	1:05	554
			/oracle/C11/sapraw/rawd_1
5	5251072	0:56	322
			/oracle/C11/sapdata4/stabi_1/stabi.data1
6	4202496	0:35	412
			/oracle/C11/sapdata3/testd_1/testd.data1
7	2105344	0:29	249
			/oracle/C11/sapdata4/testi_1/testi.data1
8	5251072	0:23	784
			/oracle/C11/sapdata2/temp_1/temp.data1
9	336896	0:17	68
			/oracle/C11/sapdata1/cntrl/cntrlC11.dbf
10	205824	0:16	44
			/oracle/C11/origlogA/log_g11ml.dbf
11	205824	0:16	44
			/oracle/C11/origlogA/log_g13ml.dbf
12	205824	0:16	44
			/oracle/C11/origlogB/log_g14ml.dbf
13	205824	0:15	47
			/oracle/C11/origlogB/log_g12ml.dbf
total:	28464128	4:48*	339

BR284I Time stamp 1997-09-10 18.50.57, elapsed time: 0:00

BR057I Backup of database: C11

BR058I BRBACKUP action ID: bcvojuqu

BR059I BRBACKUP function ID: qub

BR060I Tablespaces for backup: ALL

BR077I Database files for backup:

/oracle/C11/origlogA/log_g11ml.dbf

/oracle/C11/origlogB/log_g12ml.dbf

/oracle/C11/origlogA/log_g13ml.dbf

/oracle/C11/origlogB/log_g14ml.dbf

/oracle/C11/sapdata1/cntrl/cntrlC11.dbf

BR061I 13 files found for backup, total size 107.169 MB.

BR143I Backup type: offline

BR112I Files will not be compressed.

BR130I Backup device type: tape

BR102I Following backup devices will be used:

/dev/rmt/0mn,/dev/rmt/1mn

BR103I Following backup volumes will be used:

C11B07,C11B08

BR140I Number of parallel copy processes: 2.

BR124I Query option set - no processing.

BR056I End of database backup: bcvojuqu qub 1997-09-10 18.50.58

BR284I Time stamp 1997-09-10 18.50.58, elapsed time: 0:01

BR052I BRBACKUP terminated successfully.

Log for an RMAN Preparation Run

The following is the log of an RMAN preparation run in unattended mode.

Call: brbackup -d rman_prep -c

BR051I BRBACKUP 4.5B

BR055I Start of database backup: bcopyksix.rmp 1998-11-02 18.22.55

BR101I Parameters

```
oracle_sid          GC1
oracle_home         /oracle/GC1
oracle_profile      /oracle/GC1/dbs/initGC1.ora
sapdata_home        /oracle/GC1
sap_profile          /oracle/GC1/dbs/initGC1.sap
backup_mode         ALL
backup_type         online
backup_dev_type     rman_prep
exec_parallel       0
```

BR116I ARCHIVE LOG LIST before backup for database instance GC1

```
Database log mode          Archive Mode
Automatic archival        Enabled
Archive destination       /oracle/GC1/saparch/GC1arch
Oldest online log sequence 122
Next log sequence to archive 124
Current log sequence      124          SCN: 83681
Database block size       2048          Thread: 1
```

BR118I Tablespaces and data files

```
PSAPPROTD      ONLINE*      ONLINE      /oracle/GC1/sapdata5/protd_1/protd.data1  8390656
8 35389455     NOLINK      FILE
PSAPPROTI      ONLINE      ONLINE      /oracle/GC1/sapdata6/proti_1/proti.data1  4196352
9 35389455     NOLINK      FILE
```

Log for an RMAN Preparation Run

PSAPSTABD	ONLINE*	ONLINE	/oracle/GC1/sapdata2/stabd_1/stabd.data1	
16779264	6	35389455	NOLINK	FILE
PSAPSTABI	ONLINE	ONLINE	/oracle/GC1/sapdata3/stabi_1/stabi.data1	8390656
7	35389455		NOLINK	FILE
PSAPTESTD	ONLINE	ONLINE	/oracle/GC1/sapdata3/testd_1/testd.data1	4196352
10	35389455		NOLINK	FILE
PSAPTESTI	ONLINE	ONLINE	/oracle/GC1/sapdata2/testi_1/testi.data1	2099200
11	35389455		NOLINK	FILE
PSAPTTTTD	ONLINE	ONLINE	/oracle/GC1/sapdata6/ttttd_1/ttttd.data1	4196352
12	35389455		NOLINK	FILE
RBS	ONLINE*	ONLINE	/oracle/GC1/sapdata1/roll_1/roll.data1	15730688
2	35389455		NOLINK	FILE
SYSTEM	ONLINE*	SYSTEM	/oracle/GC1/sapdata1/system_1/system.data1	
83888128	1	35389455	NOLINK	FILE
TEMP	ONLINE	ONLINE	/oracle/GC1/sapdata1/temp_1/temp.data1	1050624
3	35389455		NOLINK	FILE
TOOLS	ONLINE*	ONLINE	/oracle/GC1/sapdata4/tools_1/tools.data1	
26216448	4	35389455	NOLINK	FILE
USERS	ONLINE	ONLINE	/oracle/GC1/sapdata4/users_1/users.data1	1050624
5	35389455		NOLINK	FILE

BR119I Redo log files

/oracle/GC1/origlog/redoGC101.log	512512	1	35389455	INUSE	NOLINK	FILE
/oracle/GC1/mirrlog/redoGC201.log	512512	1	35389455	INUSE	NOLINK	FILE
/oracle/GC1/origlog/redoGC102.log	512512	2	35389455	INUSE	NOLINK	FILE
/oracle/GC1/mirrlog/redoGC202.log	512512	2	35389455	INUSE	NOLINK	FILE
/oracle/GC1/origlog/redoGC103.log	512512	3	35389455	INUSE	NOLINK	FILE
/oracle/GC1/mirrlog/redoGC203.log	512512	3	35389455	INUSE	NOLINK	FILE

BR120I Control files

/oracle/GC1/sapdata1/cntrl/cntrl.ct1	1693696	0	35389455	NOLINK	FILE
/oracle/GC1/sapdata2/cntrl/cntrl.ct1	1693696	0	35389455	NOLINK	FILE
/oracle/GC1/sapdata3/cntrl/cntrl.ct1	1693696	0	35389455	NOLINK	FILE
/oracle/GC1/sapdata4/cntrl/cntrl.ct1	1693696	0	35389455	NOLINK	FILE

BR280I Time stamp 1998-11-02 18.22.57

BR057I Backup of database: GC1

Log for an RMAN Preparation Run

```

BR058I BRBACKUP action ID: bcypksix
BR059I BRBACKUP function ID: rmp
BR110I Backup mode: ALL
BR077I Database file for backup: /oracle/GC1/sapdata1/cntrl/cntrl.ctl
BR061I 13 files found for backup, total size 44.990 MB
BR143I Backup type: online
BR130I Backup device type: rman_prep
BR523I Preparation for using RMAN only - no backup
BR126I Unattended mode active - no operator confirmation required

BR525I Preparing backup of /oracle/GC1/sapdata5/protd_1/protd.data1 using RMAN...

#FILE..... /oracle/GC1/sapdata5/protd_1/protd.data1 65536 18.1089:1 3619

BR280I Time stamp 1998-11-02 18.23.09
BR063I 1 of 13 files processed - 0.062 MB of 44.990 MB done
BR204I Percentage done: 0.14%, estimated end time: 20:46
BR001I _____

BR525I Preparing backup of /oracle/GC1/sapdata6/proti_1/proti.data1 using RMAN...

#FILE..... /oracle/GC1/sapdata6/proti_1/proti.data1 65536 30.3970:1 2156

BR280I Time stamp 1998-11-02 18.23.22
BR063I 2 of 13 files processed - 0.125 MB of 44.990 MB done
BR204I Percentage done: 0.28%, estimated end time: 20:52
BR001I _____

BR525I Preparing backup of /oracle/GC1/sapdata2/stabd_1/stabd.data1 using RMAN...

#FILE..... /oracle/GC1/sapdata2/stabd_1/stabd.data1 851968 4.4895:1 189769

BR280I Time stamp 1998-11-02 18.23.32
BR063I 3 of 13 files processed - 0.938 MB of 44.990 MB done
BR204I Percentage done: 2.08%, estimated end time: 18:50

```

Log for an RMAN Preparation Run

BR001I * _____

BR525I Preparing backup of /oracle/GC1/sapdata3/stabi_1/stabi.data1 using RMAN...

#FILE..... /oracle/GC1/sapdata3/stabi_1/stabi.data1 327680 4.6661:1 70226

BR280I Time stamp 1998-11-02 18.23.44

BR063I 4 of 13 files processed - 1.250 MB of 44.990 MB done

BR204I Percentage done: 2.78%, estimated end time: 18:51

BR001I * _____

BR525I Preparing backup of /oracle/GC1/sapdata3/testd_1/testd.data1 using RMAN...

#FILE..... /oracle/GC1/sapdata3/testd_1/testd.data1 65536 84.4536:1 776

BR280I Time stamp 1998-11-02 18.23.58

BR063I 5 of 13 files processed - 1.312 MB of 44.990 MB done

BR204I Percentage done: 2.92%, estimated end time: 18:57

BR001I * _____

BR525I Preparing backup of /oracle/GC1/sapdata2/testi_1/testi.data1 using RMAN...

#FILE..... /oracle/GC1/sapdata2/testi_1/testi.data1 65536 88.6820:1 739

BR280I Time stamp 1998-11-02 18.24.14

BR063I 6 of 13 files processed - 1.375 MB of 44.990 MB done

BR204I Percentage done: 3.06%, estimated end time: 19:04

BR001I ** _____

BR525I Preparing backup of /oracle/GC1/sapdata6/ttttd_1/ttttd.data1 using RMAN...

#FILE..... /oracle/GC1/sapdata6/ttttd_1/ttttd.data1 65536 85.5561:1 766

BR280I Time stamp 1998-11-02 18.24.28

BR063I 7 of 13 files processed - 1.438 MB of 44.990 MB done

Log for an RMAN Preparation Run

BR204I Percentage done: 3.20%, estimated end time: 19:10

BR001I ** _____

BR525I Preparing backup of /oracle/GC1/sapdata1/roll_1/roll.data1 using RMAN...

#FILE..... /oracle/GC1/sapdata1/roll_1/roll.data1 3014656 3.5263:1 854908

BR280I Time stamp 1998-11-02 18.24.41

BR063I 8 of 13 files processed - 4.312 MB of 44.990 MB done

BR204I Percentage done: 9.59%, estimated end time: 18:41

BR001I ***** _____

BR525I Preparing backup of /oracle/GC1/sapdata1/system_1/system.data1 using RMAN...

#FILE..... /oracle/GC1/sapdata1/system_1/system.data1 39059456 2.6312:1 14844999

BR280I Time stamp 1998-11-02 18.25.38

BR063I 9 of 13 files processed - 41.562 MB of 44.990 MB done

BR204I Percentage done: 92.38%, estimated end time: 18:25

BR001I ***** _____

BR525I Preparing backup of /oracle/GC1/sapdata1/temp_1/temp.data1 using RMAN...

#FILE..... /oracle/GC1/sapdata1/temp_1/temp.data1 851968 4.4948:1 189546

BR280I Time stamp 1998-11-02 18.25.47

BR063I 10 of 13 files processed - 42.375 MB of 44.990 MB done

BR204I Percentage done: 94.19%, estimated end time: 18:25

BR001I ***** _____

BR525I Preparing backup of /oracle/GC1/sapdata4/tools_1/tools.data1 using RMAN...

#FILE..... /oracle/GC1/sapdata4/tools_1/tools.data1 983040 4.6577:1 211055

BR280I Time stamp 1998-11-02 18.25.57

Log for an RMAN Preparation Run

```

BR063I 11 of 13 files processed - 43.312 MB of 44.990 MB done
BR204I Percentage done: 96.27%, estimated end time: 18:26
BR001I *****
BR525I Preparing backup of /oracle/GC1/sapdata4/users_1/users.data1 using RMAN...

#FILE..... /oracle/GC1/sapdata4/users_1/users.data1 65536 92.5650:1 708

BR280I Time stamp 1998-11-02 18.26.07
BR063I 12 of 13 files processed - 43.375 MB of 44.990 MB done
BR204I Percentage done: 96.41%, estimated end time: 18:26
BR001I *****
BR201I Compressing /oracle/GC1/sapdata1/cntrl/cntrl.ct1

#FILE..... /oracle/GC1/sapdata1/cntrl/cntrl.ct1 12.6978:1 133385

BR280I Time stamp 1998-11-02 18.26.09
BR063I 13 of 13 files processed - 44.990 MB of 44.990 MB done
BR204I Percentage done: 100.00%, estimated end time: 18:26
BR001I *****

BR280I Time stamp 1998-11-02 18.26.11
BR533I Uncataloging backups of database files...

BR522I 12 files/savesets processed by RMAN

BR280I Time stamp 1998-11-02 18.26.33
BR534I Backups of database files uncataloged successfully

BR527I Savesets with 1 file:

saveset  size      rate  compressed  name

1 39059456 2.6312:1 14844999 /oracle/GC1/sapdata1/system_1/system.data1

```


Log for an RMAN Preparation Run

2	3014656	3.5263:1	854908	/oracle/GC1/sapdata1/roll_1/roll.data1
3	851968	4.4948:1	189546	/oracle/GC1/sapdata1/temp_1/temp.data1
4	983040	4.6577:1	211055	/oracle/GC1/sapdata4/tools_1/tools.data1
5	65536	92.5650:1	708	/oracle/GC1/sapdata4/users_1/users.data1
6	851968	4.4895:1	189769	/oracle/GC1/sapdata2/stabd_1/stabd.data1
7	327680	4.6661:1	70226	/oracle/GC1/sapdata3/stabi_1/stabi.data1
8	65536	18.1089:1	3619	/oracle/GC1/sapdata5/protd_1/protd.data1
9	65536	30.3970:1	2156	/oracle/GC1/sapdata6/proti_1/proti.data1
10	65536	84.4536:1	776	/oracle/GC1/sapdata3/testd_1/testd.data1
11	65536	88.6820:1	739	/oracle/GC1/sapdata2/testi_1/testi.data1
12	65536	85.5561:1	766	/oracle/GC1/sapdata6/ttttd_1/ttttd.data1

BR527I Savesets with 2 files:

saveset	size	rate	compressed	name
1	40042496	2.6596:1	15056054	/oracle/GC1/sapdata1/system_1/system.data1 /oracle/GC1/sapdata4/tools_1/tools.data1
2	3866624	3.7013:1	1044677	/oracle/GC1/sapdata2/stabd_1/stabd.data1 /oracle/GC1/sapdata1/roll_1/roll.data1
3	393216	5.3249:1	73845	/oracle/GC1/sapdata5/protd_1/protd.data1 /oracle/GC1/sapdata3/stabi_1/stabi.data1
4	131072	44.7040:1	2932	/oracle/GC1/sapdata6/proti_1/proti.data1 /oracle/GC1/sapdata3/testd_1/testd.data1
5	131072	87.0910:1	1505	/oracle/GC1/sapdata2/testi_1/testi.data1 /oracle/GC1/sapdata6/ttttd_1/ttttd.data1
6	917504	4.8225:1	190254	/oracle/GC1/sapdata1/temp_1/temp.data1 /oracle/GC1/sapdata4/users_1/users.data1

BR527I Savesets with 3 files:

saveset	size	rate	compressed	name
1	40894464	2.6823:1	15245823	/oracle/GC1/sapdata2/stabd_1/stabd.data1 /oracle/GC1/sapdata1/system_1/system.data1

Log for an RMAN Preparation Run

2	3407872	3.6693:1	928753	/oracle/GC1/sapdata4/tools_1/tools.data1 /oracle/GC1/sapdata5/protd_1/protd.data1 /oracle/GC1/sapdata3/stabi_1/stabi.data1 /oracle/GC1/sapdata1/roll_1/roll.data1
3	196608	53.1660:1	3698	/oracle/GC1/sapdata6/proti_1/proti.data1 /oracle/GC1/sapdata3/testd_1/testd.data1 /oracle/GC1/sapdata6/ttttd_1/ttttd.data1
4	983040	5.1470:1	190993	/oracle/GC1/sapdata2/testi_1/testi.data1 /oracle/GC1/sapdata1/temp_1/temp.data1 /oracle/GC1/sapdata4/users_1/users.data1

BR527I Savesets with 4 files:

saveset	size	rate	compressed	name
1	43909120	2.7272:1	16100731	/oracle/GC1/sapdata2/stabd_1/stabd.data1 /oracle/GC1/sapdata1/roll_1/roll.data1 /oracle/GC1/sapdata1/system_1/system.data1 /oracle/GC1/sapdata4/tools_1/tools.data1
2	524288	6.8287:1	76777	/oracle/GC1/sapdata5/protd_1/protd.data1 /oracle/GC1/sapdata6/proti_1/proti.data1 /oracle/GC1/sapdata3/stabi_1/stabi.data1 /oracle/GC1/sapdata3/testd_1/testd.data1
3	1048576	5.4682:1	191759	/oracle/GC1/sapdata2/testi_1/testi.data1 /oracle/GC1/sapdata6/ttttd_1/ttttd.data1 /oracle/GC1/sapdata1/temp_1/temp.data1 /oracle/GC1/sapdata4/users_1/users.data1

BR527I Savesets with 'tsp' files:

saveset	size	rate	compressed	name
1	65536	18.1089:1	3619	/oracle/GC1/sapdata5/protd_1/protd.data1
2	65536	30.3970:1	2156	/oracle/GC1/sapdata6/proti_1/proti.data1
3	851968	4.4895:1	189769	/oracle/GC1/sapdata2/stabd_1/stabd.data1

Log for an RMAN Preparation Run

4	327680	4.6661:1	70226	/oracle/GC1/sapdata3/stabi_1/stabi.data1
5	65536	84.4536:1	776	/oracle/GC1/sapdata3/testd_1/testd.data1
6	65536	88.6820:1	739	/oracle/GC1/sapdata2/testi_1/testi.data1
7	65536	85.5561:1	766	/oracle/GC1/sapdata6/ttttd_1/ttttd.data1
8	3014656	3.5263:1	854908	/oracle/GC1/sapdata1/roll_1/roll.data1
9	39059456	2.6312:1	14844999	/oracle/GC1/sapdata1/system_1/system.data1
10	851968	4.4948:1	189546	/oracle/GC1/sapdata1/temp_1/temp.data1
11	983040	4.6577:1	211055	/oracle/GC1/sapdata4/tools_1/tools.data1
12	65536	92.5650:1	708	/oracle/GC1/sapdata4/users_1/users.data1

BR527I Savesets with 'all' files:

save	set	size	rate	compressed	name
1	45481984	2.7785:1	16369267	/oracle/GC1/sapdata5/protd_1/protd.data1 /oracle/GC1/sapdata6/proti_1/proti.data1 /oracle/GC1/sapdata2/stabd_1/stabd.data1 /oracle/GC1/sapdata3/stabi_1/stabi.data1 /oracle/GC1/sapdata3/testd_1/testd.data1 /oracle/GC1/sapdata2/testi_1/testi.data1 /oracle/GC1/sapdata6/ttttd_1/ttttd.data1 /oracle/GC1/sapdata1/roll_1/roll.data1 /oracle/GC1/sapdata1/system_1/system.data1 /oracle/GC1/sapdata1/temp_1/temp.data1 /oracle/GC1/sapdata4/tools_1/tools.data1 /oracle/GC1/sapdata4/users_1/users.data1	

BR056I End of database backup: bcypksix.rmp 1998-11-02 18.26.33

BR052I BRBACKUP terminated successfully

Summary BRBACKUP Log

Summary BRBACKUP Log

bcvmrnej	qub	1997-09-01	18.37.33	1997-09-01	18.37.57	0	ALL
	offline	tape	-----					
bcvmrnfw	qub	1997-09-01	18.38.12	1997-09-01	18.38.30	0	partial
	online	tape	-----					
bcvmuqta	pnt	1997-09-02	09.56.30	1997-09-02	09.59.10	5		
bcvmuwjy	pnt	1997-09-02	11.00.10	1997-09-02	11.02.47	5	partial
	online	tape	-----					
bcvmuwrv	pnt	1997-09-02	11.03.35	0000-00-00	00.00.00	6	partial
	online	tape	-----					
bcvmuycb	pnt	1997-09-02	11.19.17	0000-00-00	00.00.00	6	partial
	online	tape	-----					
bcvmvisp	pnf	1997-09-02	13.19.07	1997-09-02	13.19.28	3
bcvmviue	pnf	1997-09-02	13.19.48	1997-09-02	13.19.55	3
bcvmvixi	pnf	1997-09-02	13.21.10	1997-09-02	13.21.17	3
bcvmvjft	pnf	1997-09-02	13.24.49	1997-09-02	13.25.02	3
bcvmvjgn	pnf	1997-09-02	13.25.09	1997-09-02	13.25.52	5	partial
	online_split	util_file	-----					
bcvmvlnu	qub	1997-09-02	13.50.50	1997-09-02	13.51.19	0	ALL
	online	disk	-----					
bcvmvlqo	afd	1997-09-02	13.52.02	1997-09-02	13.52.12	2	0.0000:0	
bcvmvlts	afd	1997-09-02	13.53.24	1997-09-02	14.05.02	0	31.6413:1	ALL
	offline	disk	---kw					
bcvmvsuy	pnf	1997-09-02	15.12.48	1997-09-02	15.13.47	5	partial
	online_split	util_file	-----					
bcvnglcm	qub	1997-09-04	19.28.12	1997-09-04	19.28.43	0	ALL
	online	tape	-----					
bcvngnbq	qub	1997-09-04	19.50.22	1997-09-04	19.50.48	0	ALL
	online	tape	-----					
bcvnjiyj	pnt	1997-09-05	09.42.41	1997-09-05	09.43.27	4	partial
	online	tape	-----					
bcvnjjax	pnt	1997-09-05	09.43.47	0000-00-00	00.00.00	6	partial
	online	tape	-----					
bcvnjjog	pnt	1997-09-05	09.49.34	1997-09-05	09.51.30	5	partial
	online	tape	-----					
bcvnjjtd	pnt	1997-09-05	09.51.41	1997-09-05	09.51.46	2
bcvnjjjuq	pnt	1997-09-05	09.52.20	1997-09-05	09.52.44	5	partial
	online	tape	-----					
bcvnjjxd	pnt	1997-09-05	09.53.25	1997-09-05	09.53.41	5	partial
	online	tape	-----					
bcvnjkgo	pnt	1997-09-05	09.57.30	0000-00-00	00.00.00	6	partial
	online	tape	-----					
bcvnkqgl	pnt	1997-09-05	15.57.59	1997-09-05	16.01.42	0	partial
	online	tape	-----					
bcvnkrun	aft	1997-09-05	16.15.21	1997-09-05	16.29.32	0	31.2059:1	ALL
	offline	tape	---kw					
bcvnkxnz	aft	1997-09-05	17.20.07	1997-09-05	17.24.50	4	0.0000:0	ALL
	offline	tape	---kw					
bcvnkzko	aft	1997-09-05	17.41.10	1997-09-05	17.59.28	0	31.1195:1	ALL
	offline	tape	---kw					
bcvnkcbq	pnd	1997-09-05	18.11.06	1997-09-05	18.11.35	0	partial
	online	disk	-----					
bcvnydrc	aft	1997-09-08	09.57.12	1997-09-08	10.04.43	5	ALL
	offline	tape	-----					
bcvnyejp	qub	1997-09-08	10.05.13	1997-09-08	10.05.23	0	ALL
	offline	tape	-----					
bcvnyekg	qub	1997-09-08	10.05.30	1997-09-08	10.05.34	0	ALL
	offline	tape	--f--					
bcvnyepn	aft	1997-09-08	10.07.47	1997-09-08	10.11.34	0	ALL
	offline	tape	--f--					
bcvnyezy	aft	1997-09-08	10.12.18	1997-09-08	10.20.22	0	ALL
	offline	tape	-----					
bcvnyqfd	afd	1997-09-08	12.18.29	1997-09-08	12.18.47	3	0.0000:0

Summary BRBACKUP Log

bcvzvsj	pft	1997-09-08	18.13.29	1997-09-08	18.18.31	0	partial
	offline	tape	-----					
bcvzwt	aft	1997-09-08	18.19.17	1997-09-08	18.26.33	0	ALL
	offline	tape	--f--					
bcvzxqz	aft	1997-09-08	18.35.25	1997-09-08	18.42.36	0	ALL
	offline	tape	--f--					
bcvoevy	anf	1997-09-09	15.28.10	1997-09-09	15.28.47	2	ALL
	online	util_file	-----					
bcvoexv	dbv	1997-09-09	15.28.59	1997-09-09	15.29.37	2	ALL
	online	tape	----w					
bcvoeozo	anf	1997-09-09	15.29.44	1997-09-09	15.29.53	2	ALL
	online	util_file	-----					
bcvoeuqh	dbv	1997-09-09	18.25.59	1997-09-09	18.26.22	5	partial
	online	tape	----w					
bcvoiwv	ant	1997-09-10	14.15.17	1997-09-10	14.26.27	0	33.0751:1	ALL
	online_cons	tape	---kw					
bcvoixsg	pnt	1997-09-10	14.32.22	1997-09-10	14.32.31	2
bcvoixsv	pnt	1997-09-10	14.32.37	1997-09-10	14.32.47	3
bcvoixty	pnt	1997-09-10	14.33.06	1997-09-10	14.36.07	0	partial
	online	tape	-----					
bcvojmwr	qub	1997-09-10	17.23.17	1997-09-10	17.23.42	0	ALL
	online	tape	-----					
bcvojnag	qub	1997-09-10	17.24.50	1997-09-10	17.24.54	3
bcvojndc	qub	1997-09-10	17.26.04	1997-09-10	17.26.06	3
bcvojnho	ant	1997-09-10	17.28.00	1997-09-10	17.29.17	4	ALL
	online	tape	-----					
bcvojoed	qub	1997-09-10	17.37.47	1997-09-10	17.38.07	2
bcvojojk	qub	1997-09-10	17.40.04	1997-09-10	17.40.17	0	ALL
	online	tape	-----					
bcvojonp	qub	1997-09-10	17.41.53	1997-09-10	17.42.02	2
bcvojooh	qub	1997-09-10	17.42.11	1997-09-10	17.42.19	2
bcvojoox	qub	1997-09-10	17.42.27	1997-09-10	17.42.30	3
bcvojrno	qub	1997-09-10	18.15.40	1997-09-10	18.16.07	0	ALL
	online	tape	-----					
bcvojsbg	pnt	1997-09-10	18.21.36	1997-09-10	18.23.59	5	partial
	online	tape	-----					
bcvojshb	qub	1997-09-10	18.24.07	1997-09-10	18.24.22	0	ALL
	online	tape	-----					
bcvojsr	aft	1997-09-10	18.29.09	1997-09-10	18.29.49	5	ALL
	offline	tape	-----					
bcvojtat	aft	1997-09-10	18.32.39	1997-09-10	18.33.05	5	ALL
	offline	tape	-----					
bcvojtga	qub	1997-09-10	18.34.56	1997-09-10	18.35.13	0	ALL
	online	tape	-----					
bcvojtgt	aft	1997-09-10	18.35.15	1997-09-10	18.45.14	0	ALL
	offline	tape	-----					
bcvojuoq	qub	1997-09-10	18.49.56	1997-09-10	18.50.14	0	ALL
	online	tape	-----					
bcvojuqu	qub	1997-09-10	18.50.52	1997-09-10	18.50.58	0	ALL
	offline	tape	-----					

Detail BRARCHIVE Log

Detail BRARCHIVE Log

Log from the archiving run for existing offline redo log files and subsequent deletion of these files. Archiving was performed on two tape devices in parallel and in unattended mode.

Call: brarchive -ssd -c

```
BR002I BRARCHIVE 4.0A
BR006I Start of offline redo log processing: acvojevun ssd 1997-09-
10 19.03.45
```

BR101I Parameters

```
oracle_sid          C11
oracle_home         /oracle/C11
oracle_profile      /oracle/C11/dbs/initC11.ora
sap_profile         /oracle/C11/dbs/initC11.sap
backup_dev_type     tape
compress           no
tape_copy_cmd       cpio
cpio_flags          -ovB
cpio_in_flags       -iuVB
rewind              mt -t $ rew
rewind_offline      mt -t $ offl
tape_size           1200.000 MB
tape_address        /dev/rmt/0mn,/dev/rmt/1mn
tape_address_rew    /dev/rmt/0m,/dev/rmt/1m
volume_archive
C11A01,C11A02,C11A03,C11A04,C11A05,C11A06,C11A07,C11A08,C11A09,C11A
10,
C11A11,C11A12,C11A13,C11A14,C11A15,C11A16,C11A17,C11A18,C11A19,C11A
20,
C11A21,C11A22,C11A23,C11A24,C11A25,C11A26,C11A27,C11A28,C11A29,C11A
30
expir_period        30
tape_use_count      100
```

```
BR280I Time stamp 1997-09-10 19.03.50
BR008I Offline redo log processing for database instance: C11
BR009I BRARCHIVE action ID: acvojevun
BR010I BRARCHIVE function ID: ssd
BR011I 4 offline redo log files found for processing, total size
0.560 MB.
BR112I Files will not be compressed.
BR130I Backup device type: tape
BR102I Following backup devices will be used:
/dev/rmt/0mn,/dev/rmt/1mn
BR103I Following backup volumes will be used: C11A04,C11A05
BR126I Unattended mode active - no confirmation required.
```

```
BR208I Volume with name:
C11A04|C11A05 required in device /dev/rmt/0mn
```

Detail BRARCHIVE Log

```
BR280I Time stamp 1997-09-10 19.03.50
BR226I Rewinding tape volume in device /dev/rmt/0m...

BR351I Restoring /oracle/C11/sapreorg/.tape.hdr0
BR355I from /dev/rmt/0mn...

BR241I Checking label on volume in device /dev/rmt/0mn

BR208I Volume with name C11A05 required in device /dev/rmt/1mn

BR280I Time stamp 1997-09-10 19.04.03
BR226I Rewinding tape volume in device /dev/rmt/1m...

BR351I Restoring /oracle/C11/sapreorg/.tape.hdr0
BR355I from /dev/rmt/1mn...

BR241I Checking label on volume in device /dev/rmt/1mn

BR280I Time stamp 1997-09-10 19.04.16
BR226I Rewinding tape volume in device /dev/rmt/0m...

BR202I Saving /oracle/C11/sapreorg/.tape.hdr0
BR203I to /dev/rmt/0mn...

BR209I Volume in device /dev/rmt/0mn has name C11A04

BR202I Saving init_ora
BR203I to /dev/rmt/0mn...

BR202I Saving /oracle/C11/dbs/initC11.sap
BR203I to /dev/rmt/0mn...

BR280I Time stamp 1997-09-10 19.04.53
BR226I Rewinding tape volume in device /dev/rmt/1m...

BR202I Saving /oracle/C11/sapreorg/.tape.hdr0
BR203I to /dev/rmt/1mn...

BR209I Volume in device /dev/rmt/1mn has name C11A05

BR202I Saving init_ora
BR203I to /dev/rmt/1mn...

BR202I Saving /oracle/C11/dbs/initC11.sap
BR203I to /dev/rmt/1mn...

BR280I Time stamp 1997-09-10 19.05.28

BR202I Saving /oracle/C11/saparch/C11arch1_231.dbf
BR203I to /dev/rmt/0mn...

BR202I Saving /oracle/C11/saparch/C11arch1_231.dbf
```

Detail BRARCHIVE Log

```
BR203I to /dev/rmt/1mn...

#ARCHIVE.. /oracle/C11/saparch/C11arch1_231.dbf
#SAVED.... C11arch1_231.dbf  C11A05/4

#ARCHIVE.. /oracle/C11/saparch/C11arch1_231.dbf
#SAVED.... C11arch1_231.dbf  C11A04/4

BR015I Offline redo log file /oracle/C11/saparch/C11arch1_231.dbf
deleted.

BR280I Time stamp 1997-09-10 19.05.47
BR014I 1 of 4 offline redo log files processed - 0.115 MB of 0.560
MB done.
BR204I Percentage done: 20.59%, estimated end time: 19:06
BR001I *****

BR202I Saving /oracle/C11/saparch/C11arch1_232.dbf
BR203I to /dev/rmt/0mn...

BR202I Saving /oracle/C11/saparch/C11arch1_232.dbf
BR203I to /dev/rmt/1mn...

#ARCHIVE.. /oracle/C11/saparch/C11arch1_232.dbf
#SAVED.... C11arch1_232.dbf  C11A04/5

#ARCHIVE.. /oracle/C11/saparch/C11arch1_232.dbf
#SAVED.... C11arch1_232.dbf  C11A05/5

BR015I Offline redo log file /oracle/C11/saparch/C11arch1_232.dbf
deleted.

BR280I Time stamp 1997-09-10 19.06.05
BR014I 2 of 4 offline redo log files processed - 0.167 MB of 0.560
MB done.
BR204I Percentage done: 29.84%, estimated end time: 19:07
BR001I *****

BR202I Saving /oracle/C11/saparch/C11arch1_233.dbf
BR203I to /dev/rmt/0mn...

BR202I Saving /oracle/C11/saparch/C11arch1_233.dbf
BR203I to /dev/rmt/1mn...

#ARCHIVE.. /oracle/C11/saparch/C11arch1_233.dbf
#SAVED.... C11arch1_233.dbf  C11A04/6

#ARCHIVE.. /oracle/C11/saparch/C11arch1_233.dbf
#SAVED.... C11arch1_233.dbf  C11A05/6

BR015I Offline redo log file /oracle/C11/saparch/C11arch1_233.dbf
deleted.
```


Detail BRARCHIVE Log

```

BR280I Time stamp 1997-09-10 19.06.25
BR014I 3 of 4 offline redo log files processed - 0.363 MB of 0.560
MB done.
BR204I Percentage done: 64.92%, estimated end time: 19:06
BR001I *****

BR202I Saving /oracle/C11/saparch/C11arch1_234.dbf
BR203I to /dev/rmt/0mn...

BR202I Saving /oracle/C11/saparch/C11arch1_234.dbf
BR203I to /dev/rmt/1mn...

#ARCHIVE.. /oracle/C11/saparch/C11arch1_234.dbf
#SAVED.... C11arch1_234.dbf C11A04/7

#ARCHIVE.. /oracle/C11/saparch/C11arch1_234.dbf
#SAVED.... C11arch1_234.dbf C11A05/7

BR015I Offline redo log file /oracle/C11/saparch/C11arch1_234.dbf
deleted.

BR280I Time stamp 1997-09-10 19.06.43
BR014I 4 of 4 offline redo log files processed - 0.560 MB of 0.560
MB done.
BR204I Percentage done: 100.00%, estimated end time: 19:06
BR001I *****

BR016I 4 offline redo log files processed, total size 0.560 MB.

BR202I Saving reorg_log
BR203I to /dev/rmt/0mn...

BR202I Saving /oracle/C11/saparch/acvojevun.ssd
BR203I to /dev/rmt/0mn...

BR202I Saving /oracle/C11/saparch/archC11.log
BR203I to /dev/rmt/0mn...

BR280I Time stamp 1997-09-10 19.07.27
BR226I Rewinding tape volume in device /dev/rmt/0m...

BR351I Restoring /oracle/C11/sapreorg/.tape.hdr0
BR355I from /dev/rmt/0mn...

BR241I Checking label on volume in device /dev/rmt/0mn

BR280I Time stamp 1997-09-10 19.07.39
BR226I Rewinding tape volume in device /dev/rmt/0m...

BR202I Saving reorg_log
BR203I to /dev/rmt/1mn...

BR202I Saving /oracle/C11/saparch/acvojevun.ssd

```

Detail BRARCHIVE Log

```
BR203I to /dev/rmt/1mn...

BR202I Saving /oracle/C11/saparch/archC11.log
BR203I to /dev/rmt/1mn...

BR280I Time stamp 1997-09-10 19.08.52
BR226I Rewinding tape volume in device /dev/rmt/1m...

BR351I Restoring /oracle/C11/sapreorg/.tape.hdr0
BR355I from /dev/rmt/1mn...

BR241I Checking label on volume in device /dev/rmt/1mn

BR280I Time stamp 1997-09-10 19.09.03
BR226I Rewinding tape volume in device /dev/rmt/1m...

BR007I End of offline redo log processing: acvojvun ssd 1997-09-10
19.09.43
BR003I BRARCHIVE terminated successfully.
```

Summary BRARCHIVE Log

```
#DISK..... 219 /oracle/C11/saparch/C11arch1_219.dbf 1997-09-02 09.59.24
205824      8332 1

#DISKSAV.. acvnmni sve /oracle/C11/sapreorg/C11arch1_219.dbf 1997-09-05
15.16.08

#DISKDEL..

#

#DISK..... 220 /oracle/C11/saparch/C11arch1_220.dbf 1997-09-02 13.25.43
93184      8494 1

#DISKSAV.. acvnmni sve /oracle/C11/sapreorg/C11arch1_220.dbf 1997-09-05
15.16.09

#DISKDEL..

#

#DISK..... 221 /oracle/C11/saparch/C11arch1_221.dbf 1997-09-02 15.13.38
74752      8575 1

#DISKSAV.. acvnmni sve /oracle/C11/sapreorg/C11arch1_221.dbf 1997-09-05
15.16.11

#DISKDEL..

#

#DISK..... 222 /oracle/C11/saparch/C11arch1_222.dbf 1997-09-02 16.51.15
205824      8603 1

#DISKSAV.. acvnmni sve /oracle/C11/sapreorg/C11arch1_222.dbf 1997-09-05
15.16.12.....

#DISKDEL.....

#

#* C11 disk acvnmni sve 1997-09-05 15.15.54 1997-09-05 15.16.13 0
..... -----

#

#ARCHIVE.. 219 /oracle/C11/saparch/C11arch1_219.dbf 1997-09-02 09.59.24
205824      8332 1

#SAVED.... acvnlcex svd #C11A01/4 1997-09-05 18.13.26

#COPIED...

#DELETED.. acvnlcex svd 1997-09-05 18.13.26

#

#ARCHIVE.. 220 /oracle/C11/saparch/C11arch1_220.dbf 1997-09-02 13.25.43
93184      8494 1

#SAVED.... acvnlcex svd #C11A01/5 1997-09-05 18.13.42

#COPIED...

#DELETED.. acvnlcex svd 1997-09-05 18.13.42

#
```

Summary BRARCHIVE Log

```
#ARCHIVE.. 221 /oracle/C11/saparch/C11arch1_221.dbf 1997-09-02 15.13.38
74752      8575  1
#SAVED.... acvnlcex svd #C11A01/6    1997-09-05 18.13.59
#COPIED...
#DELETED.. acvnlcex svd 1997-09-05 18.13.59
#
#ARCHIVE.. 222 /oracle/C11/saparch/C11arch1_222.dbf 1997-09-02 16.51.15
205824     8603  1
#SAVED.... acvnlcex svd #C11A01/7    1997-09-05 18.14.19
#COPIED...
#DELETED.. acvnlcex svd 1997-09-05 18.14.19
#
#* C11 tape acvnlcex svd 1997-09-05 18.12.31 1997-09-05 18.15.18  0
.....  -----
#
#ARCHIVE.. 223 /oracle/C11/saparch/C11arch1_223.dbf 1997-09-02 17.13.55
203776     8698  1
#SAVED.... acvnlcmf ssv #C11A01/4    1997-09-05 18.17.21    2.7379:1    74427
#COPIED... acvnlcmf ssv #C11A02/4    1997-09-05 18.17.21    2.7379:1    74427
#DELETED..
#
#ARCHIVE.. 224 /oracle/C11/saparch/C11arch1_224.dbf 1997-09-02 17.44.41
205824     8761  1
#SAVED.... acvnlcmf ssv #C11A01/5    1997-09-05 18.17.46    3.5148:1    58559
#COPIED... acvnlcmf ssv #C11A02/5    1997-09-05 18.17.46    3.5148:1    58559
#DELETED..
#
#ARCHIVE.. 225 /oracle/C11/saparch/C11arch1_225.dbf 1997-09-02 17.59.37
205824     8796  1
#SAVED.... acvnlcmf ssv #C11A01/6    1997-09-05 18.18.07    3.0547:1    67379
#COPIED... acvnlcmf ssv #C11A02/6    1997-09-05 18.18.07    3.0547:1    67379
#DELETED..
#
#ARCHIVE.. 226 /oracle/C11/saparch/C11arch1_226.dbf 1997-09-04 09.25.42
205824     8860  1
#SAVED.... acvnlcmf ssv #C11A01/7    1997-09-05 18.18.28    2.9085:1    70767
#COPIED... acvnlcmf ssv #C11A02/7    1997-09-05 18.18.28    2.9085:1    70767
#DELETED..
#
#* C11 tape acvnlcmf ssv 1997-09-05 18.15.41 1997-09-05 18.20.46  0
3.0290:1  ---kw-
```

Summary BRARCHIVE Log

```
#
#* C11 tape acvoizij sve 1997-09-10 14.50.37 1997-09-10 14.50.51 2
..... -----
#
#ARCHIVE.. 227 /oracle/C11/saparch/C11arch1_227.dbf 1997-09-05 16.00.36
164864 8935 1
#SAVED.... acvoizjs svd #C11A03/4 1997-09-10 14.52.42
#COPIED...
#DELETED.. acvoizjs svd 1997-09-10 14.52.42
#
#ARCHIVE.. 228 /oracle/C11/saparch/C11arch1_228.dbf 1997-09-05 18.11.33
88064 9057 1
#SAVED.... acvoizjs svd #C11A03/5 1997-09-10 14.53.09
#COPIED...
#DELETED.. acvoizjs svd 1997-09-10 14.53.09
#
#* C11 tape acvoizjs svd 1997-09-10 14.51.12 1997-09-10 14.54.36 0
..... -----
#
#ARCHIVE.. 229 /oracle/C11/saparch/C11arch1_229.dbf 1997-09-09 15.29.56
205824 9108 1
#SAVED.... acvoizsq ssv #C11A01/4 1997-09-10 14.57.07 3.8055:1 54086
#COPIED... acvoizsq ssv #C11A02/4 1997-09-10 14.57.07 3.8055:1 54086
#DELETED..
#
#ARCHIVE.. 230 /oracle/C11/saparch/C11arch1_230.dbf 1997-09-10 14.17.36
11264 9235 1
#SAVED.... acvoizsq ssv #C11A01/5 1997-09-10 14.57.24 2.5312:1 4450
#COPIED... acvoizsq ssv #C11A02/5 1997-09-10 14.57.24 2.5312:1 4450
#DELETED..
#
#* C11 tape acvoizsq ssv 1997-09-10 14.55.04 1997-09-10 14.59.27 0
3.7086:1 ---kw-
#
#* C11 tape acvojmye qua 1997-09-10 17.23.56 1997-09-10 17.24.04 0
..... -----
#
#ARCHIVE.. 231 /oracle/C11/saparch/C11arch1_231.dbf 1997-09-10 14.24.47
120832 9243 1
#SAVED.... acvojevun ssd #C11A04/4 1997-09-10 19.05.46
#COPIED... acvojevun ssd #C11A05/4 1997-09-10 19.05.46
```

Summary BRARCHIVE Log

```
#DELETED.. acvojevun ssd 1997-09-10 19.05.46
#
#ARCHIVE.. 232 /oracle/C11/saparch/C11arch1_232.dbf 1997-09-10 14.35.05
54272 9261 1
#SAVED... acvojevun ssd #C11A04/5 1997-09-10 19.06.04
#COPIED... acvojevun ssd #C11A05/5 1997-09-10 19.06.04
#DELETED.. acvojevun ssd 1997-09-10 19.06.04
#
#ARCHIVE.. 233 /oracle/C11/saparch/C11arch1_233.dbf 1997-09-10 18.34.32
205824 9285 1
#SAVED... acvojevun ssd #C11A04/6 1997-09-10 19.06.25
#COPIED... acvojevun ssd #C11A05/6 1997-09-10 19.06.25
#DELETED.. acvojevun ssd 1997-09-10 19.06.25
#
#ARCHIVE.. 234 /oracle/C11/saparch/C11arch1_234.dbf 1997-09-10 18.34.46
205824 9361 1
#SAVED... acvojevun ssd #C11A04/7 1997-09-10 19.06.42
#COPIED... acvojevun ssd #C11A05/7 1997-09-10 19.06.42
#DELETED.. acvojevun ssd 1997-09-10 19.06.42
#
#* C11 tape acvojevun ssd 1997-09-10 19.03.45 1997-09-10 19.09.43 0
..... -----
#
```

BRRESTORE Logs: Restoring a Complete Backup

BRRESTORE Logs: Restoring a Complete Backup

This example shows the detail log of a BRRESTORE run, in which a complete backup was restored. This was done on two tape devices in parallel.

Call: **brrestore -m full**

```
BR401I BRRESTORE 4.0A
BR405I Start of database restore: rcvojvfh rsb 1997-09-10 18.57.09

BR101I Parameters

oracle_sid          C11
oracle_home         /oracle/C11
oracle_profile      /oracle/C11/dbs/initC11.ora
sap_profile         /oracle/C11/dbs/initC11.sap
backup_mode         FULL
backup_dev_type     tape
compress           no
tape_copy_cmd       cpio
cpio_in_flags       -iuvB
cpio_disk_flags     -pdcu
dd_in_flags         ibs=16k bs=16k
rewind              mt -t $ rew
rewind_offline      mt -t $ offl
tape_pos_cmd        mt -t $ fsf $
tape_address        /dev/rmt/0mn,/dev/rmt/1mn
tape_address_rew    /dev/rmt/0m,/dev/rmt/1m
exec_parallel       0

BR428W File /oracle/C11/sapraw/rawd_1 will be overwritten.
BR428W File /oracle/C11/sapdata2/roll_1/roll.data1 will be overwritten.
BR428W File /oracle/C11/sapdata3/stabd_1/stabd.data1 will be overwritten.
BR428W File /oracle/C11/sapdata4/stabi_1/stabi.data1 will be overwritten.
BR428W File /oracle/C11/sapdata2/temp_1/temp.data1 will be overwritten.
BR428W File /oracle/C11/sapdata3/testd_1/testd.data1 will be overwritten.
BR428W File /oracle/C11/sapdata4/testi_1/testi.data1 will be overwritten.
BR428W File /oracle/C11/sapdata1/system_1/system.data1 will be
overwritten.
BR428W File /oracle/C11/origlogA/log_g11m1.dbf will be overwritten.
BR428W File /oracle/C11/mirrlogA/log_g11m2.dbf will be overwritten.
BR428W File /oracle/C11/origlogB/log_g12m1.dbf will be overwritten.
BR428W File /oracle/C11/mirrlogB/log_g12m2.dbf will be overwritten.
BR428W File /oracle/C11/origlogA/log_g13m1.dbf will be overwritten.
BR428W File /oracle/C11/mirrlogA/log_g13m2.dbf will be overwritten.
BR428W File /oracle/C11/origlogB/log_g14m1.dbf will be overwritten.
BR428W File /oracle/C11/mirrlogB/log_g14m2.dbf will be overwritten.
BR428W File /oracle/C11/sapdata1/cntrl/cntrlC11.dbf will be overwritten.
BR428W File /oracle/C11/sapdata2/cntrl/cntrlC11.dbf will be overwritten.
BR428W File /oracle/C11/sapdata3/cntrl/cntrlC11.dbf will be overwritten.
BR428W File /oracle/C11/sapdata4/cntrl/cntrlC11.dbf will be overwritten.
BR280I Time stamp 1997-09-10 18.57.10
BR256I Please enter 'cont' to continue, 'stop' to cancel the program:
BR257I Your reply: 'cont'
BR259I Program execution will be continued...

BR280I Time stamp 1997-09-10 18.57.13
BR407I Restore of database: C11
```

BRRESTORE Logs: Restoring a Complete Backup

```
BR408I BRRESTORE action ID: rcvojvfh
BR409I BRRESTORE function ID: rsb
BR410I Tablespaces for restore: FULL
BR411I Database files for restore:
/oracle/C11/origlogA/log_g11m1.dbf
/oracle/C11/mirrlogA/log_g11m2.dbf
/oracle/C11/origlogB/log_g12m1.dbf
/oracle/C11/mirrlogB/log_g12m2.dbf
/oracle/C11/origlogA/log_g13m1.dbf
/oracle/C11/mirrlogA/log_g13m2.dbf
/oracle/C11/origlogB/log_g14m1.dbf
/oracle/C11/mirrlogB/log_g14m2.dbf
/oracle/C11/sapdata/cntrl/cntrlC11.dbf
/oracle/C11/sapdata2/cntrl/cntrlC11.dbf
/oracle/C11/sapdata3/cntrl/cntrlC11.dbf
/oracle/C11/sapdata4/cntrl/cntrlC11.dbf
BR419I Files will be restored from backup: bcvojtgt.aft 1997-09-10
18.35.15
BR416I 13 files found to restore, total size 107.169 MB.
BR424I Files will not be decompressed.
BR421I Backup device type for restore: tape
BR102I Following backup devices will be used:
/dev/rmt/0mn,/dev/rmt/1mn
BR103I Following backup volumes will be used:
C11B06 1997-09-10 18.35.15, C11B05 1997-09-10 18.35.15
BR140I Number of parallel copy processes: 2.
BR280I Time stamp 1997-09-10 18.57.13
BR256I Please enter 'cont' to continue, 'stop' to cancel the program:
BR257I Your reply: 'cont'
BR259I Program execution will be continued...

BR208I Volume with name:
C11B06 1997-09-10 18.35.15 | C11B05 1997-09-10 18.35.15
  required in device /dev/rmt/0mn
BR210I Please mount the volume, if you have not already done so.
BR280I Time stamp 1997-09-10 18.57.16
BR256I Please enter 'cont' to continue, 'stop' to cancel the program:
BR257I Your reply: 'cont'
BR259I Program execution will be continued...

BR280I Time stamp 1997-09-10 18.57.18
BR226I Rewinding tape volume in device /dev/rmt/0m...

BR351I Restoring /oracle/C11/sapbackup/.tape.hdr0
BR355I from /dev/rmt/0mn...

BR241I Checking label on volume in device /dev/rmt/0mn

BR209I Volume in device /dev/rmt/0mn has name C11B05 1997-09-10 18.35.15

BR208I Volume with name C11B06 1997-09-10 18.35.15 required in device
/dev/rmt/1mn
BR210I Please mount the volume, if you have not already done so.
BR280I Time stamp 1997-09-10 18.57.19
BR256I Please enter 'cont' to continue, 'stop' to cancel the program:
BR257I Your reply: 'cont'
BR259I Program execution will be continued...

BR280I Time stamp 1997-09-10 18.57.21
```


BRRESTORE Logs: Restoring a Complete Backup

```

BR226I Rewinding tape volume in device /dev/rmt/lm...

BR351I Restoring /oracle/C11/sapbackup/.tape.hdr0
BR355I from /dev/rmt/lmn...

BR241I Checking label on volume in device /dev/rmt/lmn

BR209I Volume in device /dev/rmt/lmn has name C11B06 1997-09-10 18.35.15

BR351I Restoring /oracle/C11/sapraw/rawd_1
BR355I from /dev/rmt/lmn...

BR351I Restoring /oracle/C11/sapdata1/system_1/system.data1
BR355I from /dev/rmt/0mn...

#FILE..... /oracle/C11/sapraw/rawd_1
#RESTORED. rawd_1 C11B06/4

BR280I Time stamp 1997-09-10 18.57.57
BR418I 1 of 13 files restored - 10.008 MB of 107.169 MB done.
BR204I Percentage done: 9.34%, estimated end time: 19:03
BR001I *****
_____

BR351I Restoring /oracle/C11/sapdata3/stabd_1/stabd.data1
BR355I from /dev/rmt/lmn...

#FILE..... /oracle/C11/sapdata3/stabd_1/stabd.data1
#RESTORED. stabd.data1 C11B06/5

BR280I Time stamp 1997-09-10 18.58.24
BR418I 2 of 13 files restored - 20.016 MB of 107.169 MB done.
BR204I Percentage done: 18.68%, estimated end time: 19:02
BR001I *****
_____

BR351I Restoring /oracle/C11/sapdata4/stabi_1/stabi.data1
BR355I from /dev/rmt/lmn...

#FILE..... /oracle/C11/sapdata4/stabi_1/stabi.data1
#RESTORED. stabi.data1 C11B06/6

BR280I Time stamp 1997-09-10 18.58.37
BR418I 3 of 13 files restored - 25.023 MB of 107.169 MB done.
BR204I Percentage done: 23.35%, estimated end time: 19:02
BR001I *****
_____

BR351I Restoring /oracle/C11/sapdata2/temp_1/temp.data1
BR355I from /dev/rmt/lmn...

#FILE..... /oracle/C11/sapdata2/temp_1/temp.data1
#RESTORED. temp.data1 C11B06/7

BR280I Time stamp 1997-09-10 18.58.51
BR418I 4 of 13 files restored - 30.031 MB of 107.169 MB done.
BR204I Percentage done: 28.02%, estimated end time: 19:02
BR001I *****
_____

BR351I Restoring /oracle/C11/sapdata3/testd_1/testd.data1
BR355I from /dev/rmt/lmn...

```

BRRESTORE Logs: Restoring a Complete Backup

```
#FILE..... /oracle/C11/sapdata3/testd_1/testd.data1
#RESTORED. testd.data1 C11B06/8

BR280I Time stamp 1997-09-10 18.59.04
BR418I 5 of 13 files restored - 34.039 MB of 107.169 MB done.
BR204I Percentage done: 31.76%, estimated end time: 19:02
BR001I *****
_____

BR351I Restoring /oracle/C11/sapdata4/testi_1/testi.data1
BR355I from /dev/rmt/lmn...

#FILE..... /oracle/C11/sapdata4/testi_1/testi.data1
#RESTORED. testi.data1 C11B06/9

BR280I Time stamp 1997-09-10 18.59.10
BR418I 6 of 13 files restored - 36.047 MB of 107.169 MB done.
BR204I Percentage done: 33.64%, estimated end time: 19:02
BR001I *****
_____

BR351I Restoring /oracle/C11/sapdata1/cntrl/cntrlC11.dbf
BR355I from /dev/rmt/lmn...

#FILE..... /oracle/C11/sapdata1/cntrl/cntrlC11.dbf
#RESTORED. cntrlC11.dbf C11B06/10

BR280I Time stamp 1997-09-10 18.59.11
BR418I 7 of 13 files restored - 36.368 MB of 107.169 MB done.
BR204I Percentage done: 33.94%, estimated end time: 19:02
BR001I *****
_____

BR351I Restoring /oracle/C11/origlogA/log_g11m1.dbf
BR355I from /dev/rmt/lmn...

#FILE..... /oracle/C11/origlogA/log_g11m1.dbf
#RESTORED. log_g11m1.dbf C11B06/11

BR280I Time stamp 1997-09-10 18.59.12
BR418I 8 of 13 files restored - 36.564 MB of 107.169 MB done.
BR204I Percentage done: 34.12%, estimated end time: 19:02
BR001I *****
_____

BR351I Restoring /oracle/C11/origlogB/log_g12m1.dbf
BR355I from /dev/rmt/lmn...

#FILE..... /oracle/C11/origlogB/log_g12m1.dbf
#RESTORED. log_g12m1.dbf C11B06/12

BR280I Time stamp 1997-09-10 18.59.13
BR418I 9 of 13 files restored - 36.761 MB of 107.169 MB done.
BR204I Percentage done: 34.30%, estimated end time: 19:02
BR001I *****
_____

BR351I Restoring /oracle/C11/origlogA/log_g13m1.dbf
BR355I from /dev/rmt/lmn...

#FILE..... /oracle/C11/origlogA/log_g13m1.dbf
#RESTORED. log_g13m1.dbf C11B06/13

BR280I Time stamp 1997-09-10 18.59.14
```

BRRESTORE Logs: Restoring a Complete Backup

BR418I 10 of 13 files restored - 36.957 MB of 107.169 MB done.
 BR204I Percentage done: 34.48%, estimated end time: 19:02
 BR001I *****

BR351I Restoring /oracle/C11/origlogB/log_g14m1.dbf
 BR355I from /dev/rmt/lmn...

#FILE.... /oracle/C11/origlogB/log_g14m1.dbf
 #RESTORED. log_g14m1.dbf C11B06/14

BR280I Time stamp 1997-09-10 18.59.15
 BR418I 11 of 13 files restored - 37.153 MB of 107.169 MB done.
 BR204I Percentage done: 34.67%, estimated end time: 19:02
 BR001I *****

#FILE.... /oracle/C11/sapdata1/system_1/system.data1
 #RESTORED. system.data1 C11B05/4

BR280I Time stamp 1997-09-10 18.59.36
 BR418I 12 of 13 files restored - 87.161 MB of 107.169 MB done.
 BR204I Percentage done: 81.33%, estimated end time: 19:00
 BR001I *****

BR351I Restoring /oracle/C11/sapdata2/roll_1/roll.data1
 BR355I from /dev/rmt/0mn...

#FILE.... /oracle/C11/sapdata2/roll_1/roll.data1
 #RESTORED. roll.data1 C11B05/5

BR280I Time stamp 1997-09-10 19.00.16
 BR418I 13 of 13 files restored - 107.169 MB of 107.169 MB done.
 BR204I Percentage done: 100.00%, estimated end time: 19:00
 BR001I *****

BR351I Restoring /oracle/C11/mirrlogA/log_g11m2.dbf
 BR355I from /oracle/C11/origlogA/log_g11m1.dbf...

BR351I Restoring /oracle/C11/mirrlogB/log_g12m2.dbf
 BR355I from /oracle/C11/origlogB/log_g12m1.dbf...

BR351I Restoring /oracle/C11/mirrlogA/log_g13m2.dbf
 BR355I from /oracle/C11/origlogA/log_g13m1.dbf...

BR351I Restoring /oracle/C11/mirrlogB/log_g14m2.dbf
 BR355I from /oracle/C11/origlogB/log_g14m1.dbf...

BR351I Restoring /oracle/C11/sapdata2/cntrl/cntrlC11.dbf
 BR355I from /oracle/C11/sapdata1/cntrl/cntrlC11.dbf...

BR351I Restoring /oracle/C11/sapdata3/cntrl/cntrlC11.dbf
 BR355I from /oracle/C11/sapdata1/cntrl/cntrlC11.dbf...

BR351I Restoring /oracle/C11/sapdata4/cntrl/cntrlC11.dbf
 BR355I from /oracle/C11/sapdata1/cntrl/cntrlC11.dbf...

BR280I Time stamp 1997-09-10 19.00.20
 BR226I Rewinding tape volume in device /dev/rmt/0m...

BR280I Time stamp 1997-09-10 19.00.20

BRRESTORE Logs: Restoring a Complete Backup

BR226I Rewinding tape volume in device /dev/rmt/lm...

BR406I End of database restore: rcvojvfh rsb 1997-09-10 19.00.47

BR403I BRRESTORE terminated successfully with warnings.

Messages and Return Codes: BRBACKUP, BRARCHIVE, and BRRESTORE

The BRBACKUP, BRARCHIVE, and BRRESTORE programs issue messages in the following form:

BR<NNN><X> Message text

NNN: Sequential number of the message

x: Type of message:

- I: Information
- W: Warning
- E: Error message

The text of the message contains the following variables:

%s: a string

%d: an integer

%f: a real number in floating-point format

In most cases, the SAP utilities issue a group of messages, which will enable you to precisely analyze the error.

A list of the messages appears below (sorted by message number).

[Return Codes \[Page 494\]](#)

[Messages \[Page 495\]](#)

Return Codes for BRBACKUP / BRRESTORE**Return Codes for BRBACKUP / BRRESTORE**

The following return codes are possible:

0	Successful backup/restore.
1	Backup/restore ended with warnings. All files were backed up/restored.
2	Backup/restore canceled during the initialization phase by a user or other signal.
3	Errors occurred during the initialization phase, backup/restore was not started.
4	Backup/restore was canceled by a user or other signal during processing.
5	Backup/restore was started, but not completed because errors occurred during processing.
6	Internal termination of backup/restore.

Messages

[Messages from BR002I to BR028E \[Page 496\]](#)

[Messages from BR029E to BR70W \[Page 501\]](#)

[Messages from BR071E to BR115I \[Page 506\]](#)

[Messages from BR116I to BR151E \[Page 511\]](#)

[Messages from BR152I to BR181E \[Page 516\]](#)

[Messages from BR182E to BR218E \[Page 521\]](#)

[Messages from BR219W to BR241I \[Page 527\]](#)

[Messages from BR242I to BR269W \[Page 532\]](#)

[Messages from BR270W to BR307I \[Page 537\]](#)

[Messages from BR308I to BR339W \[Page 542\]](#)

[Messages from BR340I to BR383E \[Page 547\]](#)

[Messages from BR384E to BR429W \[Page 552\]](#)

[Messages from BR430E to BR464W \[Page 557\]](#)

[Messages from BR465E to BR468I \[Page 561\]](#)

Messages from BR002I to BR028E

Messages from BR002I to BR028E

BR002I BRARCHIVE %s

- *Explanation:* BRARCHIVE version number. This number is independent of the SAP release number.
- *Program response:* None.
- *User action:* None.

BR003I BRARCHIVE terminated successfully.

- *Explanation:* BRARCHIVE ended successfully with return code 0.
- *Program response:* None.
- *User action:* None.

BR004I BRARCHIVE terminated with warnings.

- *Explanation:* BRARCHIVE ended with return code 1. The archiving/ deletion was performed successfully.
- *Program response:* None.
- *User action:* Pay careful attention to the warnings. Analyze them and - if necessary - correct the cause of the warnings.

BR005I BRARCHIVE terminated with errors.

- *Explanation:* BRARCHIVE finished with a return code greater than 1.
- *Program response:* The BRARCHIVE run was terminated.
- *User action:* Analyze and correct the specified errors. Then repeat the archiving run.

BR006I Start of offline redo log processing: %s %s %s

- *Explanation:* BRARCHIVE start message. The action ID (a<encoded timestamp>), the function ID (extension), and the start timestamp (date, time) are listed in sequence to characterize the backup. The action ID and the function ID form the file name of the detail BRARCHIVE log.
- *Program response:* None.
- *User action:* None.

BR007I End of offline redo log processing: %s %s %s

- *Explanation:* BRARCHIVE end message. The action ID, function ID, and end timestamp are issued. See BR006I.
- *Program response:* None.
- *User action:* None.

BR008I Offline redo log processing for database instance/s: %s

- *Explanation:* The name of the database instance(s) whose offline redo log files are to be processed is listed.

- *Program response:* None.
- *User action:* None.

BR009I BRARCHIVE action ID: %s

- *Explanation:* The action ID contains an encoded timestamp that specifies the starting time of BRARCHIVE. This timestamp appears at the beginning of the file name of the detail BRARCHIVE log, after an "a".
- *Program response:* None.
- *User action:* None.

BR010I BRARCHIVE function ID: %s

- *Explanation:* The function ID (extension) indicates the backup mode in which BRARCHIVE ran. It appears second in the file name of the detail BRARCHIVE log. See [Names of the Detail Logs \[Page 446\]](#).
- *Program response:* None.
- *User action:* None.

BR011I %d offline redo log files found for processing, total size %4.3f MB.

- *Explanation:* The number of offline redo log files selected for processing by BRARCHIVE is displayed.
- *Program response:* None.
- *User action:* None.

BR012I %d next offline redo log files found for processing, total size %4.3f MB.

- *Explanation:* Once all the offline redo log files in the archiving directory have been processed, BRARCHIVE checks whether ORACLE has written new offline redo log files into the archiving directory in the meantime. If so, their number is displayed in this message, and processing is continued. This case may occur during a BRARCHIVE run with options `-s`, `-sd`, `-ss`, or `-ssd`, and of course during a run with option `-f`.
- *Program response:* If offline redo log files are found, BRARCHIVE archives them.
- *User action:* None.

BR013W No offline redo log files found for processing.

- *Explanation:* No suitable offline redo logs were found in the archiving directory for the selected BRARCHIVE function.
- *Program response:* BRARCHIVE is terminated.
- *User action:* None.

BR014I %d of %d offline redo log files processed - %4.3f MB of %4.3f MB done.

- *Explanation:* The current status of the processing of the offline redo log files is displayed.
- *Program response:* None.

Messages from BR002I to BR028E

- *User action:* None.

BR015I Offline redo log file %s deleted.

- *Explanation:* Confirms the deletion of an offline redo log file.
- *Program response:* None.
- *User action:* None.

BR016I %d Offline redo log files processed.

- *Explanation:* The total number of processed offline redo log files is displayed when the BRARCHIVE run is finished.
- *Program response:* None.
- *User action:* None.

BR017W Offline redo log file %s not found.

- *Explanation:* This warning informs you that gaps are present in the archiving directory. The database cannot be recovered in such cases if the recovery process is started at an offline redo log file with a smaller log sequence number. Possible causes:

Case 1: Offline redo log files were deleted using operating system utilities before they could be archived, or ORACLE ARCHIVELOG mode was temporarily deactivated. BRARCHIVE was started with option `-s`, `-sd`, `-ss`, or `-ssd`.

Case 2: Offline redo log files were deleted using operating system utilities after they were archived once by BRARCHIVE, but not yet deleted by BRARCHIVE. BRARCHIVE was started with option `-sc`, `-scd`, `-ds`, or `-dc`.

- *Program response:* The offline redo log file is ignored.
- *User action:* Case 1: SAP recommends a full backup of the database. The warning is normally only issued once. After at least one offline redo log file with a larger log sequence number is archived, the message is suppressed automatically, and no manual intervention is required.

Case 2: The warning is issued repeatedly when BRARCHIVE is started with option `-sc`, `-scd`, `-ds`, or `-dc`. To suppress the message, you have to edit the summary BRARCHIVE log `<ORACLE_HOME>/saparch/arch<DBSID>.log`. In the `#COPIED` and `#DELETED` lines, which refer to the affected offline redo log files, replace the first character, `#`, with `+` or `*`.

Example:

Display before:

```
#ARCHIVE.. 622 /oracle/C11/saparch/C11arch1_622 1995-03-03
03.36.47
#SAVED.... acovzgvj ssd #C11A34/13 1995-03-03 08.05.18
#COPIED.....
#DELETED.....
```

Display after:

```
#ARCHIVE.. 622 /oracle/C11/saparch/C11arch1_622 1995-03-03
03.36.47
#SAVED.... acovzgvj ssd #C11A34/13 1995-03-03 08.05.18
```

```
+COPIED.....
+DELETED.....
```

BR018I Waiting for the next offline redo log file...

- *Explanation:* When you select option `-f` in the BRARCHIVE command, BRARCHIVE waits for the next offline redo log file and immediately archives it in the archiving directory when it appears. This message is also displayed when the offline redo log files were processed successfully and a new file is being added to the archiving directory. See BR012I.
- *Program response:* Program continues as soon as new offline redo log files appear in the archiving directory.
- *User action:* None. If necessary, you can interrupt the BRARCHIVE run with option `-f stop`. See [Canceling a Backup \[Page 226\]](#), and [-f|-fill \[Page 396\]](#).

BR019I Volume size limit reached - no more offline redo log files will be processed.

- *Explanation:* BRARCHIVE can only use one volume for the requested archiving - it does not support continuation tapes. The number of selected offline redo log files depends on the parameter `tape_size` (see [Parameter tape_size \[Ext.\]](#)).
- *Program response:* BRARCHIVE processes the selected offline redo log files and shuts down automatically. Several redo log files remain in the archiving directory.
- *User action:* The remaining offline redo log files will be archived during the next BRARCHIVE run. If you want to save them now, you will have to start BRARCHIVE again.

BR020E BRARCHIVE already running or was killed.

- *Explanation:* You tried to start an archiving process for a specific database.
- *Program response:* BRARCHIVE is not started.
- *User action:* When another BRARCHIVE process is running, you have to wait until it is finished. If a BRARCHIVE run was canceled or terminated, see the other messages.

BR021I Please delete the file %s if BRARCHIVE was killed.

- *Explanation:* This message appears together with BR020E.
- *Program response:* BRARCHIVE is not started.
- *User action:* Check whether BRARCHIVE is running (e.g. using the UNIX command `ps`). If not, delete the specified file and start BRARCHIVE again.

BR022I Setting BRARCHIVE lock failed.

- *Explanation:* This message appears together with BR020E.
- *Program response:* BRARCHIVE processing terminates because the lock cannot be set.
- *User action:* See BR020E and BR021I.

BR025E Not enough backup devices for this function.

- *Explanation:* You used option `-ss` or `-ssd` (parallel archiving on two backup devices), but have only specified one backup device `init<DBSID>.sap` parameter file.

Messages from BR002I to BR028E

- *Program response:* The BRARCHIVE run is not performed.
- *User action:* Make sure at least two backup devices are used for archiving. More devices can also be specified, but only two are used. See [tape address \[Page 347\]](#). Restart BRARCHIVE.

BR026I BRARCHIVE started with option %s will be stopped now.

- *Explanation:* A brarchive -f run was stopped with brarchive -f stop.
- *Program response:* BRARCHIVE terminates regularly.
- *User action:* None.

BR027I BRARCHIVE started with option %s has been stopped.

- *Explanation:* This message confirms that a BRARCHIVE run started with brarchive -f was stopped successfully.
- *Program response:* None.
- *User action:* None.

BR028E BRARCHIVE started with option %s is not running.

- *Explanation:* The BRARCHIVE run could not be stopped because brarchive -f is not running.
- *Program response:* Program call is not performed.
- *User action:* None.

Messages from BR029E to BR70W

BR029E Stopping BRARCHIVE started with option %s failed.

- *Explanation:* The BRARCHIVE run started with `brarchive -f` could not be stopped.
- *Program response:* `brarchive -f stop` is canceled
- *User action:* Note the other messages. Try to correct the error(s), then repeat the process.

BR030W BRARCHIVE log %s not found - volume management reset.

- *Explanation:* The summary BRARCHIVE log `<ORACLE_HOME>/saparch/arch<DBSID>.log` forms the foundation for volume management. If it no longer exists, BRARCHIVE loses the information about the volumes (tapes) used.
- *Program response:* BRARCHIVE requests the first volume from pool `volume_archive` and continues processing.
- *User action:* If you do not want this response, cancel BRARCHIVE and restore the summary log. The summary log from the last BRARCHIVE backup volume can be imported with the call `brrestore -n sum_log`.

BR031W Redo log sequence number was reset for database instance %s.

- *Explanation:* BRARCHIVE has detected that the current log sequence number is smaller than the log sequence number of the last archived offline redo log file. This can occur, for example, when the database instance was opened with `ALTER DATABASE OPEN RESETLOGS` after a recovery.
- *Program response:* Processing is continued.
- *User action:* None.

BR032I %d offline redo log file/s found from database instance %s.

- *Explanation:* The number of offline redo log files selected by BRARCHIVE for a database instance is output.
- *Program response:* None.
- *User action:* None.

BR033E Archive directory %s not found.

- *Explanation:* The archiving directory defined in profile `init<DBSID>.ora` in `log_archive_dest` does not exist.
- *Program response:* The program is terminated because it cannot find any offline redo log files.
- *User action:* Correct the parameters or create the directory. Then repeat the process.

BR051I BRBACKUP %s

- *Explanation:* BRBACKUP version number. This number is independent of the SAP release number.

Messages from BR029E to BR70W

- *Program response:* None.
- *User action:* None.

BR052I BRBACKUP terminated successfully

- *Explanation:* BRBACKUP terminated successfully, with return code 0.
- *Program response:* None.
- *User action:* None.

BR053I BRBACKUP terminated with warnings.

- *Explanation:* BRBACKUP terminated with return code 1. The backup was performed successfully.
- *Program response:* None.
- *User action:* Pay careful attention to the warnings. Analyze them and - if necessary - correct the cause of the warnings.

BR054I BRBACKUP terminated with errors.

- *Explanation:* BRARCHIVE finished with a return code greater than 1.
- *Program response:* The BRBACKUP run was terminated.
- *User action:* Analyze and correct the specified errors. Then repeat the backup.

BR055I Start of database backup: %s %s %s

- *Explanation:* BRBACKUP start message. The action ID (b<encoded timestamp>), the function ID (extension), and the start timestamp (date, time) are listed in that order, to characterize the backup. The action ID and the function ID form the file name of the detail BRBACKUP log.
- *Program response:* None.
- *User action:* None.

BR056I End of database backup: %s %s %s

- *Explanation:* BRBACKUP end message. The action ID, function ID, and end timestamp are output. See BR055I.
- *Program response:* None.
- *User action:* None.

BR057I Backup of database: %s

- *Explanation:* The name of the database for backup is displayed.
- *Program response:* None.
- *User action:* None.

BR058I BRBACKUP action ID: %s

- *Explanation:* The action ID contains an encoded timestamp that specifies the starting time of BRBACKUP. It appears at the start of the file name of the detail BRBACKUP log, after a "b".
- *Program response:* None.

- *User action:* None.

BR059I BRBACKUP function ID: %s

- *Explanation:* The function ID (extension) indicates the backup mode in which the BRBACKUP run was performed. It appears second in the file name of the detail BRBACKUP log. See [Names of the Detail Logs \[Page 439\]](#).
- *Program response:* None.
- *User action:* None.

BR060I Tablespaces for backup: %s

- *Explanation:* List of the tablespaces for backup. If `all` is specified, all the tablespaces are backed up. If you perform an offline backup, all the online redo log files are also backed up.
- *Program response:* None.
- *User action:* None.

BR061I %d files found for backup, total size %4.3f MB.

- *Explanation:* The total number and total size of files for backup is displayed.
- *Program response:* None.
- *User action:* None.

BR062I %d files/directories found for backup, total size %4.3f MB.

- *Explanation:* The total number and total size of the files for backup is displayed. This message is issued when you also want to back up directories.
- *Program response:* None.
- *User action:* None.

BR063I %d of %d files processed - %4.3f MB of %4.3f MB done.

- *Explanation:* The current status of the database backup is displayed.
- *Program response:* None.
- *User action:* None.

BR064I The database instance %s will be shut down now.

- *Explanation:* During an offline backup, the database instance is shut down once the BRBACKUP initialization phase is complete.
- *Program response:* None.
- *User action:* If you are not working in unattended mode, you can terminate the backup at this point.

BR065E Tablespace %s not found.

- *Explanation:* This error message is issued when a tablespace you specified (using the BRBACKUP option `-m <tablespace name>` or by defining the tablespace name in parameter `backup_mode` of the `init.sap` profile) is not defined in the database.
- *Program response:* The BRBACKUP run is terminated.
- *User action:* Check the tablespace name and start BRBACKUP again.

Messages from BR029E to BR70W

BR066E For online backup, the database must be in ARCHIVELOG mode.

- *Explanation:* An online backup only possible when ARCHIVELOG mode is active. This ensures that the online redo log files are archived in the archiving directory before they are overwritten during a redo log file switch. The redo logs are required for any necessary recovery.
- *Program response:* No BRBACKUP run possible.
- *User action:* Start the database system with the ARCHIVELOG mode active..

BR067E For online backup, automatic archival must be ENABLED for all database instances.

- *Explanation:* An online backup can only be performed when automatic archiving is active - that is, ORACLE background process ARCH (for archiving the online redo log files to the archiving directory during a redo log file switch) must be active.
- *Program response:* No BRBACKUP run possible.
- *User action:* Activate automatic archiving of redo log files and repeat the operation. See [Archiving Parameters \[Page 34\]](#).

BR068E SAP system is running or SAPR3 user is connected at offline backup.

- *Explanation:* This error message appears when an SAP System is still running (or an SAPR3 user is still logged on to the database via SQLPLUS/SQLDBA) and you tried to start an offline backup. Transactions and jobs in the SAP System would be terminated when the database instance was shut down. BRBACKUP prevents this.
- *Program response:* BRBACKUP terminates.
- *User action:* Shut down the SAP System (with `stopsap`, for example) and then start BRBACKUP again. You can also set the `init<DBSID>.ora` parameter `backup_type` to `offline_force`. The database instance will then be shut down when BRBACKUP is started regardless of whether the SAP System is running or not

BR069E File name %s not unique - backup/verification impossible.

- *Explanation:* Because the file name is not unique, files in the backup directory (during a backup to disk) could be overwritten.
- *Program response:* The BRBACKUP run is terminated.
- *User action:* Shut down the SAP System. Use the following SQLDBA to rename the file:

```
connect internal
shutdown immediate
startup mount
alter database rename '<file name>' to '<file name>';
alter database open;
Then start BRBACKUP again.
```


Messages from BR029E to BR70W

BR070W File %s greater than volume size - trying to back it up, anyway.

- *Explanation:* The file may not fit on the tape. The actual tape switch is triggered by `cpio` when the physical end of the tape has been reached.
- *Program response:* BRBACKUP attempts to save the file on a volume individually.
- *User action:* When the physical end of the tape is reached, mount a continuation tape in the tape device. Please note that BRBACKUP considers both tapes to be one logical tape. It is the user's responsibility to provide these two volumes, in the correct sequence, even though BRBACKUP only requests one tape. SAP recommends that you not define any files in the database that are larger than the physical tape size. See [cpio Continuation Tape \[Page 224\]](#).

Messages from BR071E to BR115I

Messages from BR071E to BR115I

BR071E BRBACKUP currently running or was killed.

- *Explanation:* Only one backup process should be started for each specific database.
- *Program response:* BRBACKUP is not started.
- *User action:* When another BRBACKUP process is running, you have to wait until it is finished. If a BRBACKUP run was canceled, refer to the other messages.

BR072I Please delete file %s if BRBACKUP was killed.

- *Explanation:* This message appears together with BR071E.
- *Program response:* BRBACKUP is not started.
- *User action:* Check whether BRBACKUP is running (e.g. using the UNIX command `ps`). If not, delete the specified file and start BRBACKUP again.

BR073I Setting BRBACKUP lock failed.

- *Explanation:* This message appears together with BR071E.
- *Program response:* BRBACKUP processing is canceled because the lock cannot be set.
- *User action:* See BR071E and BR072I.

BR074W BRCONNECT call failed.

BR074E BRCONNECT call failed.

- *Explanation:* The BRCONNECT program is started when you start a backup with BRBACKUP, in order to monitor the database status.
- *Program response:* BRBACKUP is continued (BR074W) or canceled (BR074E).
- *User action:* Analyze the other error messages that were issued. Try to correct the error(s). If BRBACKUP was canceled, repeat the operation.

BR077I Database files for backup: %s

- *Explanation:* The online redo log files for backup, the control file, and other database files that were specified individually using parameter `backup_mode` or option `-m` (not via the tablespace name) are displayed.
- *Program response:* None.
- *User action:* None.

BR078I Non-database files for backup: %s

- *Explanation:* The non-database files for backup are displayed.
- *Program response:* None.
- *User action:* None.

BR079I Directories for backup: %s

- *Explanation:* The directories for backup are displayed.

- *Program response:* None.
- *User action:* None.

BR080E File %s to be backed up not found.

- *Explanation:* The file for backup does not exist.
- *Program response:* Termination of the BRBACKUP run.
- *User action:* Correct the `backup_mode` parameter or use option `-m` to specify the correct file name. Then repeat the operation.

BR081W Directories will not be compressed.

- *Explanation:* Software compression is activated (`compress = yes` or option `-k yes`). However, the contents of the directories will not be compressed.
- *Program response:* Directory contents are not compressed.
- *User action:* None. This is a limitation of BRBACKUP.

BR082W Backups of directories will not be verified.

- *Explanation:* Backup verification is activated (option `-w|-verify`). However, the backed up contents of directories will not be verified.
- *Program response:* Backed up directory contents are not verified.
- *User action:* None. This is a limitation of BRBACKUP.

BR083W Compression rates out of date - run BRBACKUP with option '-k only'.

- *Explanation:* The last simulated backup with compression for determining the compression rates (`compress = only` or option `-k only`) was started more than a year ago.

This is a warning that is only issued when hardware compression is activated (`compress = hardware` or option `-k hardware`). This message is also displayed the first time BRBACKUP Version 3.0 is started, regardless of when the last simulated backup with compression took place.

- *Program response:* The old compression rates are accepted or - if none are available - the default values are used (see [Log Supplements \[Page 442\]](#)).
- *User action:* You can continue processing. Regularly start the routine for determining the compression rate (`compress = only` or option `-k only`), to enable BRBACKUP to calculate the current compression rates. See [Hardware Compression \[Page 232\]](#).

BR100E Internal error for '%s' at location %s-%d.

- *Explanation:* Internal error during processing.
- *Program response:* Processing terminated.
- *User action:* Please contact SAP.

BR101I Parameters

Messages from BR071E to BR115I

- *Explanation:* The current values of the parameters relevant for the backup/restore operation are displayed.
- *Program response:* None.
- *User action:* None.

BR102I Following backup devices will be used: %s

- *Explanation:* The backup devices (in accordance with the value of the init.sap profile parameter `tape_address`) that will be used for backup are displayed.
- *Program response:* None.
- *User action:* None.

BR103I The following backup volumes will be used: %s

- *Explanation:* The name of the volume - selected by automatic volume management or with option `-v|-volume` - that will be used for the backup is displayed.
- *Program response:* None.
- *User action:* None.

BR104I %d scratch volume will be used.

- *Explanation:* The number of the volume required for the backup is displayed. You used option `-v` or one of the init.sap parameters `volume_backup` and `volume_archive` with the reserved word `SCRATCH`. This deactivated the automatic volume management. This means that the program only checks the expiration period for the mounted backup volume(s); any name is possible. To find out the actual name of the volume used, see message BR209I.
- *Program response:* Any volume with an expired expiration period is accepted.
- *User action:* Provide any volume with an expired expiration period.

BR105E Only %d free backup volumes found, %d required.

- *Explanation:* Free volumes (with an expired expiration period) are sought in the list of backup volumes specified in profile init.sap. Fewer volumes were found than are required.
- *Program response:* Processing terminated.
- *User action:* Either add the required number of valid volumes to the list in the profile, or use the command option `-v SCRATCH` when you start the program. Repeat the operation. The problem can also be corrected by reducing the expiration period (parameter `expir_period` in profile `init<DBSID>.sap`). Observe the SAP recommendations for the expiration period, see [Volume Expiration Period \[Page 147\]](#).

BR106I Files will be saved on disk in directories: %s

- *Explanation:* During a backup to disk, only the directory names (in accordance with init.sap parameter `backup_root_dir`) that will be used for the backup are displayed.
- *Program response:* None.
- *User action:* None.

BR107I Remote backup device will be accessed through pipe.

Messages from BR071E to BR115I

- *Explanation:* Confirms this parameter `backup_dev_type` or option `-d|-device` is set to `pipe` or `pipe_auto`.
- *Program response:* The backup is performed on a remote system defined in the `init.sap` parameter `read_fifo_cmd`.
- *User action:* None.

BR108I Automatic volume remount active.

- *Explanation:* BRBACKUP was called with the `init<DBSID>.sap` parameter `backup_dev_type=tape_auto|pipe_auto` (or option `-d tape_auto|pipe_auto`).
- *Program response:* None.
- *User action:* No operator action is necessary during a volume switch when tape devices with automatic tape changing are used.

BR109I Files will be saved by backup utility at file level.

- *Explanation:* Confirms that the backup will be performed at file level using an external backup program. Parameter `backup_dev_type = util_file` or option `-d util_file` was set.
- *Program response:* None.
- *User action:* None.

BR111I Files will be compressed.

- *Explanation:* Confirms that the software compression configured with parameter `compress = yes` or option `-k yes` is active.
- *Program response:* None.
- *User action:* None.

BR112I Files will not be compressed.

- *Explanation:* Confirms that files will not be compressed during the backup. Parameter `compress = no` or option `-k no` was set.
- *Program response:* None.
- *User action:* None.

BR113I Files will be compressed by hardware.

- *Explanation:* Confirms that tape devices with hardware compression are being used, and that BRBACKUP will take this fact into account. Parameter `compress = hardware` or option `-k hardware` was set. See [Tape Devices with Hardware Compression \[Page 115\]](#).
- *Program response:* None.
- *User action:* None.

BR114I Compression will not be performed for backup utility.

- *Explanation:* The requested compression will not be performed when the backup is performed using an external backup program.

Messages from BR071E to BR115I

- *Program response:* No compression is performed.
- *User action:* Set the init.sap parameter `compress = no` or use the command option `-k no`.

BR115I Compression rate for all files %5.4f:1

- *Explanation:* The average compression rate is calculated as the total size of the files before compression divided by the total size of the files after compression.
- *Program response:* None.
- *User action:* None.

Messages from BR116I to BR151E

BR116I %s before backup for database instance %s

- *Explanation:* Records that command ARCHIVE LOG LIST was issued by SQLDBA before the backup was started.
- *Program response:* None.
- *User action:* None.

BR117I %s after backup for database instance %s

- *Explanation:* Records that command ARCHIVE LOG LIST was issued by SQLDBA after the backup was completed.
- *Program response:* None.
- *User action:* None.

BR118I Tablespaces and data files

- *Explanation:* A description of all tablespaces and their database files is displayed. See [Detail Log \[Page 440\]](#).
- *Program response:* None.
- *User action:* None.

BR119I Redo log files

- *Explanation:* A description of all online redo log files in the database is output. See [Detail Log \[Page 440\]](#).
- *Program response:* None.
- *User action:* None.

BR120I Control files

- *Explanation:* A description of all control files in the database is displayed. See [Detail Log \[Page 440\]](#).
- *Program response:* None.
- *User action:* None.

BR121W Processing of the log file %s failed.

BR121E Processing of the log file %s failed.

- *Explanation:* The BRARCHIVE/BRBACKUP log files can only be processed when they correspond to the SAP standard.
- *Program response:* The message is ignored (BR121W) or processing is terminated (BR121E).
- *User action:* Note the additional messages and check the log information. If you have manually changed a log file, reset your change. If a termination occurred, restart the operation.

Messages from BR116I to BR151E

BR122E Line %d in %s longer than %d characters.

- *Explanation:* This message appears together with BR121X. A line in a BRARCHIVE/BRBACKUP log file is too long.
- *Program response:* Processing terminated.
- *User action:* Correct the log. Then restart the operation.

BR123E Format error in line %d in %s found.

- *Explanation:* This message appears together with BR121X. A line in a BRARCHIVE/BRBACKUP log file has an incorrect format.
- *Program response:* Processing terminated.
- *User action:* Correct the log. Then restart the operation.

BR124I Query option set - no processing.

- *Explanation:* The program was called with option `-q`.
- *Program response:* The required resources are displayed, but no backup is started.
- *User action:* If you want to start an actual backup now, mount the requested volume and make any other requested resources available.

BR125I Compression only - no backup.

- *Explanation:* The program was started with parameter `compress = only` or with option `-k only`. The program simulates a backup with compression, but no backup is started. The compression rates can be determined in this manner.
- *Program response:* None.
- *User action:* None.

BR126I Unattended mode active - no confirmation required

- *Explanation:* The SAP utility was started with option `-c`. See [-c|confirm \[Page 369\]](#).
- *Program response:* The first confirmations and mount requests are suppressed. Remount prompts are displayed.
- *User action:* None.

BR127I Fill volume option set - waiting for offline redo log files possible.

- *Explanation:* BRARCHIVE was called with option `-f`. Any new offline redo logs saved to the archiving directory are archived immediately.
- *Program response:* If all the redo logs to this point have been processed, BRARCHIVE waits for the next offline redo log files.
- *User action:* None. To cancel this BRARCHIVE run use option `-f stop`. See [-f|fill \[Page 396\]](#).

BR128W Option '%s' ignored for '%s'.

- *Explanation:* If two contradictory options (option and parameter) are entered, one is given priority.
- *Program response:* The option is ignored.

Messages from BR116I to BR151E

- *User action:* During the next call, please do not use the option that was ignored this time.

BR129W Force option set - no label checking during initialization.

- *Explanation:* The SAP utility was called with option `-i force`. See [Initializing Volumes \[Page 143\]](#). Caution! No label check is performed during volume initialization with option `-i force`. This means that a volume that is still locked could be overwritten.
- *Program response:* No label check.
- *User action:* None.

BR130I Backup device type: %s.

- *Explanation:* The value of init.sap parameter `backup_dev_type` or option `-d|-device` is displayed.
- *Program response:* None.
- *User action:* None.

BR131I All volumes get the name SCRATCH.

- *Explanation:* During volume initialization, parameter `volume_archive/volume_backup` or option `-v|-volume` was set to SCRATCH. This solution is not normally recommended, because the volume names are not unique. This may cause problems when reloading such volumes. See [Scratch Volumes \[Page 146\]](#).
- *Program response:* All volumes are assigned the name SCRATCH.
- *User action:* None.

BR132I Show option set - volume label will be displayed without initialization.

- *Explanation:* The SAP utility was called with option `-i show`. The volume labels are displayed.
- *Program response:* Volume is not initialized, only the label is displayed.
- *User action:* None.

BR133I Label of volume in device %s:

- *Explanation:* The contents of the volume label are displayed.
- *Program response:* None.
- *User action:* None.

BR134I Unattended mode with 'force' active - no confirmation allowed.

- *Explanation:* The program was started with option `-c force`. See [-c|-confirm \[Page 369\]](#).
- *Program response:* All interaction with the operator and cpio is suppressed (exception: password entry).
- *User action:* None.

BR135I Query with check option set - no processing.

Messages from BR116I to BR151E

- *Explanation:* The program was called with option `-q check`. The required resources are displayed. The system checks that the correct volumes are mounted in the backup devices. No backup is started.
- *Program response:* If incorrect volume are mounted in the devices, these volumes are rejected.
- *User action:* None. Use this option, for example, if you want to use CRON to schedule backups. This enables you to ensure that the right volumes have been mounted.

BR136I Verify option set - double the work to do.

- *Explanation:* The SAP utility was called with option `-w|-verify`. See [-w|-verify \[Page 389\]](#) for BRBACKUP or [-w|-verify \[Page 412\]](#) for BRARCHIVE
- *Program response:* The saved files are read again and compared with the original data.
- *User action:* Using this option roughly doubles the backup time. Therefore, allow more time for the backup.

BR137W Verification will not be performed for backup utility.

- *Explanation:* The option `-w|-verify` is ignored when the backup is performed using an external backup program.
- *Program response:* The backup is performed without a verification run.
- *User action:* None. The next time you start a backup using an external backup program, do not use this option.

BR138I Non-database files

- *Explanation:* A description of the non-database files for backup is displayed. See [Detail Log \[Page 440\]](#).
- *Program response:* None.
- *User action:* None.

BR139I Directories

- *Explanation:* A description of the directory contents for backup is displayed. See [Detail Log \[Page 440\]](#).
- *Program response:* None.
- *User action:* None.

BR140I Number of parallel copy processes: %d.

- *Explanation:* The number of copy processes to be started by BRBACKUP is output.
- *Program response:* None.
- *User action:* None.

BR142I Files will be switched to backup status during the backup.

- *Explanation:* Parameter `backup_dev_type` or option `-d` was set to `util_file_online`. Therefore, the tablespaces are dynamically reset to backup status during the backup.
- *Program response:* None.

- *User action:* None.

BR143I Backup type: %s.

- *Explanation:* The backup type (defined by parameter `backup_type` or option `-t | -type`) is output.
- *Program response:* None.
- *User action:* None.

BR151E Environment variable %s should not be set.

- *Explanation:* Generally, this message appears when the `TWO_TASK` environment variable is set. The SAP utilities should only run on the database server.
- *Program response:* The utility terminates.
- *User action:* Display the environment variables and delete the variable. Then restart the appropriate.

Messages from BR152I to BR181E

Messages from BR152I to BR181E

BR152I Environment variable %s not set.

- *Explanation:* This message indicates that an environment variable, e.g. ORACLE_HOME or ORACLE_SID, has not been set.
- *Program response:* Processing is terminated.
- *User action:* Set the appropriate environment variable and repeat the operation.

BR153E Unknown option '%s' found at position %d.

- *Explanation:* Call error. The option used is unknown.
- *Program response:* Processing terminated.
- *User action:* Check the command syntax (e.g. using option -h) and repeat the operation.

BR154E Unexpected option value '%s' found at position %d.

- *Explanation:* Call error. The option value is not allowed in this position.
- *Program response:* Processing terminated.
- *User action:* Check the command syntax (e.g. using option -h) and repeat the operation.

BR155E No value found for option %s.

- *Explanation:* Call error. An option value is missing.
- *Program response:* Processing terminated.
- *User action:* Check the command syntax (e.g. using option -h) and repeat the operation.

BR156E Illegal value '%s' found for option '%s'.

- *Explanation:* Call error. The option value is illegal.
- *Program response:* Processing terminated.
- *User action:* Check the command syntax (e.g. using option -h) and repeat the operation.

BR157E Value '%s' for option %s longer than %d characters.

- *Explanation:* Call error. The option value is too long.
- *Program response:* Processing terminated.
- *User action:* Check the command syntax (e.g. using option -h) and repeat the operation.

BR158E Error reading init_ora profile %s

- *Explanation:* Errors were determined reading the init<DBSID>.ora profile.
- *Program response:* Processing terminated.
- *User action:* Check the syntax of the information in the profile init<DBSID>.ora. Examine the detailed error messages listed there. Then restart the operation.

BR159E Error reading init_sap profile %s

Messages from BR152I to BR181E

- *Explanation:* Errors were determined reading the `init<DBSID>.ora` profile.
- *Program response:* Processing terminated.
- *User action:* Check the syntax of the information in the profile `init<DBSID>.ora`. Examine the detailed error messages listed there. Then restart the operation.

BR160E Unexpected symbol '%s' found for parameter %s.

- *Explanation:* A syntax error was found when reading a parameter.
- *Program response:* Processing terminated.
- *User action:* Correct the syntax. See [Initialization Profile init<DBSID>.sap \[Page 306\]](#).

BR161E More data expected in file %s

- *Explanation:* Syntax error in the specified file. The definition of the parameter was not completed properly.
- *Program response:* Processing terminated.
- *User action:* Correct the syntax. See [Initialization Profile init<DBSID>.sap \[Page 306\]](#).

BR162E Parameter %s in file %s not supported.

- *Explanation:* Profile error. The specified parameter should not be used. Often involves a not-supported `init.ora` parameter.
- *Program response:* Processing terminated
- *User action:* Delete the parameter, restart the database and repeat the action.

BR163E Unknown parameter %s found in file %s

- *Explanation:* Profile error. The parameter used is unknown.
- *Program response:* Processing terminated.
- *User action:* Check your entry. Also see BR162W

BR164E Illegal value '%s' found for parameter %s.

- *Explanation:* Profile error. The parameter value is illegal.
- *Program response:* Processing terminated.
- *User action:* Check the parameter definition.

BR165E Parameter %s is not set in %s

- *Explanation:* Profile error. A required parameter was not defined.
- *Program response:* Processing terminated.
- *User action:* Check the profile. Compare the specifications, e.g. with the sample profile `initSID.sap`.

BR166W Parameter %s not found in file %s - default assumed.

- *Explanation:* Because a parameter was not found, the ORACLE default value is selected. The default value can be platform-specific.
- *Program response:* The ORACLE default value is assumed.

Messages from BR152I to BR181E

- *User action:* Normally none. If you want to suppress this warning, explicitly define the parameter.

BR167E Unexpected value '%s' found for parameter %s.

- *Explanation:* Profile error. Only one value is accepted for this parameter, not a value list.
- *Program response:* Processing terminated.
- *User action:* Correct the parameter definition.

BR168E Please define only one value for parameter %s.

- *Explanation:* Define only one value for this parameter, not a value list.
- *Program response:* Processing terminated.
- *User action:* Correct the parameter definition.

BR169W Value '%s' of parameter %s ignored for '%s' - '%s' assumed.

- *Explanation:* The parameter value is illegal for this call constellation.
- *Program response:* The value is replaced by a different value and processing is continued.
- *User action:* Do not use this value in future calls of the program.

BR170E Value of parameter %s should be enclosed in double quotes.

- *Explanation:* Profile error. Double quotes are missing.
- *Program response:* Processing terminated.
- *User action:* Enclose the value of this parameter in double quotes. BRBACKUP/BRARCHIVE uses this syntax in the `init<DBSID>.sap` profile. Make sure you adhere to this syntax (compare to e.g. the sample profile `initSID.sap`).

BR171E Value '%s' of parameter %s longer than %d characters.

- *Explanation:* The parameter value is too long.
- *Program response:* Processing terminated.
- *User action:* Correct the parameter definition.

BR172E Root directories %s and %s on same disk volume.

- *Explanation:* If you specify several directories in profile parameter `backup_root_dir`, these directories should be on different disks.
- *Program response:* Processing terminated.
- *User action:* Correct the information in the `init.sap` profile.

BR173E Option %s missing in parameter %s.

- *Explanation:* The specified option is missing. This message normally refers to option `-c` of parameter `compress_cmd`. SAP recommends always using option `-c` of the compress command in the `init<DBSID>.sap` parameter `compress_cmd`, to ensure that the original file is not deleted during compression.
- *Program response:* Processing terminated.

- *User action:* Correct the parameter definition.

BR174E Dollar character missing.

- *Explanation:* The parameter syntax requires one or more \$ characters in the parameter value as placeholders. This message appears together with BR164E, in which the parameter name is displayed.
- *Program response:* Processing terminated.
- *User action:* Correct the parameter definition.

BR175E Character 'K' or 'M' should directly follow volume size.

- *Explanation:* The value of parameter `tape_size` must end with K or M as the last character. This message appears together with BR164E, in which the parameter name is displayed.
- *Program response:* Processing terminated.
- *User action:* Correct the parameter definitions.

BR176E Parameter %s may not begin with a pipe character.

- *Explanation:* The init.sap parameter `read_fifo_cmd` does not require a "|" character at the beginning of the parameter. BRBACKUP/BRARCHIVE uses the syntax specified in sample profile `initSID.sap` for this parameter.
- *Program response:* Processing terminated.
- *User action:* Correct the parameter value.

BR177E Parameter %s may not include the dd command.

- *Explanation:* The init.sap parameter `read_fifo_cmd` does not require a dd command within the parameter information. BRBACKUP/BRARCHIVE uses the syntax specified in sample profile `initSID.sap` for this parameter.
- *Program response:* Processing terminated.
- *User action:* Correct the parameter value.

BR178E Please use %s and %s parameters to define dd command.

- *Explanation:* The dd command is defined in the specified parameters. The two parameters `copy_in_cmd` and `copy_out_cmd` control the backup to a remote system. Only the host name and (optionally) the user name is specified in parameter `read_fifo_cmd`.
- *Program response:* Processing terminated.
- *User action:* Correct the parameter value.

BR179E Number of backup devices with and without rewind is not the same.

- *Explanation:* The number of the device addresses in both parameters must agree.
- *Program response:* Processing terminated.

Messages from BR152I to BR181E

- *User action:* Check the values for parameters `tape_address` and `tape_address_rew`. Make sure that the number of device addresses in both parameters agrees.

BR180E Backup devices with and without rewind have the same address.

- *Explanation:* The names of the device addresses in both parameters must be different.
- *Program response:* Processing terminated.
- *User action:* Check the values for parameters `tape_address` and `tape_address_rew`. Make sure that the device addresses are different. For no-rewind addresses, the only difference to the rewind address is often an “n” in the name. See [tape_address \[Page 347\]](#).

BR181W Option '%s' not supported for '%s'.

BR181E Option '%s' not supported for '%s'.

- *Explanation:* The option is not supported in the call constellation.
- *Program response:* The option is ignored and processing is continued (BR181W). Processing is terminated (BR181E).
- *User action:* Check the command syntax (e.g. using option `-h`). Repeat the operation if a termination occurred.

Messages from BR182E to BR218E

BR182E Checking parameter/option %s failed

- *Explanation:* Errors were discovered when checking a parameter. The cause of the error is described in additional messages.
- *Program response:* Processing terminates
- *User action:* Correct the parameter definition and repeat the process.

BR183E Values '%s' of option '%s' not ascending

- *Explanation:* The values of the option must be defined in ascending order.
 - *Program response:* Processing terminates
- *User action:* Correct the parameter and repeat the process.

BR184E Maximal %d values can be defined for parameter %s

- *Explanation:* The number of values for the parameter exceeds the maximum number permitted.
- *Program response:* Processing terminates
 - *User action:* Correct the parameter and repeat the process.

BR185E File %s defined in IFILE parameter not in %s

- *Explanation:* The file defined in the `init<DBSID>.ora` parameter `IFILE` is not in the directory `<ORACLE_HOME>/dbs`.
- *Program response:* Processing terminates
- *User action:* Correct the parameter and repeat the process.

BR186E Compress directories %s and %s on the same disk volume

- *Explanation:* If you define several directories in the profile parameter `compress_dir`, then they should be on different disks.
- *Program response:* Processing terminates
- *User action:* Correct the definition in the `init.sap` profile.

BR187E Value %d of parameter/option %s smaller than %d

- *Explanation:* The value of the specified parameter or option is lower than the minimum permitted value.
- *Program response:* Processing terminates
- *User action:* Correct the parameter or option and repeat the process.

BR188E Value %d of parameter/option %s greater than %d

- *Explanation:* The value of the specified parameter exceeds the maximum value permitted.
- *Program response:* Processing terminates
- *User action:* Correct the parameter or option and repeat the process.

Messages from BR182E to BR218E

BR189W Expiration period equals 0 - backup volumes could be immediately overwritten

- *Explanation:* The value of the `expir_period` parameter is 0 - the expiration check has been switched off. This means that volumes created on this day can be overwritten. This could lead to the loss of a current backup.
- *Program response:* Processing continues.
- *User action:* Correct the parameter. The value of `expiration_period` should be set to 0 for test purposes only.

BR190E Parameter %s in form \\.\tapex not supported any more - please use %s instead

- *Explanation:* For reliability, tape device addresses of the form `\\.\tapex` are no longer supported.
- *Program response:* The program terminates.
- *User action:* Set the `init<DBSID>.sap` parameter `tape_address` to a value such as `/dev/nmtx`.

BR191E Value '%s' of parameter/option '%s' not supported for '%s'

- *Explanation:* The parameter or option is not supported.
- *Program response:* Processing terminates.
- *User action:* Check the parameter definition or command syntax (e.g. using option `-h`). Repeat the process.

BR192E cpio block size in option - C is not a multiple of 512

- *Explanation:* (Windows NT only) When the definition of the `init<DBSID>.sap` parameter `cpio_flags` or `cpio_in_flags` was being checked, it was discovered that the block size defined there is not a multiple of 512. However, under Windows NT this is a requirement for being able to restore a backup.
- *Program response:* Processing terminates
- *User action:* Change the cpio block size (for example, to 16384) and repeat the process.

BR193I,W,E Files to be backed up are already compressed

- *Explanation:* The software was already compressed during the backup to disk that you now want to copy with the BRBACKUP option `-b`.
- *Program response:* Backup continues if `brbackup -b` runs without software compression (BR193I) or if software compression is active but ignored (BR193W). A backup with BACKINT causes a termination (BR193E).
- *User action:* BR193W: Back up the disk backup without software compression (parameter setting `compress=no` or BRBACKUP option `-k no`). BR193E: Back up the disk without software compression.

BR194I,W,E First offline redo log file %s to be backed up, which was found in line %d of %s, is already compressed

- *Explanation:* The software was compressed when the offline redo log files were backed up to disk that you now want to copy with the BRARCHIVE option `-a`.

Messages from BR182E to BR218E

- *Program response:* Backup continues if `brbackup -a` runs without software compression (BR194I) or if software compression is active but ignored (BR194W). A backup with BACKINT causes a termination (BR194E).
- *User action:* BR194W: Back up the disk backup without software compression (parameter setting `compress=no` or BRARCHIVE option `-k no`). BR194E: Back up the disk without software compression.

BR195I Compressed files cannot be verified using DB_VERIFY

- *Explanation:* BRBACKUP and BRRESTORE cannot use DB_VERIFY to verify a backup made with software compression.
- *Program response:* BRBACKUP or BRRESTORE terminates.
- *User action:* Start any subsequent backups that you want to verify with DB_VERIFY without software compression, or verify the database files by using the BRBACKUP option `-w|-verify only_dbv` (see [Backup Verifications \[Page 155\]](#)).

BR196I Original file %s was a file on raw device - its copy on disk cannot be saved by backup utility

- *Explanation:* Backing up a backup to disk using BACKINT (BRBACKUP option `-b`) is not supported for raw device files.
- *Program response:* Files on raw devices are ignored. BRBACKUP backs up the other (non-raw device) files.
- *User action:* Use BACKINT to back up the raw device files directly (without the option `-b`).

BR197I Chained start not supported for '%s'

- *Explanation:* A BRBACKUP and BRARCHIVE backup in a single run is not supported for all combinations of parameters or options (for example, not for backups to disk with `backup_dev_type = disk`).
- *Program response:* The BRBACKUP/BRARCHIVE run terminates.
- *User action:* Do not use the parameter or option in the message for the chained start.

BR198I Profiles successfully saved

- *Explanation:* The Oracle, SAPDBA and backup profiles were copied successfully to the backup volume.
- *Program response:* none
- *User action:* none

BR199I Volume label(s) successfully checked

- *Explanation:* BRRESTORE checked the volume labels successfully.
- *Program response:* none
- *User action:* none

BR201I Compressing %s

- *Explanation:* Compression of a file has been started.
- *Program response:* none

Messages from BR182E to BR218E

- *User action:* none

BR202I Saving %s

- *Explanation:* Backup of a file has been started.
- *Program response:* none
- *User action:* none

BR203I to %s...

- *Explanation:* The backup medium for the file listed in BR202I is displayed.
- *Program response:* none
- *User action:* none

BR204I Percentage done: %3.2f%%, estimated end time: %02d:%02d

- *Explanation:* The percentage of data processed is displayed. The end time of the processing is estimated based on this number.
- *Program response:* none
- *User action:* none

BR205I Backup volume initialization

- *Explanation:* You called the SAP utility with option `-i`.
- *Program response:* The mounted volumes are initialized, or their labels are displayed.
- *User action:* none

BR206I End of backup volume initialization

- *Explanation:* End of backup volume initialization.
- *Program response:* none
- *User action:* none

BR207I Volumes in devices: %s will be initialized with names: %s

- *Explanation:* The names that will be assigned to the volumes during initialization are displayed.
- *Program response:* none
- *User action:* none

BR208I Volume with name %s required in device %s

- *Explanation:* Based on volume management, one or more volumes with the name(s) listed have been found whose expiration period has expired.
- *Program response:* The program checks whether the correct volume was mounted.
- *User action:* You must mount the volume(s) in the specified device(s) for the backup.

BR209I Volume in device %s has name %s

- *Explanation:* The name of the volume in a backup device is displayed.

Messages from BR182E to BR218E

- *Program response:* none

- *User action:* none

BR210I Please mount the volumes, if you have not already done so

- *Explanation:* The requested volume must be mounted in the device.
- *Program response:* Waits for confirmation.
- *User action:* You must mount the volume(s) in the device(s) for the backup.

BR211W Unattended mode impossible - volume remount required

BR211E Unattended mode impossible - volume remount required

- *Explanation:* If you use option `-c` in BRBACKUP/BRARCHIVE calls, you suppress most of the confirmation messages. If the utility determines that more volumes are required than the number of available devices, you have to intervene in the process and change the volume. BRBACKUP therefore issues this warning at the start of the process. You will be prompted to switch the volume at the proper time. If you use option `-c force`, this situation causes the program to terminate (see [-c-confirm \[Page 369\]](#)).
- *Program response:* Processing continues (BR211W) or terminates (BR211E).
- *User action:* If a warning was issued, you will be prompted to switch volumes at a later time. If the program terminates, do not use `force` with the `-c` option in the next program call.

BR212I Please check the volume or the device

- *Explanation:* A volume or device error has occurred.
- *Program response:* Processing terminated due to previous errors.
- *User action:* To examine the errors, view the preceding error messages.

BR213W Rewind of volume in device %s failed

BR213E Winding tape volume in device %s failed

- *Explanation:* The volume could not be rewound, or positioned on a certain file.
- *Program response:* Waits for user action (BR213W) or processing terminates (BR213E).
- *User action:* Analyze the preceding error messages and correct the error(s).

BR214W Label on volume in device %s not found

BR214E Label on volume in device %s not found

- *Explanation:* Case 1: If this error occurs at the beginning of the backup, then the volume is probably not an SAP volume.
Case 2: If this error occurs at the end of a backup, this means that the backup to that volume was not successful.
- *Program response:* Waits for user action (BR214W) or processing terminates (BR214E).
- *User action:* Case 1: Initialize the volume with option `-i force` and repeat the operation.
Fall 2: Check whether the parameters `tape_address` (no-rewind address) and `tape_address_rew` (rewind address) are defined correctly. Make sure than no other

Messages from BR182E to BR218E

processes access the backup device during the backup. Correct the contents of the volume if necessary and repeat the operation.

BR215W Label on volume in device %s has wrong format

BR215E Label on volume in device %s has wrong format

- *Explanation:* The label of a volume is damaged.
- *Program response:* Waits for user action (BR215W) or processing terminates (BR215E).
- *User action:* Correct the error and then repeat the operation. Also see the information on message BR214X

BR216W Wrong volume %s found in device %s, expected: %s BR216E Wrong

volume %s found in device %s, expected %s %s

- *Explanation:* You mounted an incorrect volume in the backup device. The volume name was determined automatically. The program requires that you stick to this value.
- *Program response:* Waits for the user to mount the correct volume (BR216W) or processing terminated (BR216E).
- *User action:* Mount the correct volume and continue the operation or repeat the operation.

BR217W Lock period for volume %s in device %s not expired

BR217E Lock period for volume %s in device %s not expired

- *Explanation:* The volume mounted in the backup device is still locked. You can only use the volume when the period configured by parameter `expir_period` in the `init.sap` profile has expired.
- *Program response:* Waits for the user to mount the correct volume (BR217W) or processing terminates (BR217E).
- *User action:*

BR218W Checking label on volume in device %s failed

BR218E Reading volume label from %s failed

- *Explanation:* This is a general error, and is displayed along with other label errors. You may have mounted a non-SAP volume, although an SAP volume was requested, or problems exist with the device used.
- *Program response:* Waits for the user to mount the correct volume (BR218W) or processing terminates (BR218E).
- *User action:* Analyze and correct the error(s), repeat the operation if necessary.

Messages from BR219W to BR241I

BR219W Initialization of volume %s in device %s failed.

BR219E Initialization of volume %s in device %s failed.

- *Explanation:* The initialization of a volume has failed.
- *Program response:* Waits for the user to mount the correct volume (BR219W) or processing terminated (BR219E).
- *User action:* If you used option `-i` for initialization, the mounted volume must be an SAP volume. Verify this! If the correct volume was mounted, inspect the device. If you want to initialize a different (non-SAP) volume, use the option `-i force`. Also note the other error messages.

BR220W Volume in device %s is write protected!

- *Explanation:* The volume in the device is write protected.
- *Program response:* The operation can be continued when cpio can write to the terminal window.
- *User action:* Remove the write protection and continue the operation.

BR221W Writing of volume header files to %s failed.

BR221E Writing of volume header files to %s failed.

- *Explanation:* The write operation for the first files (label, `init<DBSID>.ora`, `init<DBSID>.sap`) of the volume failed.
- *Program response:* Waits for user action (BR221W) or processing terminated (BR221E).
- *User action:* Note the other error messages issued along with this message. Correct the error and then repeat the operation.

BR222W Copying %s to/from %s failed due to previous errors.

BR222E Copying %s to/from %s failed due to previous errors.

- *Explanation:* This message is displayed after other error messages that caused a backup to terminate. If you only use one device, the entire backup is terminated. If several devices are available, the backup is continued on the other devices.
- *Program response:* Waits for user action (BR222W) or processing terminated (BR222E).
- *User action:* Note the other messages. Correct the error and repeat the operation, if necessary, for the objects that were not backed up.

BR223E Last copy process failed.

- *Explanation:* The last copy process on the last available device failed due to a previous error.
- *Program response:* Processing terminated.
- *User action:* Note the other messages. Correct the error and then repeat the operation.

BR224E Compress of %s to %s failed.

- *Explanation:* A file could not be compressed.

Messages from BR219W to BR241I

- *Program response:* Processing terminated.
- *User action:* Note the other messages. Correct the error and then repeat the operation.

BR225W Maximal 255 backup devices can be used.

- *Explanation:* This is the maximum number of that can be used in parallel. These devices were defined in the parameters `tape_address` and `tape_address_rewind`.
- *Program response:* Only the first 255 backup devices are used. The rest are ignored.
- *User action:* Remove the unnecessary device addresses from the parameters in the `init<DBSID>.sap` profile.

BR226I Rewinding tape volume in device %s...

- *Explanation:* Volume is rewound in a tape device.
- *Program response:* None
- *User action:* None

BR227W Not enough disk space in %s for backup, missing %4.3f MB.**BR227E Not enough disk space in %s for backup, missing %4.3f MB.**

- *Explanation:* During a disk backup, the freespace check determined that the existing disk space is not sufficient for the backup.
- *Program response:* If you used option `-q`, only a warning is issued. Otherwise, processing is terminated.
- *User action:* Define a different file directory in parameter `backup_root_dir`, if you are backing up to disk, or try to create more space in the existing directory. If necessary, restart the operation.

BR228W Not enough disk space in %s for compression, missing %4.3f MB.**BR228E Not enough disk space in %s for compression, missing %4.3f MB.**

- *Explanation:* During the freespace check, the system determined that the directory defined in parameter `compress_dir` does not have enough freespace for the largest compressed file. The size of the compressed file is calculated based on the last available compression rates. If none exist, default values for the compression rates are assumed. See [Log Supplements \[Page 442\]](#).
- *Program response:* If you used option `-q`, only a warning is issued. Otherwise, processing is terminated.
- *User action:* Define a different file directory in parameter `compress_dir` or try to create more space in the existing directory. Then restart the operation.

BR229I Calling backup utility...

- *Explanation:* Reports that the BACKINT interface program is being called in order to establish a connection to an external backup program.
- *Program response:* None.
- *User action:* None.

BR230I Backup utility called successfully.

Messages from BR219W to BR241I

- *Explanation:* The external backup program has successfully completed the backup request.
- *Program response:* None.
- *User action:* None.

BR231E Backup utility call failed.

- *Explanation:* The external backup program could not carry out the backup request.
- *Program response:* Processing terminated.
- *User action:* Note the other error messages, particularly those from the external backup program. Correct the error and then repeat the operation. If necessary, contact the supplier of the external backup program.

BR232I %d of %d files saved by backup utility.**BR232E %d of %d files saved by backup utility.**

- *Explanation:* The number of files saved by the external backup program is displayed.
- *Program response:* If this number is smaller than the number of files for backup, this means errors have occurred, and processing is terminated.
- *User action:* If errors occurred, analyze them. Also note the other error messages, particularly those from the external backup program. If necessary, contact the supplier of the external backup program.

BR233E Error in backup utility while saving file %s

- *Explanation:* The external backup program reported an error while saving the file.
- *Program response:* Processing terminated.
- *User action:* Check the other messages from the external backup program. If necessary, contact the supplier of the external backup program (the supplier of the BACKINT interface program).

BR234E Unknown file %s saved by backup utility.

- *Explanation:* A file that was not part of the backup request was saved by the external backup program.
- *Program response:* Processing terminated.
- *User action:* Contact the supplier of the external backup program (the supplier of the BACKINT interface program).

BR235W Recommended number of usages of volume %s exceeded.

- *Explanation:* The recommended number of usages defined in `tape_use_count` has been exceeded by the specified volume.
- *Program response:* The backup is continued.
- *User action:* Do not use this volume for any more backups.

BR236I Perhaps volume initialization with option '-i force' required.

Explanation: This message appears when you attempt to start a backup on a non-initialized volume. The following cpio messages (platform-specific) may appear along with this message:

Messages from BR219W to BR241I

Operating System	Message
SINIX	cpio: Bad header - checksum error. cpio: Not a cpio file.
AIX	Cannot read from the specified input. Out of phase.
HP-UX	Out of phase -- get help. End of volume.
OSF1	Not a cpio archive.
SUN	cpio: impossible header type.
NT	cpio: bad magic number in archive

- *Program response:* Processing terminated.
- *User action:* Initialize the volume using option `-i force` and then repeat the operation.

BR237I Scratch volume required in device %s

- *Explanation:* Prompt to provide a scratch volume for the backup.
- *Program response:* Waits for a volume to be mounted.
- *User action:* Any volume whose expiration period has expired can be mounted in this device.

BR238I cpio continuation volume required in device %s.

- *Explanation:* The `cpio` command reached the physical end of the volume. See [cpio Continuation Tape \[Page 224\]](#).
- *Program response:* Waits for a continuation volume to be mounted.
- *User action:* In this case, you should only mount a volume that has not been used for BRBACKUP/BRARCHIVE backups as the continuation volume (e.g. a volume that is not specified in the pool). When a cpio continuation volume is written, any existing labels are simply overwritten. To avoid this situation in future, reduce the value of parameter `tape_size`.

BR239W No label checking on cpio continuation volume

- *Explanation:* No label checking will be performed for a cpio continuation volume.
- *Program response:* The backup is continued.
- *User action:* To avoid this situation in future, SAP recommends reducing the volume size defined by parameter `tape_size` in the `init<DBSID>.sap` profile such that BRBACKUP/BRARCHIVE can assume the continuation volume handling.

BR240E Less data backed up than the size of file %s

- *Explanation:* The quantity of saved data reported by `cpio` is less than the size of the file for backup.
- *Program response:* Processing terminated.
- *User action:* Also check the other error messages to find and correct the cause of the error.

Messages from BR219W to BR241I

BR241I Checking label on volume in device %s

- *Explanation:* Before a volume can be used for a backup, its label is checked. This check is also performed after the backup to ensure that the backup was performed successfully.
- *Program response:* Processing is continued.
- *User action:* Depends on the program response (see [Volume Management \[Page 142\]](#)).

Messages from BR242I to BR269W

Messages from BR242I to BR269W

BR242I Scratch volume in device %s will be renamed to %s

- *Explanation:* A volume with name SCRATCH will be renamed during a backup. This can only happen when the volume had the name SCRATCH.
- *Program response:* The volume is renamed.
- *User action:* None. You should not initialize any volumes with the name SCRATCH, however. See [Scratch Volumes \[Page 146\]](#).

BR243W Wrong creation date %s for volume %s in device %s

BR243E Wrong creation date %s for volume %s in device %

- *Explanation:* The creation date found in the label is wrong. The volume was not written during the backup that you selected for the restore operation.
- *Program response:* Processing terminated.
- *User action:* Investigate the cause of the error. Repeat the operation with a correct volume.

BR244I Unload the volume in device %s and mount it once again!

- *Explanation:* The properties of the backup device require you to unload the volume and mount it again.
- *Program response:* Waits for execution.
- *User action:* Unload the volume and mount it again.

BR245W Directory %s will not be compressed.

- *Explanation:* Despite the fact that software compression is active (parameter `compress = yes` or option `-k yes`), the directory contents will not be compressed. This is a limitation of BRBACKUP.
- *Program response:* Processing is continued without compression.
- *User action:* None.

BR246E Error in backup utility for unknown file %s

- *Explanation:* The external backup program reported an error when backing up a file that was not part of the backup request.
- *Program response:* Processing terminated.
- *User action:* Please contact the supplier of the external backup program (supplier of interface program BACKINT).

BR247E No Backup utility ID or ID longer than %d characters.

- *Explanation:* The backup utility ID returned by the external backup program is longer than 16 characters. This is illegal.
- *Program response:* Processing terminated.

Messages from BR242I to BR269W

- *User action:* Please contact the supplier of the external backup program (supplier of interface program BACKINT).

BR251W Function %s failed at location %s-%d

BR251E Function %s failed at location %s-%d

- *Explanation:* A standard C function could not be executed.
- *Program response:* Warning and processing is continued (BR251W) or processing terminated (BR251E).
- *User action:* If a termination occurs, the function returned an error return code (BR253E). The information about the termination point should be forwarded to SAP Support.

BR252W Function %s failed for '%s' at location %s-%d

BR252E Function %s failed for '%s' at location %s-%d

- *Explanation:* A standard C function could not be executed.
- *Program response:* Warning and processing is continued (BR252W) or processing terminated (BR252E).
- *User action:* See BR251X. In addition, the parameters of the function that caused the termination are displayed.

BR253W errno %d: %s

BR253E errno %d: %s

- *Explanation:* The system error number and the corresponding system text are displayed - these should help you find out why the standard C function failed.
- *Program response:* Warning and processing is continued (BR253W) or processing terminated (BR253E).
- *User action:* Analyze this message and try to correct the error.

BR254E Memory allocation of %d bytes failed at location %s

- *Explanation:* No main memory is available. Either involves an internal program error or the swap areas were defined too small.
- *Program response:* Processing terminated.
- *User action:* Try to correct the error(s).

BR255W Cannot read from standard input.

BR255E Cannot read from standard input.

- *Explanation:* The standard input (console) is not available, for example, when you started the program with CRON or through the SAP CCMS.
- *Program response:* Warning and processing is continued (BR255W) or processing terminated (BR255E).
- *User action:* If additional error messages are displayed, analyze them and correct the error(s).

BR256I Please enter 'cont' to continue, 'stop' to cancel the program:

- *Explanation:* Entry prompt to confirm the action.
- *Program response:* Waits for entry.

Messages from BR242I to BR269W

- *User action:* Enter one of the two strings to either continue processing or cancel it.

BR257I Your reply: '%s'

- *Explanation:* Confirms a correct reply to message BR256I.
- *Program response:* Reply “cont” - Processing is continued. Reply “stop” - Processing terminated.
- *User action:* None.

BR258I Wrong reply: '%s'

- *Explanation:* An incorrect reply was entered for message BR256I.
- *Program response:* The program requires entry of a correct reply and displays message BR256I again.
- *User action:* Reply properly with “cont” or “stop”.

BR259I Program execution will be continued...

- *Explanation:* Confirms that program execution will be continued when you reply to message BR256I with “cont”.
- *Program response:* Processing is continued.
- *User action:* None.

BR260E Program cancelled by user!

- *Explanation:* You cancelled the program by entering “stop” as the reply to BR256I.
- *Program response:* Processing terminated.
- *User action:* None.

BR261E Program cancelled by signal %d!

- *Explanation:* The program received a signal from the user or the system. If this message appears together with BR312E or BR313E, BRBACKUP was interrupted by BRCONNECT.
- *Program response:* Processing terminated.
- *User action:* Analyze the situation and repeat the operation if necessary.

BR262I Please enter DBA user name [/password]:

- *Explanation:* You used option -u without any other entries.
- *Program response:* Prompt for interactive entry of user name and password.
- *User action:* Enter the user name and password. In this case, the password will not appear in the process list. The following length restrictions apply to the passwords: HP-UX: 8 characters, OSF/1: 16 characters, AIX: 32 characters, SINIX: 8 characters, SUN: 8 characters.

BR263I Please enter password for user '%s' (maximal %d characters):

- *Explanation:* You used option -u but only specified the user name.
- *Program response:* Prompt for interactive entry of password.

Messages from BR242I to BR269W

- *User action:* Enter the password. In this case, the password will not appear in the process list, and is hidden upon entry. The following length restrictions apply to the passwords: HP-UX: 8 characters, OSF/1: 16 characters, AIX: 32 characters, SINIX: 8 characters, SUN: 8 characters.

BR264E Username/password cannot be read.

- *Explanation:* The program tried to read from the standard input (console). This is not available, however, when the program was started with CRON or through the SAP CCMS, for example.
- *Program response:* Processing terminated.
- *User action:* See [Unattended Backup \[Page 139\]](#). You will find examples of how to schedule backups with CRON.

BR265W Execution of program '%s' failed.

BR265E Execution of program '%s' failed.

- *Explanation:* The execution of a subprogram that was called by the main program failed. This does not mean that the subprogram reported an error, but rather that the program could not be called properly.
- *Program response:* Processing is continued (BR265W) or terminated (BR265E).
- *User action:* Analyze and correct the error(s) and then repeat the operation.

BR266W Program '%s' interrupted, exit status: %x

BR266E Program '%s' interrupted, exit status: %x

- *Explanation:* A subprogram was interrupted and an exit status specified.
- *Program response:* Processing is continued (BR266W) or terminated (BR266E).
- *User action:* Information about the cause of the error can be found in the exit status. The status information is operating system-specific.

BR267W Asynchronous execution of program failed.

BR267E Asynchronous execution of program failed.

- *Explanation:* The asynchronous execution of a subprogram failed. This error can occur, for example, when you use several devices for asynchronous copy processes. The message does not mean that the subprogram reported an error, but rather that the program could not be called properly.
- *Program response:* Processing is continued (BR267W) or terminated (BR267E).
- *User action:* Analyze and correct the error(s) and then repeat the operation.

BR268W Asynchronous execution of a program interrupted, exit status: %x

BR268E Asynchronous execution of a program interrupted, exit status: %x

- *Explanation:* The asynchronous execution of a subprogram was interrupted and an exit status specified. This error can occur, for example, when you use several devices for asynchronous copy processes.
- *Program response:* Processing is continued (BR268W) or terminated (BR268E).

Messages from BR242I to BR269W

- *User action:* Information about why the subprogram was interrupted can be found in the exit status. The status information is operating system-specific.

BR269W Wait situation interrupted - probably cannot continue.

- *Explanation:* The program was interrupted with program with `ctrl-C` (or by the SIGINT signal) while it was waiting for one of its subprograms to finish.
- *Program response:* Continuation of processing will probably fail.
- *User action:* If message BR256I is also displayed, decide whether you want to continue or cancel the program.

Messages from BR270W to BR307I

BR270W Unknown process with ID %d terminated.

- *Explanation:* A subprogram that was not started by the main program was terminated.
- *Program response:* Termination of the subprogram is ignored and processing is continued.
- *User action:* Could involve an ORACLE background process error or an internal error.

BR271W Read from a pipe failed for '%s'

BR271E Read from a pipe failed for '%s'

- *Explanation:* The reading of messages from subprograms whose output was routed through a pipe mechanism failed.
- *Program response:* Processing is continued (BR271W) or terminated (BR271E).
- *User action:* Analyze the other error messages and repeat the operation, once you correct the error.

BR272W Execution of program '%s' through pipe failed.

BR272E Execution of program '%s' through pipe failed.

- *Explanation:* The start of subprograms whose output was routed through a pipe mechanism failed.
- *Program response:* Processing is continued (BR272W) or terminated (BR272E).
- *User action:* Analyze the other error messages and repeat the operation, once you correct the error.

BR273W Determination of file status for %s failed.

BR273E Determination of file status for %s failed.

- *Explanation:* Information about a file could not be determined, possibly because the file does not exist or the user is not authorized, for example.
- *Program response:* Processing is continued (BR273W) or terminated (BR273E).
- *User action:* Check the other error messages and correct the error(s).

BR274W File %s not found.

BR274E File %s not found.

- *Explanation:* The specified file does not exist.
- *Program response:* Processing is continued (BR274W) or terminated (BR274E).
- *User action:* The action depends on the file category.

BR275E File %s is not a regular file.

- *Explanation:* Symbolic links, named pipes, and device files (special files) cannot be backed up.
- *Program response:* Processing terminated.
- *User action:* The action depends on the file category.

Messages from BR270W to BR307I

BR276E File/directory %s is empty.

- *Explanation:* Files with length 0 and empty directories cannot be backed up.
- *Program response:* Processing terminated.
- *User action:* None.

BR277W Determination of disk volume status for %s failed.

BR277E Determination of disk volume status for %s failed.

- *Explanation:* Information about a disk volume could not be determined. The disk volume status is used mainly to determine the freespace on a disk.
- *Program response:* Processing is continued (BR277W) or terminated (BR277E).
- *User action:* Correct the error and then start the operation again.

BR278W Command output of '%s': %s

BR278E Command output of '%s': %s

- *Explanation:* The error messages from this command are displayed. In most cases, messages from the `cpio` or `mt` commands are involved. A few examples:

Message	Explanation
[cpio:]cannot read... [cpio:]read error... [cpio:]can't read input... [cpio:]I/O Error... [cpio:]read failed:I/O Error.	The file for backup cannot be read. In most cases, this involves hardware problems on the disk where the file is stored.
[cpio:]cannot write... [cpio:]write error... [cpio:]can't write output... [cpio:]I/O Error... [cpio:]write failed:I/O Error.	The file for backup cannot be written to the volume. In most cases, this involves hardware problems with the volume or the backup device.
No such device or address [sh:]/dev/...:cannot create	The backup device is not responding, e.g. when no volume (tape) has been mounted in the backup device.
[cpio:]cannot open...	The input or output file cannot be open. In most cases, this involves authorization problems with the operating system user.
[cpio:]can't open /dev/tty to prompt for more media	The physical end of the tape was reached. Because a parallel backup is being performed or the option <code>-c force</code> was used, however, <code>cpio</code> is not able to request a continuation volume. The tape size (parameter <code>tape_size</code>) must be reduced.

Messages from BR270W to BR307I

- *Program response:* Processing is continued (BR278W) or terminated (BR278E).
- *User action:* Analyze the error messages. Correct the error(s) listed in the message displayed for the subprogram command.

BR279W Return code from '%s': %d

BR279E Return code from '%s': %d

- *Explanation:* The return code of a command is displayed. This involves the return code of a subprogram started by the main program. The message often appears together with BR278X.
- *Program response:* Processing is continued (BR279W) or terminated (BR279E).
- *User action:* Also analyze the error messages from the subprogram (BR278X) and correct the error(s).

BR280I Time stamp %s

- *Explanation:* The timestamp is displayed.
- *Program response:* None.
- *User action:* None.

BR281W Program interrupted by signal %d.

- *Explanation:* The program was interrupted by a SIGINT signal from the user or the system.
- *Program response:* Message BR256I is then displayed. Waits for your entry.
- *User action:* Decide whether you want to continue or cancel the program. Continuation of the program will probably fail.

BR282E Directory %s not found.

- *Explanation:* The specified directory does not exist.
- *Program response:* Processing terminated.
- *User action:* Analyze the situation. If necessary, repeat the operation and enter the correct directory name.

BR283E Command/Output buffer overflow.

- *Explanation:* The buffer for the command lines and command output overflowed.
- *Program response:* The buffer contents are displayed and processing is terminated.
- *User action:* Probably involves an internal error. Please contact SAP.

BR284I Time stamp %s, elapsed time: %s

- *Explanation:* An extended timestamp is displayed when you use option `-o time`. The time that has elapsed since the last timestamp is displayed.
- *Program response:* None.
- *User action:* None.

BR285I This function can take several seconds/minutes - be patient.

- *Explanation:* An action was started that may take several seconds/minutes.

Messages from BR270W to BR307I

- *Program response:* Action running.
- *User action:* None. Wait for the action to finish.

BR286W Size of file %s is different from Oracle file size.

- *Explanation:* The file size in the file system is different from the file size saved in the ORACLE Dictionary (in the control file). The file may be corrupted. This warning is also sent if the size of the file changed during an online backup with the Oracle auto-extend mechanism. The size given in the header of the BRBACKUP log is then no longer up-to-date.
- *Program response:* The program tries to save the file.
- *User action:* Check your disks, the file system (file system check), and the ORACLE ALERT file. If you discover errors, you can possibly correct the situation by performing a recovery. Seek help from an ORACLE specialist before you start a recovery. When auto-extending a file take the increased storage requirements into account when reloading the file.

BR301W SQL error %d at location %s-%d**BR301E SQL error %d at location %s-%d**

- *Explanation:* An SQL error has occurred.
- *Program response:* Processing is continued (BR301W) or terminated (BR301E).
- *User action:* Check the SQL code and repeat the operation.

BR302E SQLDBA call for database instance %s failed.

- *Explanation:* The call of the SQLDBA program failed.
- *Program response:* Processing terminated.
- *User action:* This error often occurs when starting up and shutting down a database instance. Check any ORACLE error messages. Correct the error and then repeat the operation.

BR303E Determination of Oracle RDBMS version failed.

- *Explanation:* Before the database is accessed, SQLDBA determines the ORACLE version. This operation failed.
- *Program response:* Processing terminated.
- *User action:* Note the ORACLE error messages.

BR304I Starting the database instance %s...

- *Explanation:* The database instance is being started by BRBACKUP via SQLDBA.
- *Program response:* None.
- *User action:* None.

BR305I Startup of database instance %s successful.

- *Explanation:* The database instance startup via SQLDBA was successful.
- *Program response:* None.
- *User action:* None.

Messages from BR270W to BR307I

BR306E Startup of database instance %s failed.

- *Explanation:* The database instance startup via SQLDBA failed.
- *Program response:* Processing terminated.
- *User action:* Analyze the detailed error information and repeat the operation, once you correct the error.

BR307I Shutting down the database instance %s...

- *Explanation:* The database instance is being shut down by BRBACKUP via SQLDBA.
- *Program response:* None.
- *User action:* None.

Messages from BR308I to BR339W

Messages from BR308I to BR339W

BR308I Shutdown of database instance %s successful.

- *Explanation:* The database instance shutdown via SQLDBA was successful.
- *Program response:* None.
- *User action:* None.

BR309E Shutdown of database instance %s failed.

- *Explanation:* The database instance shutdown via SQLDBA failed.
- *Program response:* Processing terminated.
- *User action:* Analyze the detailed error information and repeat the operation, once you correct the error.

BR310W Connect to database instance %s failed.

BR310E Connect to database instance %s failed.

- *Explanation:* The connection to the database instance could not be established.
- *Program response:* Processing is continued (BR310W) or terminated (BR310E).
- *User action:* Analyze the ORACLE error message and - if necessary - repeat the operation once you correct the error.

BR311E Connect to database instance %s during online backup failed.

- *Explanation:* The database instance must be available during an online backup. This was not the case.
- *Program response:* Processing terminated.
- *User action:* Analyze the ORACLE error message and repeat the operation once you have corrected the error.

BR312E Database instance %s shutdown during online backup.

- *Explanation:* The database instance was shut down during an online backup.
- *Program response:* Processing terminated.
- *User action:* The database instance must always be active during an online backup. Repeat the operation.

BR313E Database instance %s started up during offline backup.

- *Explanation:* The database instance was started up during an offline backup.
- *Program response:* Processing terminated.
- *User action:* The database instance must always remain shut down during an offline backup. Repeat the operation.

BR314E Collection of information on database files failed.

Messages from BR308I to BR339W

- *Explanation:* Information about the database could not be determined.
- *Program response:* Processing terminated.
- *User action:* Involves an SQL error that you must correct. Then restart the operation.

BR315I Alter tablespace %s begin backup successful.

- *Explanation:* A tablespace was successfully switched to backup status.
- *Program response:* None.
- *User action:* None.

BR316E Alter tablespace %s begin backup failed.

- *Explanation:* The attempt to switch a tablespace to backup status failed.
- *Program response:* Processing terminated.
- *User action:* Analyze the ORACLE error message and repeat the operation once you have corrected the error.

BR317I Alter tablespace %s end backup successful.

- *Explanation:* The termination of the backup status for the tablespace was successful.
- *Program response:* None.
- *User action:* None.

BR318E Alter tablespace %s end backup failed.

- *Explanation:* The termination of the backup status for the tablespace failed.
- *Program response:* Processing terminated.
- *User action:* Analyze the ORACLE error message and repeat the operation once you have corrected the error.

BR319I Control file copied to %s

- *Explanation:* A copy of the control file has been created. Such a copy is always created when you start an online backup.
- *Program response:* None.
- *User action:* None.

BR320E Copying of control file to %s failed.

- *Explanation:* The copy of the control file could not be created.
- *Program response:* Processing terminated.
- *User action:* This error message is displayed during an online backup. Analyze the ORACLE error messages and correct the error(s). Then repeat the backup.

BR321I Switch to next online redo log file for database instance %s successful.

Messages from BR308I to BR339W

- *Explanation:* The redo log file switch was successful, that is, the program switched to the next online redo log file. This file switch is always performed after an online backup.
- *Program response:* None.
- *User action:* None.

BR322E Switch to next online redo log file for database instance %s failed

- *Explanation:* The redo log file switch failed, that is, the program could not switch to the next online redo log file.
- *Program response:* Processing terminated.
- *User action:* This error message is displayed during an online backup. Analyze the ORACLE error messages and correct the error(s). Then repeat the backup.

BR323W 'Archive log list' for database instance %s failed.

BR323E 'Archive log list' for database instance %s failed.

- *Explanation:* The SQLDBA call for ARCHIVE LOG LIST failed.
- *Program response:* Processing is continued (BR323W) or terminated (BR323E).
- *User action:* Often involves an ORACLE problem. Analyze the ORACLE error messages and correct the error(s). If the backup run was already finished, it does not need to be repeated.

BR324W Insertion of database log header failed.

- *Explanation:* An entry could not be added to table SDBAH. This error can occur when a message is sent by BRBACKUP/BRARCHIVE to the Computing Center Management System.
- *Program response:* The table is not changed, and the operation is continued.
- *User action:* Often involves an ORACLE problem. Analyze the ORACLE error messages and correct the error(s), so that it no longer occurs during the next run.

BR325W Writing of database log failed.

- *Explanation:* Table SDBAD could not be updated.
- *Program response:* The table is not changed, and the operation is continued.
- *User action:* Often involves an ORACLE problem. Analyze the ORACLE error messages and correct the error(s), so that it no longer occurs during the next run.

BR326W Update of database log header failed.

- *Explanation:* An entry in table SDBAH could not be updated.
- *Program response:* The table is not changed, and the operation is continued.
- *User action:* Often involves an ORACLE problem. Analyze the ORACLE error messages and correct the error(s), so that it no longer occurs during the next run.

BR327W Disconnect from database instance %s failed.

- *Explanation:* The connection to the database instance could not be closed properly.
- *Program response:* The error is ignored and processing is continued.

Messages from BR308I to BR339W

- *User action:* Often involves an ORACLE problem. Analyze and correct the error(s) and repeat the operation.

BR328E Database file %s from tablespace %s is already in backup status.

- *Explanation:* The tablespace to which the file belongs was just backed up, or the last backup was cancelled. The file is not changed to backup status.
- *Program response:* Processing terminated.
- *User action:* If the last backup was cancelled, manually enter the SQLDBA command ALTER TABLESPACE <tablespace name> END BACKUP.

BR329I Database instance %s is shut down - it will be started now.

- *Explanation:* BRBACKUP confirmation prompt as to whether the database instance can be started up before the backup is started.
- *Program response:* Waits for confirmation.
- *User action:* When message BR256I is displayed, decide whether you want to continue or cancel the program.

BR330I Trying 'alter tablespace end backup' using sqldba...

- *Explanation:* When an online backup is cancelled, BRBACKUP tries to change all the tablespaces from backup status. When the direct attempt fails, BRBACKUP tries to use SQLDBA.
- *Program response:* None.
- *User action:* None.

BR331I 'alter tablespace end backup' using sqldba successful.

- *Explanation:* The attempt described in message BR330I was successful.
- *Program response:* None.
- *User action:* None.

BR332E 'alter tablespace end backup' using sqldba failed.

- *Explanation:* The attempt described in message BR330I failed.
- *Program response:* Processing terminated.
- *User action:* Correct the error(s) that caused the online backup to terminate and repeat the operation.

BR333W Status of tablespace %s is %s.

- *Explanation:* The tablespace status given in DBA_TABLESPACES is neither "ONLINE" nor "READ_ONLY".
- *Program response:* None.
- *User action:* Decide whether the status is acceptable. If required, change the status with the corresponding SQLDBA command.

Messages from BR308I to BR339W

BR334W Status of database file %s is %s.

- *Explanation:* The file status given in V\$DATAFILE is neither "ONLINE" nor "SYSTEM".
- *Program response:* None.
- *User action:* Decide whether the status is acceptable. If required, change the status with the corresponding SQLDBA command.

BR335W Status of online redo log file %s is %s.

- *Explanation:* The online redo log file status given in V\$LOGFILE is neither "" nor "STALE".
- *Program response:* None.
- *User action:* Decide whether the status is acceptable. If required, change the status with the corresponding SQLDBA command.

BR339W Number of parallel instances different from number of threads.

- *Explanation:* The number of ORACLE instances (threads) defined with the `init<DBSID>.sap` parameter `parallel_instances` does not agree with the contents of table V\$THREAD.
- *Program response:* The process can be continued despite this warning.
- *User action:* Check the `init.sap` parameter, correct it if necessary and repeat the process.

Messages from BR340I to BR383E

BR340I Switching to next online redo log file for database instance %s...

- *Explanation:* There is a redo log file change for the specified instance. This means that there is a switch to the next online redo log file.
- *Program response:* None.
- *User action:* None.

BR351I Restoring %s

- *Explanation:* Restore operation started for a file.
- *Program response:* None.
- *User action:* None.

BR352I Decompressing %s

- *Explanation:* Decompression of a file has been started.
- *Program response:* None.
- *User action:* None.

BR353I Verifying backup of %s

- *Explanation:* Verification of a file has been started.
- *Program response:* None.
- *User action:* None.

BR354I Decompressing and verifying the backup of %s

- *Explanation:* Decompression and verification of one file started in one operation.
- *Program response:* None.
- *User action:* None.

BR355I from %s...

- *Explanation:* Device or file from which the file is being restored/decompressed/verified.
- *Program response:* None.
- *User action:* None.

BR357I Verifying volume %s in device %s...

- *Explanation:* Verification of the backups of all files on a volume has been started.
- *Program response:* None.
- *User action:* None.

BR359E Restore of %s from %s failed.

Messages from BR340I to BR383E

- *Explanation:* Restore operation for a file failed.
- *Program response:* Processing terminated.
- *User action:* Analyze the error messages and repeat the operation if necessary.

BR360W Not enough disk space in %s for restore/verification, missing %4.3f MB.

BR360E Not enough disk space in %s for restore/verification, missing %4.3f MB.

- *Explanation:* The check of free disk space shows that the space is not sufficient for restore/verification.
- *Program response:* Processing is continued (BR360W) or terminated (BR360E).
- *User action:* Make sure that enough free disk space is available, and repeat the operation if necessary.

BR361I Restore/verification of %s waiting for disk space...

- *Explanation:* During a parallel backup with verification, the directories from the value list of parameter `backup_root_dir` are used for the restore operation. The free space in these directories must be at least as large as the file for backup. When the freespace is larger than the total size of the n largest files (n is the number of backup devices used), then no bottlenecks can occur with the directories.
- *Program response:* If temporary space shortages occur in the directories, BRBACKUP waits until other verification runs are finished.
- *User action:* If you want to avoid the wait situation, provide more free disk space.

BR362I Verification of backup of %s successful.

- *Explanation:* Confirms that the verification of a saved file was successful.
- *Program response:* None.
- *User action:* None.

BR363E Verification of backup of %s failed.

- *Explanation:* The verification of a saved file failed.
- *Program response:* Processing terminated.
- *User action:* Analyze and correct the error(s).

BR364E Verification of header files on volume in device %s failed.

- *Explanation:* The verification of the volume header files failed: `.tape.hdr0`, `init<DBSID>.ora`, and `init<DBSID>.sap`.
- *Program response:* Processing terminated.
- *User action:* Analyze and correct the error(s).

BR365E The contents of file %s and %s are different.

- *Explanation:* The verification showed that the contents of the saved file are different from those of the original file. The backup was not successful.
- *Program response:* Processing terminated.

Messages from BR340I to BR383E

- *User action:* Check your volume and the backup devices. Repeat the backup to make sure you have a valid backup available.

BR366E The size of the saved file %s is wrong.

- *Explanation:* The verification showed that the size of the saved file is different from that of the original file. The backup was not successful.
- *Program response:* Processing terminated.
- *User action:* Check your volume and the backup devices. Repeat the backup to make sure you have a valid backup available.

BR367W No verification on cpio continuation volume.

- *Explanation:* Files that were saved on the cpio continuation volumes (including the files on the first volume) cannot be verified.
- *Program response:* No verification is performed.
- *User action:* None.

BR368W Backup of directory %s will not be verified.

- *Explanation:* The backup of the contents of a directory cannot be verified.
- *Program response:* No verification is performed.
- *User action:* None. This is a limitation of BRBACKUP.

BR369E %s is not a directory.

- *Explanation:* The path displayed is supposed to refer to a directory.
- *Program response:* Processing terminated.
- *User action:* Correct your entry.

BR370I Directory %s created.

- *Explanation:* A directory was created.
- *Program response:* None.
- *User action:* None.

BR371I Link from %s to %s created.

- *Explanation:* A symbolic link was created.
- *Program response:* None.
- *User action:* None.

BR373I Perhaps cpio_disk_flags '%s' should be used.

- *Explanation:* Special cpio flags are necessary when backing up or restoring directories to/from disk.
- *Program response:* Processing terminated due to previous error(s).
- *User action:* Correct the parameter definition for `cpio_disk_flags` in profile `init<DBSID>.sap`.

Messages from BR340I to BR383E

BR374I %d of %d files restored by backup utility.

BR374E %d of %d files restored by backup utility.

- *Explanation:* The number of files successfully restored by the external backup program is displayed. If the number of restored files is less than the number of files that were supposed to be restored, an error has occurred.
- *Program response:* Processing is continued (BR374I) or terminated (BR374E).
- *User action:* Check the other messages from the external backup program. If necessary, contact the supplier of the external backup program (supplier of interface program BACKINT).

BR375E Error in backup utility while restoring file %s

- *Explanation:* The external backup program reported an error while restoring the file.
- *Program response:* Processing terminated.
- *User action:* Check the other messages from the external backup program. If necessary, contact the supplier of the external backup program (supplier of interface program BACKINT).

BR376E Unknown file %s restored by backup utility.

- *Explanation:* A file that was not part of the restore request was restored by the external backup program.
- *Program response:* Processing terminated.
- *User action:* Contact the supplier of the external backup program (supplier of interface program BACKINT).

BR377E BRTOOLS call failed.

- *Explanation:* A call from the BRTOOLS program failed. This program is used for internal purposes, for example, during the verification of backups.
- *Program response:* Processing terminated.
- *User action:* Analyze the other error messages and correct the error(s).

BR378W Volume name %s not unique - automatic restore impossible.

BR378E Volume name %s not unique - automatic restore impossible.

- *Explanation:* To distinguish the volumes during a restore, their names within the backup must be unique.
- *Program response:* Processing is continued (BR378W, during backup) or terminated (BR278E, during restore).
- *User action:* Make sure that the volume names are unique.

BR379I Distribution of files on volume %s:

- *Explanation:* The SAP utility was called with option `-o dist`. As a result, a list of the file distribution is displayed. See [Log Supplements \[Page 442\]](#).
- *Program response:* None.
- *User action:* None.

BR380I Duration of backup for files saved to volume %s:

Messages from BR340I to BR383E

- *Explanation:* BRBACKUP/BRARCHIVE was started with option `-o time`. As a result, BRBACKUP/BRARCHIVE displays a list with backup durations. See [Log Supplements \[Page 442\]](#).
- *Program response:* None.
- *User action:* None.

BR381I Duration of restore for files saved to volume %s:

- *Explanation:* BRRESTORE was called with option `-o time`. As a result, BRRESTORE displays a list with restore durations. See [Log Supplements \[Page 442\]](#).
- *Program response:* None.
- *User action:* None.

BR383E Raw disk %s not found.

- *Explanation:* A file is to be restored on a raw device but it does not exist.
- *Program response:* Termination of the process.
- *User action:* Compare the database configuration with the hardware configuration and adjust the hardware configuration if necessary.

Messages from BR384E to BR429W

Messages from BR384E to BR429W

BR384E %s is not a raw disk.

- *Explanation:* A backup of a file residing on a raw device can only be restored to a raw device.
- *Program response:* Termination of the process.
- *User action:* Compare the database configuration with the hardware configuration and adjust the hardware configuration if necessary.

BR385E Compress of %s waiting for disk space...

- *Explanation:* In parallel compression, the directories in the value list of parameter `compress_dir` are used for restoring. The free space in these directories must be at least as large as the largest file to be compressed.
- *Program response:* If there should be a temporary shortage of space in these directories, BRBACKUP waits until other compression runs have finished.
- *User action:* Provide more free space if you want to avoid having to wait.

BR401I BRRESTORE %s

- *Explanation:* BRRESTORE version number. This number is independent of the SAP release number.
- *Program response:* None.
- *User action:* None.

BR402I BRRESTORE terminated successfully

- *Explanation:* BRRESTORE ended successfully with return code 0.
- *Program response:* None.
- *User action:* None.

BR403I BRRESTORE terminated with warnings.

- *Explanation:* BRRESTORE ended with return code 1. The restore was performed successfully.
- *Program response:* None.
- *User action:* Pay careful attention to the warnings. Analyze them and - if necessary - correct the cause of the warnings.

BR404I BRRESTORE terminated with errors.

- *Explanation:* BRRESTORE finished with a return code greater than 1.
- *Program response:* The BRRESTORE run was terminated.
- *User action:* Analyze and correct the specified errors. Then repeat the restore operation.

BR405I start of database restore: %s %s %s

Messages from BR384E to BR429W

- *Explanation:* BRRESTORE start message. The action ID (r<encoded timestamp>), the function ID (extension), and the start timestamp (date, time) are listed in sequence to characterize the restore process. The action ID and the function ID form the file name of the detail BRRESTORE log.
- *Program response:* None.
- *User action:* None.

BR406I End of database restore: %s %s %s

- *Explanation:* BRRESTORE end message. The action ID, function ID, and end timestamp are displayed. See BR405I.
- *Program response:* None.
- *User action:* None.

BR407I Restore of database: %s

- *Explanation:* Name of the database to which the files for the restore operation belong is displayed.
- *Program response:* None.
- *User action:* None.

BR408I BRRESTORE action ID: %s

- *Explanation:* The action ID contains an encoded timestamp that specifies the starting time of BRRESTORE. It appears at the start of the file name of the detail BRRESTORE log after an "r".
- *Program response:* None.
- *User action:* None.

BR409I BRRESTORE function ID: %s

- *Explanation:* The function ID (extension) indicates the mode of the restore process. It appears second in the file name of the detail BRRESTORE log. See [Names of the Detail Logs \[Page 452\]](#).
- *Program response:* None.
- *User action:* None.

BR410I Tablespaces for restore: %s

- *Explanation:* List of tablespaces for restore. When brrestore -all or brrestore -full is specified, all the tablespaces are restored. See [-m|-mode \[Page 427\]](#).
- *Program response:* None.
- *User action:* None.

BR411I Database files for restore: %s

- *Explanation:* List of the database files for restore.
- *Program response:* None.
- *User action:* None.

Messages from BR384E to BR429W**BR412I Non-database files for restore: %s**

- *Explanation:* List of non-database files for restore.
- *Program response:* None.
- *User action:* None.

BR413I Directories for restore: %s

- *Explanation:* List of directories for restore.
- *Program response:* None.
- *User action:* None.

**BR414I Offline redo log file/s for restore from database instance %s:
%s**

- *Explanation:* List of offline redo log files for restore. The log sequence numbers of these files are displayed. BRRESTORE was called with one of the options `-a` | `-a1` | `-a2`.
- *Program response:* None.
- *User action:* None.

BR415I Files for restore using backup utility ID %s: %s

- *Explanation:* List of the files for restore using an external backup program.
- *Program response:* None.
- *User action:* None.

BR416I %d files found to restore, total size %4.3f

- *Explanation:* The total number and the total size of the files to restore is displayed.
- *Program response:* None.
- *User action:* None.

BR417E No files found for restoring.

- *Explanation:* No valid files were found for restoring.
- *Program response:* Processing terminated.
- *User action:* Correct the call options for BRRESTORE.

BR418I %d of %d files restored - %4.3f MB of %4.3f MB done

- *Explanation:* Indicates the current status of the restore process.
- *Program response:* None.
- *User action:* None.

BR419I Files will be restored from backups: %s

- *Explanation:* The name of the log of the BRBACKUP backup is displayed.
- *Program response:* None.
- *User action:* None.

BR420I Files will be restored from directories: %s

Messages from BR384E to BR429W

- *Explanation:* The names of the directories where the BRBACKUP backup was stored (involves a backup to disk) are displayed.
- *Program response:* None.
- *User action:* None.

BR421I Backup device type for restore: %s

- *Explanation:* The type of backup devices that will be used for the restore is displayed.
- *Program response:* None.
- *User action:* None.

BR422I File for restore directly from tape volume: %s

- *Explanation:* The position of a file on tape or its symbolic name (init_ora, init_sap, det_log, sum_log, reorg_log) is displayed. BRRESTORE was called with option -n.
- *Program response:* None.
- *User action:* None.

BR423I Files will be decompressed.

- *Explanation:* Confirms that the files will be decompressed during the restore. The backup that is being used for the restore was created using software compression (`compress = yes`).
- *Program response:* None.
- *User action:* None.

BR424I Files will not be decompressed.

- *Explanation:* Confirms that the files will not be decompressed during the restore. The backup that is being used for the restore did not use software compression (`compress = no`).
- *Program response:* None.
- *User action:* None.

BR425I Files will be decompressed by hardware.

- *Explanation:* Confirms that the files will be decompressed by the hardware during the restore. The backup that is being used for the restore was created using hardware compression (`compress = hardware`).
- *Program response:* In this case, BRRESTORE has no influence on the decompression operation. It proceeds as if no compression was used.
- *User action:* Make sure that the backup device is configured properly.

BR426W Directories do not need to be decompressed.

- *Explanation:* Although software decompression is active, the directory contents will not be decompressed, since they were not compressed during the backup.
- *Program response:* None.
- *User action:* None.

Messages from BR384E to BR429W

BR427W All files will be restored to the new Oracle home directory!

- *Explanation:* The current settings for ORACLE_HOME and ORACLE_SID are different than those during the backup.
- *Program response:* The new values have priority.
- *User action:* This method can be used to create a copy of the database. Make sure you know which other steps are required.

BR428W File %s will be overwritten.

- *Explanation:* An existing file will be overwritten during the restore operation.
- *Program response:* None.
- *User action:* None.

BR429W Directory %s will be overwritten.

- *Explanation:* An existing directory will be overwritten during the restore operation.
- *Program response:* None.
- *User action:* None.

Messages from BR430E to BR464W

BR430E Start BRRESTORE either with '%s' or '%s' option.

- *Explanation:* BRRESTORE was called with contradictory options.
- *Program response:* Processing terminated.
- *User action:* Use only one of the displayed options.

BR431E BRRESTORE with option '%s' currently running or was killed.

- *Explanation:* A BRRESTORE process is already running or was killed.
- *Program response:* BRRESTORE is not started.
- *User action:* If another BRRESTORE process is running, you will have to wait until it is finished. If a BRRESTORE run was terminated, refer to the other messages.

BR432I Please delete the file %s if BRRESTORE was killed.

- *Explanation:* This message appears together with BR431E.
- *Program response:* BRRESTORE is not started.
- *User action:* Check whether BRRESTORE is running (e.g. using the UNIX command `ps`). If not, delete the specified file and start BRRESTORE again.

BR433E Setting BRRESTORE lock failed.

- *Explanation:* This message appears together with BR431E.
- *Program response:* BRRESTORE processing terminates because the lock cannot be set.
- *User action:* See BR431E and BR432I.

BR435E Last successful backup not found in %s

- *Explanation:* The requested backup could not be found in the BRBACKUP summary log.
- *Program response:* Processing terminated.
- *User action:* Use option `-b` to specify the backup to use for the restore operation.

BR436W Backup %s is not a full database backup.

- *Explanation:* BRRESTORE was started with parameter `backup_mode = all|full` or option `-m all|full` (`all` is the default value), but the selected backup was not a complete backup of the database.
- *Program response:* All the files saved during this backup are restored (`full`). When `all` is used, no redo log files or control file are restored.
- *User action:* Check whether you really want to use this backup.

BR437W File %s to be restored was not backed up.

BR437E File %s to be restored was not backed up.

- *Explanation:* The file to be restored was not found in the selected backup.

Messages from BR430E to BR464W

- *Program response:* Processing is continued (BR437W) or terminated (BR437E).
- *User action:* Check the parameters `backup_mode` or option `-m` and correct your entries when necessary.

BR438W Redo log sequence number %d not found in %s

- *Explanation:* The archived redo log files to be restored were not found in the summary BRARCHIVE log.
- *Program response:* The offline redo log file with this log sequence number is not restored.
- *User action:* Check the options used `-a` | `-a1` | `-a2` and correct them when necessary.

BR439E No offline redo log files found in %s for restoring.

- *Explanation:* None of the offline redo log files to be restored were found in the summary BRARCHIVE log.
- *Program response:* Processing terminated.
- *User action:* Check the options used `-a` | `-a1` | `-a2` and correct them when necessary.

BR440E Unknown database instance %s defined in option %s.

- *Explanation:* The name of the database instance specified in the BRRESTORE option `-a` | `-a1` | `-a2` is unknown.
- *Program response:* Processing is terminated.
- *User action:* Check the option `-a` | `-a1` | `-a2` used and correct it if necessary.

BR441E Database instance %s not unique in option %s.

- *Explanation:* The name of the database instance specified in the BRRESTORE option `-a` | `-a1` | `-a2` is not unique.
- *Program response:* Processing is terminated.
- *User action:* Check the option `-a` | `-a1` | `-a2` used and correct it if necessary.

BR442I Second copy of offline redo log files will be used for restore.

- *Explanation:* BRRESTORE was called with the option `-a2` | `-archive2`, in order to restore offline redo log files.
- *Program response:* None.
- *User action:* None.

BR451E Message %s not found in %s

- *Explanation:* One of the required messages of the BRBACKUP/BRARCHIVE log header was not found. This could indicate an internal error, or the log was changed manually.
- *Program response:* Processing terminated.
- *User action:* If the log was changed manually, change it back.

BR452E Continuation of message %s not found in %s

Messages from BR430E to BR464W

- *Explanation:* The continuation of a message that extends over several lines was not found in the BRBACKUP/BRARCHIVE log file. This could indicate an internal error, or the log was changed manually.
- *Program response:* Processing terminated.
- *User action:* If the log was changed manually, change it back.

BR453W Error message found in %s:

- *Explanation:* The error message was not found in the BRBACKUP/BRARCHIVE log. This could affect the restore operation.
- *Program response:* The message is ignored and the operation is continued.
- *User action:* Analyze the message and decide whether you want to continue (message BR256I is displayed).

BR454W Values of %s are different: current '%s', backup %s '%s'.**BR454E Values of %s are different: current '%s', backup %s '%s'.**

- *Explanation:* The current parameter value is different from the value when the backup was made.
- *Program response:* Processing is continued (BR454W) or terminated (BR454E).
- *User action:* Some values are automatically adapted by BRRESTORE, others have to be adapted manually. Start BRRESTORE again with the adapted parameter values.

BR455W Value '%s' of %s will be used for restore.

- *Explanation:* The current parameter value is different from the value when the backup was made.
- *Program response:* BRRESTORE automatically adapts the value.
- *User action:* Decide whether you want to accept the adapted parameter value (message BR256I is displayed).

BR456W Probably the database must be recovered due to restore from online backup!

- *Explanation:* BRRESTORE reports that an online backup was selected.
- *Program response:* None.
- *User action:* The database must be recovered after you restore an online backup (ORACLE recovery required).

BR457W Probably the database must be recovered due to partial restore!

- *Explanation:* BRRESTORE reports that a partial restore was selected.
- *Program response:* None.
- *User action:* The database must be recovered after a partial restore (ORACLE recovery required).

BR458E Information about backup file %s not found in log header.

- *Explanation:* All the backup files must be described in the log header of the selected BRBACKUP backup. This is not the case. This could indicate an internal error, or the log was changed manually.

Messages from BR430E to BR464W

- *Program response:* Processing terminated.
- *User action:* If the log was changed manually, change it back and repeat the operation.

BR459W Backup %s terminated with errors.

- *Explanation:* The backup terminated with errors. This could influence the restore operation.
- *Program response:* Waits for confirmation to continue the operation.
- *User action:* Analyze the message and decide whether you want to continue processing.

BR460W Termination message not found in %s - log file incomplete.

- *Explanation:* The termination message was not found in the backup log.
 - Case 1: The log was restored from a volume.
 - Case 2: You manually changed the log.
- *Program response:* Waits for confirmation to continue the operation.
- *User action:* Case 1: Continue processing.
Case 2: Reset your changes, cancel the processing, and then repeat the operation.

BR461W Redo log sequence number in line %d in %s not ascending.

- *Explanation:* The log sequence numbers recorded in the summary BRARCHIVE log must be stored in ascending order. If not, this means that a RESET LOGS has been performed, which is normal after a partial recovery.
- *Program response:* All offline redo log files that were created before the RESET LOGS are ignored.
- *User action:* Analyze the message and decide whether you want to continue processing.

BR462W All lines in %s till line %d will be ignored!

- *Explanation:* Appears together with BR461X when a RESET LOGS has been performed. The log sequence numbers have been assigned twice.
- *Program response:* All offline redo log files that were created before the RESET LOGS are ignored.
- *User action:* Analyze the message and decide whether you want to continue processing.

BR464W Line %d in %s ignored because of different backup device types.

- *Explanation:* Offline redo log files can only be restored when the device types used for the backup and the restore are compatible (the following combinations of `backup_dev_type` parameters are illegal, for example: `tape` and `util_file`, `pipe` and `util_file`. In contrast, the combination `tape` and `pipe` is allowed).
- *Program response:* Processing terminated.
- *User action:* Correct the current value of parameter and repeat the operation.

Messages from BR465E to BR468I

BR465E Wrong value of %s for the first offline redo log file found.

- *Explanation:* The parameter value during the backup of the first offline redo log file to restore and the current value (from profile `init<DBSID>.sap` or provided using the appropriate call option) for the file to be restored are contradictory. This message appears together with BR464W.
- *Program response:* Processing terminated.
- *User action:* Correct the current value of the parameter and repeat the operation.

BR466I Start BRRESTORE one after the other for each backup device type!

- *Explanation:* Offline redo log files whose specifications for the backup devices are not compatible must be restored in separate BRRESTORE runs. This message appears together with BR464W.
- *Program response:* Processing terminated.
- *User action:* Start BRRESTORE with the required options individually for each required backup device.

BR467W Line %d in %s ignored because of different compress types.

- *Explanation:* Offline redo log files can only be restored if the compression type used during backup is compatible with that specified for restoring.
- *Program response:* Processing is terminated.
- *User action:* Correct the current value of the parameter and repeat processing.

BR468I Start BRRESTORE one after other for each compress type.

- *Explanation:* Offline redo log files whose compression type definitions are not compatible must be restored in separate BRRESTORE runs. This message appears in connection with BR467W.
- *Program response:* Processing is terminated.
- *User action:* Start BRRESTORE with the requested options individually for the requested compression types.

BR501I BACKINT %s %s

- *Explanation:*
- *Program response:* Processing terminated.
- *User action:*

BR502I BACKINT terminated successfully.

- *Explanation:*
- *Program response:* Processing terminated.
- *User action:*

Messages from BR465E to BR468I

BR503I BACKINT terminated with warnings.

- *Explanation:*
- *Program response:* Processing terminated.
- *User action:*

BR504I BACKINT terminated with errors.

- *Explanation:*
- *Program response:* Processing terminated.
- *User action:*

BR505I End of BACKINT processing: %s

- *Explanation:*
- *Program response:* Processing terminated.
- *User action:*

BR506I Error reading init_utl profile %s

- *Explanation:*
- *Program response:* Processing terminated.
- *User action:*

BR507I %s call failed.

- *Explanation:*
- *Program response:* Processing terminated.
- *User action:*

BR508I Processing of the link definition file %s failed.

- *Explanation:*
- *Program response:* Processing terminated.
- *User action:*

BR509I The above message is %s error message.

- *Explanation:*
- *Program response:* Processing terminated.
- *User action:*

BR510I File %s not processed by %s.

- *Explanation:*
- *Program response:* Processing terminated.
- *User action:*

BR511I Unknown file %s processed by %s.

- *Explanation:*

- *Program response:* Processing terminated.
- *User action:*

BR512I Link definition for %s not found in %s

- *Explanation:*
- *Program response:* Processing terminated.
- *User action:*

BR513I Backup of %s not found for backup ID %s

- *Explanation:*
- *Program response:* Processing terminated.
- *User action:*

BR514I File %s never saved by %

- *Explanation:*
- *Program response:* Processing terminated.
- *User action:*

Messages from BR501I to BR607E

Messages from BR501I to BR607E

- **BR501I Starting normal database backup using RMAN...**
Explanation: A normal (non-incremental, non-full) backup of the database is started using RMAN.
- *Program response:* none
- *User action:* none

BR502I Normal database backup using RMAN successful

- *Explanation:* A normal (non-incremental, non-full) backup of the database using RMAN completed successfully.
- *Program response:* none
- *User action:* none

BR503E Normal database backup using RMAN failed

- *Explanation:* A normal (non-incremental, non-full) backup of the database using RMAN failed.
- *Program response:* Processing terminates
- *User action:* Read any other RMAN error messages that give you information about the cause of the failure and repeat the action.
- **BR504I Starting full database backup (level 0) using RMAN...**
Explanation: A full backup of the database (level 0) is started using RMAN.
- *Program response:* none
- *User action:* none

BR505I Full database backup (level 0) using RMAN successful

- *Explanation:* A full backup of the database (level 0) using RMAN completed successfully.
- *Program response:* none
- *User action:* none

BR506E Full database backup (level 0) using RMAN failed

- *Explanation:* A full backup of the database (level 0) using RMAN failed.
- *Program response:* none
- *User action:* none

BR507I Starting incremental database backup (level 1) using RMAN...

- *Explanation:* An incremental backup of the database is started using RMAN.
- *Program response:* none
- *User action:* none

BR508I Incremental database backup (level 1) using RMAN successful

Messages from BR501I to BR607E

- *Explanation:* An incremental backup of the database using RMAN completed successfully.
- *Program response:* none
- *User action:* none

BR509E Incremental database backup (level 1) using RMAN failed

- *Explanation:* An incremental backup of the database using RMAN failed.
- *Program response:* none
- *User action:* none

BR510I Starting full backup for new database files using RMAN...

- *Explanation:* A full backup of the new database files within the incremental backup is started using RMAN.
- *Program response:* none
- *User action:* none

BR511I Full backup for new database files using RMAN successful

- *Explanation:* A full backup of the new database files within the incremental backup using RMAN completed successfully.
- *Program response:* none
- *User action:* none

BR512E Full backup for new database files using RMAN failed

- *Explanation:* A full backup of the new database files within the incremental backup using RMAN failed.
- *Program response:* none
- *User action:* none

BR513I Last full database backup (level 0): %s

- *Explanation:* Output: The last full backup of the database (level 0), used as a reference by the incremental backup.
- *Program response:* none
- *User action:* none

BR514I Following files created after the last full database backup (level 0) will be fully backed up:

- *Explanation:* Output: The database files created since the last full backup of the database (level 0).
- *Program response:* none
- *User action:* none

BR515E Last full database backup (level 0) not found

Messages from BR501I to BR607E

- *Explanation:* An attempt was made to start an incremental backup without a full backup of the database (level 0) having been made previously.
- *Program response:* Processing terminates
- *User action:* Make a full backup of the database.

BR516E Last full database backup (level 0) %s was not successful

- *Explanation:* The last full backup of the database (level 0) was not successful.
- *Program response:* Processing terminates
- *User action:* Make a full backup of the database.

BR517E Last full database backup (level 0) %s was killed or restored

- *Explanation:* The last full backup of the database (level 0) was terminated or restored.
- *Program response:* Processing terminates
- *User action:* Make a full backup of the database.

BR518E Last full database backup (level 0) %s older than the value of control_file_record_keep_time

- *Explanation:* The last full backup of the database (level 0) is older than the value of the init<SID>.ora parameter control_file_record_keep_time.
- *Program response:* Processing terminates
- *User action:* Make a full backup of the database.

BR519E Redo log sequence number %d of the last full backup (level 0) %s not found in table V\$LOG_HISTORY

- *Explanation:* The start redo log sequence number of the last full backup of the database (level 0) was not found in the table V\$LOG_HISTORY. This prevents the incremental backup from being used for automatic recovery.
- *Program response:* Processing terminates
- *User action:* Make a full backup of the database.

BR520I Please start a full database backup (level 0) first

- *Explanation:* The previous error messages indicate that you have to make a full backup of the database (level 0).
- *Program response:* Processing terminates
- *User action:* Make a full backup of the database.

BR521I Database file %s included in saveset: %s

- *Explanation:* This database file is backed up in the saveset with this ID.
- *Program response:* none
- *User action:* none

BR522I %d file/saveset processed by RMAN**BR522E %d files/savesets processed by RMAN**

- *Explanation:* Output: The number of files per saveset processed by RMAN.

Messages from BR501I to BR607E

- *Program response:* BR522I: none; BR522E: Processing terminates
- *User action:* BR522I: none; BR522E: Read the previous error messages and repeat the action.

BR523I Preparation for using RMAN only - no backup

- *Explanation:* An RMAN preparation run is started (`backup_dev_type = rman_prep`). Database files are not backed up.
- *Program response:* none
- *User action:* none

BR524E Preparation of database backup using RMAN failed

- *Explanation:* RMAN preparation failed.
- *Program response:* Processing terminates
- *User action:* Read the previous error messages and repeat the action.

BR525I Preparing backup of %s using RMAN...

- *Explanation:* RMAN preparation run for this file started (`backup_dev_type = rman_prep`).
- *Program response:* none
- *User action:* none

BR526E Preparation of backup of %s using RMAN failed

- *Explanation:* RMAN preparation run for this file failed.
- *Program response:* Processing terminates
- *User action:* Read the previous error messages and repeat the action.

BR527I Savesets with %s members:

- *Explanation:* Output: The elements of the savesets determined by the RMAN preparation run (`backup_dev_type = rman_prep`). Back up the description of the BRBACKUP log files.
- *Program response:* none
- *User action:* none

BR528E Saveset information not available for %s

- *Explanation:* An attempt was made to start RMAN preparation for a selection of the database files, although there is no saveset information for some of the files not selected. No savesets can be created.
- *Program response:* Processing terminates
- *User action:* Start RMAN preparation for the entire database (see message BR529I).

BR529I Run BRBACKUP with option '-m all' to update information about all savesets

- *Explanation:* After message BR528E: Start RMAN preparation for the entire database.

Messages from BR501I to BR607E

- *Program response:* Processing terminates
- *User action:* Start RMAN preparation for the entire database.

BR530I Cataloging backups of all database files...

- *Explanation:* The database was backed up successfully without RMAN (level 0); the files are now cataloged.
- *Program response:* none
- *User action:* none

BR531I Backups of all database files cataloged successfully

- *Explanation:* Database files of the full backup (level 0) without RMAN were cataloged successfully.
- *Program response:* none
- *User action:* none

BR532E Cataloging backups of all database files failed

- *Explanation:* Cataloging of the database files of the full backup (level 0) without RMAN failed.
- *Program response:* Processing terminates
- *User action:* Read the previous error messages and repeat the action.

BR533I Uncataloging backups of database files...

- *Explanation:* RMAN preparation was started (`backup_dev_type = rman_prep`). After all database files have been processed successfully, the information on the preparation is deleted from the control file (uncataloged).
- *Program response:* none
- *User action:* none

BR534I Backups of database files uncataloged successfully

- *Explanation:* Database files of the RMAN preparation run were uncataloged successfully.
- *Program response:* none
- *User action:* none

BR535E Uncataloging backups of database files failed

- *Explanation:* Uncataloging of the database files of the RMAN preparation run failed.
- *Program response:* Processing terminates
- *User action:* Read the previous error messages and repeat the action.

BR536E RMAN call for database instance %s failed

- *Explanation:* The RMAN call to back up, restore, verify, catalog or uncatalog database files failed.
- *Program response:* Processing terminates

Messages from BR501I to BR607E

- *User action:* Read the previous error messages and repeat the action.

BR537E BR\$BTLIB call %s failed

- *Explanation:* This SAP backup library call failed.
- *Program response:* Processing terminates
- *User action:* Read the previous error messages and repeat the action.

BR538E Copying of data by BR\$BTLIB failed

- *Explanation:* The SAP backup library call to copy data failed.
- *Program response:* Processing terminates
- *User action:* Read the previous error messages and repeat the action.

BR539E Internal compression failed, return code: %d

- *Explanation:* Compression failed in the RMAN preparation run.
- *Program response:* Processing terminates
- *User action:* Contact SAP.

BR540I Saveset piece %d saved

- *Explanation:* BRBACKUP used the parameter `rman_kbytes` to save a piece of a saveset to a local or remote disk.
- *Program response:* none
- *User action:* none

BR551I Starting restore from normal database backup using RMAN...

- *Explanation:* Restoration of files from a normal (non-incremental, non-full) backup of the database started.
- *Program response:* none
- *User action:* none

BR552I Restore from normal database backup using RMAN successful

- *Explanation:* Restoration of files from a normal (non-incremental, non-full) backup of the database completed successfully.
- *Program response:* none
- *User action:* none

BR553E Restore from normal database backup using RMAN failed

- *Explanation:* Restoration of files from a normal (non-incremental, non-full) backup of the database failed.
- *Program response:* Processing terminates
- *User action:* Read the previous error messages and repeat the action.

BR554I Starting restore from full database backup using RMAN...

- *Explanation:* Restoration of files from a full backup of the database (level 0) started.

Messages from BR501I to BR607E

- *Program response:* none
- *User action:* none

BR555I Restore from full database backup using RMAN successful

- *Explanation:* Restoration of files from a full backup of the database (level 0) completed successfully.
- *Program response:* none
- *User action:* none

BR556E Restore from full database backup using RMAN failed

- *Explanation:* Restoration of files from a full backup of the database (level 0) failed.
- *Program response:* Processing terminates
- *User action:* Read the previous error messages and repeat the action.

BR557I Starting restore from incremental database backup using RMAN...

- *Explanation:* Restoration of files from an incremental database backup started.
- *Program response:* none
- *User action:* none

BR558I Restore from incremental database backup using RMAN successful

- *Explanation:* Restoration of files from an incremental database backup completed successfully.
- *Program response:* none
- *User action:* none

BR559E Restore from incremental database backup using RMAN failed

- *Explanation:* Restoration of files from an incremental database backup failed.
- *Program response:* Processing terminates
- *User action:* Read the previous error messages and repeat the action.

BR560I Restoring database file %s from saveset: %s

- *Explanation:* Database file restored from this saveset.
- *Program response:* none
- *User action:* none

BR561W Backup %s is not a full database backup (level 0)**BR561E Backup %s is not a full database backup (level 0)**

- *Explanation:* This backup is not a full backup (level 0) of the database.
- *Program response:* BRRESTORE: Warning, processing continues; BRBACKUP: Error message, processing terminates.
- *User action:* Use another backup mode (parameter `backup_mode` or option `-m`).

BR562E Backup %s is not an incremental database backup (level 1)

Messages from BR501I to BR607E

- *Explanation:* This backup is not an incremental database backup.
- *Program response:* Processing terminates
- *User action:* Use another backup mode (parameter `backup_mode` or option `-m`).

BR563E Run BRRESTORE with option '-m incr|incr_only|incr_full' to restore an incremental database backup

- *Explanation:* An attempt was made to restore an incremental database backup with an incorrect value for the option `-m`.
- *Program response:* Processing terminates
- *User action:* Use one of the proposed values for the option `-m` to restore an incremental backup.

BR564W File %s cannot be restored/verified with other data files

- *Explanation:* Control files, online redo log files, non-database files and directories cannot be restored in the same run in which RMAN is restoring other database files, if the backup device type was used (`backup_dev_type`) `rman_util`.
- *Program response:* The files are not restored.
- *User action:* Restore these files in a separate run. Shut down the database instance when you do this.

BR565W Due to running database instance control file cannot be restored during restore using RMAN

- *Explanation:* Control files cannot be restored in the same run in which RMAN is restoring other database files, if the backup device type was used (`backup_dev_type`) `rman_util`.
- *Program response:* The files are not restored.
- *User action:* Restore these files in a separate run. Shut down the database instance when you do this.

BR566I Stop the database instance and run BRRESTORE with option '-m 0' to restore the control file separately

- *Explanation:* Control files cannot be restored in the same run in which RMAN is restoring other database files, if the backup device type was used (`backup_dev_type`) `rman_util`.
- *Program response:* The files are not restored.
- *User action:* Restore these files in a separate run. Shut down the database instance when you do this.

BR567I Please stop the database and restore these files in a separate run

- *Explanation:* This message appears after 564W.
- *Program response:* none
- *User action:* Obey the message.

Messages from BR501I to BR607E**BR568I Verifying backup of %s using RMAN**

- *Explanation:* Verification of the backup of a database file using RMAN started.
- *Program response:* none
- *User action:* none

BR569I Verification of backup of %s using RMAN successful

- *Explanation:* Verification of the backup of a database file using RMAN completed successfully.
- *Program response:* none
- *User action:* none

BR570E Verification of backup of %s using RMAN failed

- *Explanation:* Verification of the backup of a database file using RMAN failed.
- *Program response:* Processing terminates
- *User action:* Read the previous error messages and repeat the action.

BR571I Starting verification of normal database backup using RMAN...

- *Explanation:* Verification of a normal backup of the database using RMAN started.
- *Program response:* none
- *User action:* none

BR572I Verification of normal database backup using RMAN successful

- *Explanation:* Verification of a normal backup of the database using RMAN completed successfully.
- *Program response:* none
- *User action:* none

BR573E Verification of normal database backup using RMAN failed

- *Explanation:* Verification of a normal backup of the database using RMAN failed.
- *Program response:* Processing terminates
- *User action:* Read the previous error messages and repeat the action.

BR574I Starting verification of full database backup using RMAN...

- *Explanation:* Verification of a full backup of the database using RMAN started.
- *Program response:* none
- *User action:* none

BR575I Verification of full database backup using RMAN successful

- *Explanation:* Verification of a full backup of the database using RMAN completed successfully.
- *Program response:* none
- *User action:* none

Messages from BR501I to BR607E

BR576E Verification of full database backup using RMAN failed

- *Explanation:* Verification of a full backup of the database using RMAN failed.
- *Program response:* Processing terminates
- *User action:* Read the previous error messages and repeat the action.

BR577I Starting verification of incremental database backup using RMAN...

- *Explanation:* Verification of an incremental backup of the database using RMAN started.
- *Program response:* none
- *User action:* none

BR578I Verification of incremental database backup using RMAN successful

- *Explanation:* Verification of an incremental backup of the database using RMAN completed successfully.
- *Program response:* none
- *User action:* none

BR579E Verification of incremental database backup using RMAN failed

- *Explanation:* Verification of an incremental backup of the database using RMAN failed.
- *Program response:* Processing terminates
- *User action:* Read the previous error messages and repeat the action.

BR580I Verification of saveset: %s

- *Explanation:* Verification of this saveset started.
- *Program response:* none
- *User action:* none

BR581I Saveset piece %d restored

- *Explanation:* BRBACKUP has restored a piece of the saveset generated with the parameter `rman_kbytes`.
- *Program response:* none
- *User action:* none

BR601E Determination of hardware ID failed, return code: %d

- *Explanation:* Determination of hardware ID failed. This ID is needed to check the license.
- *Program response:* Processing terminates
- *User action:* Contact SAP.

BR602E No valid SAP license found - please contact SAP

Messages from BR501I to BR607E

- *Explanation:* No valid license could be found.
- *Program response:* Processing terminates
- *User action:* Contact SAP.

BR603I Temporary SAP license valid only till %s

- *Explanation:* The license has expired.
- *Program response:* none
- *User action:* Contact SAP.

BR604W SAP license expires in %d days

- *Explanation:* The license expires soon.
- *Program response:* none
- *User action:* Contact SAP.

BR605E Used functionality is not covered by the current SAP license

- *Explanation:* The functions you have used are not covered by your existing license.
- *Program response:* Processing terminates
- *User action:* Contact SAP.

BR606W The database size exceeds the SAP license limits

- *Explanation:* The size of the database exceeds that allowed by the license.
- *Program response:* none
- *User action:* Contact SAP.

BR607E SAP license has already expired - please contact SAP

- *Explanation:* The functions you have used are not covered by your existing license.
- *Program response:* Processing terminates
- *User action:* Contact SAP.

Database Reorganization

A reorganization is a time-consuming and complex process and should therefore be performed infrequently and only when absolutely necessary.

See [Reorganization Concepts \[Page 576\]](#)

This section explains how to use the **Reorganization** option of program SAPDBA to reorganize the database.

See [SAPDBA: Reorganization \[Page 602\]](#)

For detailed information on carrying out a reorganization.

See [Reorganization \[Page 603\]](#)

To find out how to repeat a terminated reorganization.

See [Restarting a Reorganization \[Page 628\]](#)

Reorganization Concepts

Reorganization Concepts

1. Read about the [Status of the Database \[Page 577\]](#).
2. Note the [Limitations of the Database System \[Page 579\]](#).
3. Choose the correct [Reorganization Type \[Page 581\]](#).
4. Before you start the reorganization, examine the [Backup Concepts in Reorganizations \[Page 595\]](#) recommended by SAP.
5. Read about the [Procedure for a Reorganization \[Page 592\]](#).

For more information, see [Procedures in Detail \[Page 594\]](#).

Status of the Database

When installed, the database of the R/3 System has the following initial status:

- Most of the tables and indexes of a tablespace are stored in only one extent.
- Each tablespace consists of exactly one data file.

Changes to the Database

When data is added to the database, its status changes. The following types of change can occur:

- **Additional extents** are created for the tables and indexes of a tablespace, to make space for further data.
This may result in poorer data access times.
- When a tablespace is full (there is not enough freespace to create a new extent), **additional data files** must be added to it.
- Adding extents or deleting complete tables causes **freespace fragmentation** (external fragmentation) in a tablespace. Free storage space in data files is divided into smaller units.

If these are smaller than one requested extent, the space is lost, and cannot be used for storing data.

This problem does not occur as frequently as it used to, because the ORACLE system monitor merges at adjacent least areas of free space automatically.

- An **internal fragmentation** occurs if the fill level of the database blocks develops differently. The fill level of the individual blocks is initially identical. Creating and deleting data causes some blocks to be filled completely, where as others are relatively empty. As a result, space is used inefficiently.
- If a data record does not fit into a database block, **block chaining** occurs. The database must then follow a chain from the first block of the data record to the further blocks. As a result, more time is needed for reading data from the disk.

Since the R/3 System usually accesses a table entry using an index, the above changes to the database usually have no negative influence on the time required to access data. However, the changed database status may cause increased run times in connection with full table scans.

Effects of a Reorganization

A reorganization may have the following effects:

- The data from one object is merged in one extent.
- The data from one tablespace is merged in one or more data files.
- Freespace fragments in an object are merged into larger freespace segments (defragmentation).
- The fill level in the individual blocks is evened out (reduction of internal fragmentation). This may result in increased space requirements.
- Data chains are resolved.

Status of the Database

Limitations of the Database System

Reorganization may also be necessary to avoid problems related to the limitations of the database system or operating system. This can involve the following limitations:

- **Number of extents per table/index (MAXEXTENTS)** - software limit.
Default: 100 extents.

The SAP tables and indexes are created with the parameter MAXEXTENTS=100, which specifies a maximum number of 100 extents (with a block size of 8 KB). If a table or an index reaches this limit, the database cannot create an additional extent for that object. In this case you should first increase the maximum number of extents to keep the system executable. You are recommended to increase the NEXT parameter ([Command Option - next \[Page 613\]](#)) to avoid reaching the newly set soft limit.

See: [Changing MAXEXTENTS and NEXT \[Page 612\]](#).

A reorganization of the object should only take place in an emergency.

- **No. of extents per table/index (MAXEXTENTS)** - hard limit.
Standard: 505 extents.

This limit only exists for ORACLE releases < 7.3.

At 505 extents the database system reaches a final, unchangeable limit for the no. of extents of an object. This limit depends on the block size of the database (the value of 505 extents applies to the block size 8KB). If you reach this limit the affected table or index must be reorganized.

Block size	Maximum for MAXEXTENTS	SAP-value for MAXEXTENTS
2 KB	121	100
4 KB	249	100
8 KB	505	100



The standard setting for parameter MAXEXTENTS = 100 should enable the administrator to counter a great increase in the number of extents in good time. This prevents you from reaching the hard limit and avoids reorganization.

- **Maximum number of files per database** - soft limit.
SAP default: 254 files.

The database system only supports a specific number of data files in the database, dependent on the host system. If your database approaches this limit, you could reduce the number of data files by reorganizing tablespaces that have more than one file.

This is not likely to occur, however, as up to 254 data files can be created (R/3 System default after installation). The absolute upper limit (hard limit) is a maximum of 1022 data files, depending on operating system (AIX: 256 files; all other UNIX systems: 1024 files; ORACLE itself requires two files).

The threshold value for data files is specified by parameter db_files in profile init<DBSID>.ora. It can be increased to a maximum of MAXDATAFILES, the value

Limitations of the Database System

of which was specified when the database was created. MAXDATAFILES itself should be less than the permissible maximum number of open files supported by the operating system. The default setting for both these parameters after installation is 254.

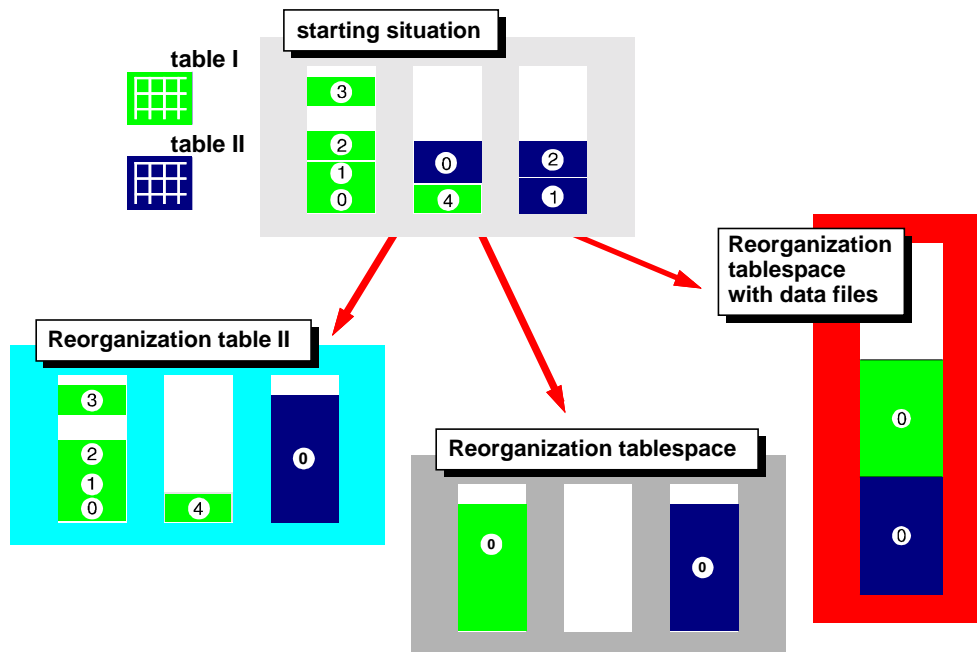


Because the possible number of data files is quite large and, as a result, will not be reached very quickly, reorganizations with the aim of reducing the number of data files should not be performed unnecessarily often.

Reorganization Type

There are several types of database reorganization. Select the correct reorganization type - for example, if a specific table is causing space problems, reorganize only that table, and not the entire tablespace.

The following figure shows the different reorganization types. It is based on a tablespace consisting of three data files with two fragmented tables and a fragmented freespace.



- **Table/index reorganization:** The quickest and simplest type of reorganization is the reorganization of an individual table or an index in a tablespace.

This type of reorganization merges the extents occupied by the table or the index to one extent (using the default value *compress extents: yes*). In addition, adjacent free storage fragments in the entire tablespace are merged to form one closed area. The Oracle database system does this with its own system monitor.

Depending on the tablespace fragmentation, a table/index reorganization with the option *Compress extents: yes* may not be possible, even though the total freespace is larger than the table or the index. SAPDBA can detect and display this situation.

See [Reorganization of a Single Object \[Page 583\]](#).

- **Reorganization of a list of tables and indexes:** Specification of an ASCII file (list name) containing one object per line ([owner.]<object name>) that is to be reorganized. If another owner is not specifically defined, **sapr3** is the owner of these objects. This ASCII file must have been created in the working directory.

Each object listed in the list file is reorganized as described in *Table/index reorganization*.

See [Reorganization of a List of Objects \[Page 584\]](#).

Reorganization Type

- **Tablespace reorganization:** All tables and all indexes of a tablespace are merged to one extent per object. The freespace in the tablespace is also merged in one fragment per data file (not when using the options *Create table ... as select* or *Index rebuild*).
See [Reorganization of a Tablespace \[Page 586\]](#).
- **Tablespace reorganization with data files:** In addition to the simple tablespace reorganization, this reorganization merges the data files of a tablespace to one data file, if the resulting file is smaller than 2 GB (limit of most operating systems). Otherwise SAPDBA recommends writing the contents of the old files to a number of files of equal size. This number is the minimum number of files required to guarantee that each single file size is smaller than 2 GB. All files are written to the same directory (name in accordance with the SAP naming conventions).



The old data files (5 files) have a total of 4.8 GB. SAPDBA recommends creating three new data files with 1.6 GB each. You can also change this default value, for example to 2x1.9 GB and 1x1.0 GB or 4x1.2 GB etc.

This reorganization type is the most complicated, and should only be performed rarely - if at all.

See [Reorganization of a Tablespace with Data Files \[Page 588\]](#).

- **Moving/renaming the data files of a tablespace:** If you only want to change the data file allocation for a tablespace, you do not have to reorganize the corresponding tablespace. In this case, use the option *Move/rename data files of a tablespace*.

See [Moving/Renaming the Data Files of a Tablespace \[Page 590\]](#)

Defragmentation

SAPDBA always performs a defragmentation of the entire database, regardless of the type of reorganization. Once all the tables and indexes to be reorganized have been deleted, the freespace now existing for the entire database is defragmented. Free fragments which are physically directly next to each other are combined within all the data files. The deleted tables and indexes are then recovered.

Reorganization of a Single Object

SAPDBA actions to reorganize a table or index (script name):

Preparation Phase

1. Generates the `<timestamp>` subdirectory in the working directory and the log `<timestamp>.rsi`.
2. Generates the SQL and shell scripts and the restart file `restart.rsi` in the `<timestamp>` subdirectory (see [Restart File for Reorganizations \[Page 632\]](#)).
3. Checks whether the object for reorganization will fit in the freespace that will be available once the object has been deleted.

See [Freespace Check During Reorganization of a Single Object \[Page 598\]](#).

[SAPDBA: Reorganization Menu \[Page 615\]](#)

Main Phase

1. If necessary, temporarily renames the table to be reorganized or the table affected by the reorganization in order to save the data (`nam<TSP>.sql`, see *HideTab* option in [SAPDBA: Reorganization Menu \[Page 615\]](#)).
2. Exports the table (`exp<TSP>.sh` or `unl<TSP>.sh` and if necessary `exc<TSP>.sh`); not in the case of an index reorganization or when using *Create table... as select*.
3. If necessary, performs a test import to check the readability of the export dump files (`imx<TSP>.sh`, see *CheckExp* option).
4. Deletes the object (`dro<TSP>.sql`); not with *Create table... as select*; not in *Deferred* mode, here it renames the table temporarily instead.
5. Merges all the remaining adjoining freespace fragments of the entire database using Oracle functions.
6. Recovery of the table, with any comments; with *Create table ... as select* the old table is deleted in this step and the new table is renamed with the old name (`tab<TSP>.sql`).
7. Imports the table data (with any indexes, constraints and grants) (`imp<TSP>.sh` or `ldr<TSP>.sh` and maybe `imc<TSP>.sh`, `ind<TSP>.sql`, `con<TSP>.sql`, `grants.sql`); in an index reorganization only the index and maybe the constraint are recovered (`ind<TSP>.sql`, `con<TSP>.sql`).
8. Delete the renamed table, if it is in *Deferred* mode (see the option *DropTab* in [SAPDBA: Reorganization Menu \[Page 615\]](#)).
9. If necessary, sets back the name of the table which was temporarily renamed for data protection (see step 4) (`ren<TSP>.sql`, see *HideTab* option in [SAPDBA: Reorganization Menu \[Page 615\]](#)).

Reorganization of an Object List

Reorganization of an Object List

SAPDBA actions to reorganize tables/indexes which were defined in a list (script name):

Preparation Phase

1. Generates the **<timestamp> subdirectory** in the working directory and the log **<timestamp>.rli**.
2. Generates the SQL and shell scripts and the restart file **restart.rli** in the **<timestamp>** subdirectory (see [Restart File for Reorganizations \[Page 632\]](#)).
3. Checks whether the objects to be reorganized will fit in the freespace which will be available after the objects have been deleted.
 - The list comprises one object per tablespace:
See [Freespace Check During Reorganization of a Single Object \[Page 598\]](#).
 - The list comprises several objects per tablespace:
See [Freespace Check During Reorganization of a Tablespace \[Page 599\]](#).
1. [SAPDBA: Reorganization Menu \[Page 615\]](#)

Main Phase

1. If necessary, temporarily renames the tables to be reorganized or to be affected by the reorganization (**nam<TSP>.sql**, see *HideTab* option in the [SAPDBA: Reorganization Menu \[Page 615\]](#)).
2. Only when exporting to tape (due to interactive export mode):
Deletes all the index statistics (**alt<TSP>.sql**) as well as any constraints (**drc<TSP>.sql**).
3. Exports the tables (**exp<TSP>.sh** and if necessary **exc<TSP>.sh** or **unl<TSP>.sh** - not in the case of an index reorganization or with *Create table ... as select*); see also [Parallel Export/Import \[Page 643\]](#).
4. If necessary, performs a test import to check the readability of the export dump files (**imx<TSP>.sh**, see *CheckExp* option in the [SAPDBA: Reorganization Menu \[Page 615\]](#)).
5. Deletes the objects (**dro<TSP>.sql**); not with *Create table ... as select*.
6. Merges all the remaining adjoining freespace fragments of the entire database using Oracle functions.
7. Recovery of the tables, if necessary with comments. With *Create table ... as select* the old table is deleted in this step and the new table is renamed with the old name (**tab<TSP>.sql**).
8. Imports the table files (possibly with indexes, constraints, and grants). (**imp<TSP>.sh** or **ldr<TSP>.sh** and possibly **imc<TSP>.sh**, **ind<TSP>.sql**, **con<TSP>.sql**, **grants.sql**). Only the indexes and maybe the constraints are recreated if there is an index reorganization (**ind<TSP>.sql**, **con<TSP>.sql**).
9. If necessary, sets back the names of the tables which were temporarily renamed for data security (in step 1) (**ren<TSP>.sql**, see *HideTab* option in the [SAPDBA: Reorganization Menu \[Page 615\]](#)).

Reorganization of a Tablespace

Reorganization of a Tablespace

SAPDBA actions to reorganize a tablespace (without data file reorganization) (script name):

Preparation Phase

1. Generates the `<timestamp>` subdirectory in the working directory and the log `<timestamp>.rtc`.
2. Generates the SQL and shell scripts and the restart file `restart.rtc` in the `<timestamp>` subdirectory (see [Restart File for Reorganizations \[Page 632\]](#)).
3. Checks whether the extents to be reorganized fit in the data file(s) of the tablespace; see [Freespace Check During Reorganization of a Tablespace \[Page 599\]](#)
4. [SAPDBA: Reorganization Menu \[Page 615\]](#)

Main Phase

1. If necessary, temporarily renames the tables to be reorganized or to be affected by the reorganization (`nam<TSP>.sql`, see *HideTab* option in the [SAPDBA: Reorganization Menu \[Page 615\]](#)).
2. Only when exporting to tape (due to interactive export mode):
Deletes all the index statistics (`alt<TSP>.sql`) as well as any constraints (`drc<TSP>.sql`).
3. Exports all tables and indexes of the tablespace (`exp<TSP>.sh` or `unl<TSP>.sh` and if necessary `exc<TSP>.sh`; not with *Create table ... as select*); see also [Parallel Export/Import \[Page 643\]](#).
4. If necessary, performs a test import to check the readability of the export dump files (`imx<TSP>.sh`, see *CheckExp* option in the [SAPDBA: Reorganization Menu \[Page 615\]](#)).
5. Deletes the tables and indexes (`dro<TSP>.sql`); not with *Create table ... as select*.
6. Merges all the remaining adjoining freespace fragments of the entire database using Oracle functions.
7. Recovery of the tables, if necessary with comments. With *Create table ... as select* the old table is deleted in this step and the new table is renamed with the old name (`tab<TSP>.sql`).
8. Imports the table files, indexes, grants, comments and constraints (`imp<TSP>.sh` or `ldr<TSP>.sh` and if necessary `imp<TSP>.sh`, `ind<TSP>.sql`, `con<TSP>.sql`, `grants.sql`).
9. If necessary, sets back the names of the tables temporarily renamed (under step 1) for data protection (`ren<TSP>.sql`, see *HideTab* option).

Reorganization of Special Tablespaces

When reorganizing the following tablespaces, you might run into problems:

- PSAPPOOLD: This tablespace contains the ATAB table with a very large number of records (around 1 million).

The import - and thus the reorganization - of this tablespace may take up to 10 hours, depending on your operating system. The run time basically depends on the import of the table data. You can influence the duration of the import by performing the reorganization of PSAPPOOLD with a modified import script. You can find detailed information about this in the SAPNet Notes database or in the online SAPDBA help.

Furthermore, if you operate a productive database in ARCHIVELOG mode, some 1-2 GB of free space will be required in the archiving directory of the offline redo log files (<ORACLE_HOME>/saparch). Therefore it can be an advantage to run the database in NOARCHIVELOG mode during the reorganization of PSAPPOOLD.

See [Performance Maximization during Reorganization \[Page 623\]](#). However, after the reorganization has been performed, a backup of the entire database is necessary.

- PSAPSTABD: This tablespace usually contains the most tables with the highest total number of extents, constraints, etc.

The size of the individual Dictionary caches must be large enough to avoid ORACLE errors (for example, errors in recursive calls) during an operation on the tablespace. ORACLE itself manages the individual caches dynamically. To enable this, the init.ora parameter `shared_pool_size` must be defined appropriately large. No bottlenecks should normally occur.

- PSAPBTABD: This tablespace grows very quickly. To avoid frequent reorganization whenever possible, we recommend regular use of the [Command Option -next \[Page 613\]](#).

Also refer to [Performance Maximization during Reorganization \[Page 623\]](#) for information about avoiding problems during the reorganization of these tablespaces. Also use the SAPDBA online help and the SAPNet Notes database for current information about maximizing performance.

Reorganization of a Tablespace with Data Files

Reorganization of a Tablespace with Data Files

SAPDBA actions to reorganize a tablespace with data files (script name):



The data transport method *Create table ... as select* is not available for this type of reorganization.

Preparation Phase

1. Generates the `<timestamp>` subdirectory in the working directory and the log `<timestamp>.rtd`.
2. Defines the number, path and sizes of the new data files of the tablespaces (and possibly of the corresponding index tablespaces). SAPDBA default: if possible a file with the path specification of the first old file of the tablespace which is as large as the sum of all the old data files (see *Tablespace Reorganization with Data Files* in [Reorganization Type \[Page 581\]](#)).
Exceptions:
a) the data does not fit in one file: multiple files are suggested;
b) Raw devices used: SAPDBA suggests the old distribution; you may have to make manual adjustments.
3. Checks whether there will be sufficient space for the files to be reorganized in the file systems of the new data files of the tablespace;
see [Freespace Check During Reorganization of a Tablespace with Data Files \[Page 600\]](#).
4. Generates the SQL and shell scripts and the restart file `restart.rtd` in the `<timestamp>` subdirectory (see [Restart File for Reorganizations \[Page 632\]](#)).
5. Checks whether the extents to be reorganized fit in the data file(s) of the tablespace; see [Freespace Check During Reorganization of a Tablespace \[Page 599\]](#).
6. [SAPDBA: Reorganization Menu \[Page 615\]](#)

Main Phase

1. If necessary, temporarily renames the tables to be reorganized or to be affected by the reorganization (`nam<TSP>.sql`, see *HideTab* option in the [SAPDBA: Reorganization Menu \[Page 615\]](#)).
2. Only when exporting to tape (due to interactive export mode):
Deletes all the index statistics (`alt<TSP>.sql`) as well as any constraints (`drc<TSP>.sql`).
3. Exports all tables and indexes of the tablespace (`exp<TSP>.sh` or `unl<TSP>.sh` and if necessary `exc<TSP>.sh`); see also [Parallel Export/Import \[Page 643\]](#).
4. If necessary, performs a test import to check the readability of the export dump files (`imx<TSP>.sh`, see *CheckExp* option in the [SAPDBA: Reorganization Menu \[Page 615\]](#)).
5. Safety mechanisms (`swi<TSP>.sql`).
6. Deletes the tablespace (`dro<TSP>.sql`).

Reorganization of a Tablespace with Data Files

7. Deletes all files and directories that belong to the tablespace (`del<TSP>.sh`) when each subdirectory only contains one file. If additional, for example non-database, files are also contained in a subdirectory, only the data file is deleted.
8. Creates the new directories and files.
9. Merges all the remaining adjoining freespace fragments of the entire database using Oracle functions.
10. Creates a new tablespace (`tsp<TSP>.sql`).
11. Recovers the tables, if necessary with comments (`tab<TSP>.sql`)
12. Imports the table files, indexes, grants, and constraints (`imp<TSP>.sh` or `ldr<TSP>.sh` and if necessary `imp<TSP>.sh`, `ind<TSP>.sql`, `con<TSP>.sql`, `grants.sql`).
13. If necessary, sets back the names of the tables temporarily renamed (under step 1) for data protection (`ren<TSP>.sql`, see *HideTab* option).

Moving/Renaming the Data Files of a Tablespace

Moving/Renaming the Data Files of a Tablespace

Actions of SAPDBA to move or rename the data files of a tablespace with *Move/rename data files of a tablespace* (script name):

Preparatory phase

1. Generates the `<timestamp>` subdirectory in the working directory and the log `<time stamp>.rmv`.
2. Defines the new data file (possibilities which are supported by SAPDBA: enter the new path, generate a symbolic link, allocate a file, allocate a raw device).
3. Generates the SQL and shell scripts and the restart file `restart.rmv` in the `<timestamp>` subdirectory (see [Restart File for Reorganizations \[Page 632\]](#)).
4. Checks whether there will be sufficient space available for the file(s) of the tablespace in the selected file system(s). Further information of SAPDBA (for example whether or not directories should be deleted).

Main phase

5. For the SYSTEM tablespace or a tablespace with rollback segments:
Closes the database instance with `shutdown immediate`. The instance is then brought to mount status.

For all other tablespaces: sets the tablespace offline (database remains open).
6. Adjusts the file structure at operating system level (possibly creates new directories, moves the files, deletes old directories and symbolic links if there is no more data in the directory, etc.) (`del<TSP>.sh`).
7. Possibly renames the data files, saves the old and new control file (`ren<TSP>.sql`).
8. Opens the database or sets the tablespace online (depending on the action taken in step 5).

Resizing the Data Files of a Tablespace

Purpose

Choose *Reorganization* → *Resize data files of a tablespace* to change the sizes of data files while the R/3 System is running (online). This uses the internal Oracle option `ALTER DATABASE DATAFILE ... RESIZE ...`

This function can be used for all tablespaces, including the SYSTEM tablespace.

Procedure

SAPDBA displays all data files of the selected tablespace, together with their current sizes. It also displays the freespace in the file system (corresponds to the maximum increase in file size allowed) and the minimum file size.

In contrast to the other types of reorganization, there is no split into two phases. No scripts are created; instead the changes in size are made directly in the data file.

Result

The changes in each tablespace are logged in the file `<timestamp>.rrs`. For more information, see [Log Files and Their Contents \[Page 297\]](#).

Procedure for a Reorganization

Procedure for a Reorganization

Preparation Phase (Creation of all Scripts)

All reorganizations (except the options *Resize* and *Move/Rename*) are made by writing the export dump files of the objects into one or more directories (it is also possible to define tape devices). The directories (tapes) must be large enough for the objects to be reorganized. The scripts, however, are always stored in the working directory on the hard disk. The export/import can be performed in parallel by setting the degree of parallelism [exp_imp_degree \[Page 274\]](#) >1 and the number of dump files > 1. See [Parallel Export/Import \[Page 643\]](#).

When a reorganization is started, the system checks whether a user `sapr3` is logged on to the database (e.g. whether an R/3 System is running). SAPDBA outputs a warning.

SAPDBA creates the necessary scripts for the reorganization process.

SAPDBA checks whether there is sufficient storage space in the tablespace (for example, it is possible that the tablespace cannot be created by compressing the extents of an object into one extent because the largest available freespace area is too small).

Before starting the reorganization process, SAPDBA checks whether the reorganized data will fit in the available data file(s).

Features Specific to Tablespace Reorganization with Data Files:

- You can change the default values for file size and file directory. You can thus optimize your disk space utilization through appropriate distribution of the files.
- You can change the number of data files, their respective sizes, and their path. This is subject to the conditions that each of the files is smaller than 2 GB and that at least one data file is created. SAPDBA checks the freespace in all relevant file systems and generates links if new directories are specified that do not conform to SAP naming conventions.

Once all checks have been successfully completed, you can decide if and when you wish to start the actual reorganization (main phase):

- Directly (`Start script(s) immediately`)
- After scheduling it in the background (`Start script(s) in batch`)
 - At a later time (or not) using the restart function (`End (later restart is possible)`)

Main Phase (Processing of Created Scripts)

SAPDBA performs the reorganization (except the option *Resize*) according to the scripts created. If a user `sapr3` is logged onto the database, a reorganization of a tablespace (with or without data files) is terminated when the main phase is started.

After the export phase of a reorganization, the object is deleted from the database. The physical data files are also deleted during a reorganization with data files. Since tables and indexes are removed from the database, please adhere to the [Backup Concepts in Reorganizations \[Page 595\]](#).

In the next phase, the data is imported back into the reorganized object. The reorganization is now complete.

Procedure for a Reorganization

After changing the structure of the database (reorganization of tablespaces with data files) we recommend performing a full database backup; it is essential that you at least back up the relevant tablespace. See [Backup Concept in Reorganizations \[Page 595\]](#).

Features Specific to Background Reorganization

- If you want to run the reorganization process in the background, find out about the necessary configuration of the `at` command (for example, see [Configuring the UNIX Command at \[Page 49\]](#)). The jobs created with `at` are transparent and can therefore be displayed using operating system methods. You will find the necessary environment variables in this kind of file, as well as the command `sapdba -r <complete path/timestamp>`, to start the reorganization.
- Note that tables and indexes which were created *before* the reorganization but *after* creation of the scripts are not taken into consideration during the reorganization, which executes at a later time in the background.
If a tablespace is reorganized without data files, it might terminate due to a lack of storage space. Such objects are lost together with their data in a tablespace reorganization including data files. You should therefore only plan to perform a reorganization at a later time if no new objects were created in between.
- To avoid risks when scheduling several reorganizations one after the other at a later time, note the following:
 - The export dump files for the individual reorganizations must be on different disks.
 - In the case of a reorganization with data files, the old and new files of the tablespaces must be on different disks.
- When scheduling several reorganizations one after the other, suitable time intervals should be chosen so that the individual processes do not overlap. Furthermore, you are not permitted to perform several reorganizations parallel, as this can lead to bottlenecks in the ORACLE Dictionary or an incorrect space check.

The runtime entries in the main log `reorg<DBSID>.log` will assist you in scheduling the reorganization (see [Main Log \[Page 300\]](#)).

See also [Procedures in Detail \[Page 594\]](#).

Procedures in Detail

SAPDBA writes the commands required for the reorganization in scripts contained in the respective work directory. As a result, the operations are transparent for the user (preparation phase). These scripts are processed by SAPDBA in the main phase.

If the reorganization process (main phase) is terminated, you can continue it with the SAPDBA function *Restart* once the error has been eliminated (see [Restarting a Reorganization \[Page 628\]](#)). In some cases (if an error occurred before or during the export), you can reset the reorganization process to its initial state with *Reset* (see [Resetting a Reorganization \[Page 634\]](#)).

When required, SQL scripts can also be started manually with *Additional functions* → *Execute SQL script* (see [Starting Individual SQL Scripts with SAPDBA \[Page 294\]](#)).

Detailed information about the procedures can be found below for the various types of reorganization. The ORACLE programs export/import are used for the export and import. For information on other export and import methods, see [SAP unload/load, SQL*Loader \[Page 665\]](#) and [Reorganization with CREATE TABLE AS SELECT \[Page 636\]](#).

- [Reorganization of a Single Object \[Page 583\]](#)
- [Reorganization of a List of Objects \[Page 584\]](#)
- [Reorganization of a Tablespace \[Page 586\]](#)
- [Reorganization of a Tablespace with Data Files \[Page 588\]](#)

If you only want to change the names and paths of the data files of tablespaces, you can use the option *Move/rename data files of a tablespace*.

For more information, see [Moving/Renaming the Data Files of a Tablespace \[Page 590\]](#).

If you want to change the sizes of the data files of tablespaces, you can use the option *Resize data files of a tablespace*.

For more information, see [Resizing the Data Files of a Tablespace \[Page 591\]](#).

Backup Concepts in Reorganizations

SAPDBA does not perform an automatic backup before or after a reorganization. Depending on the reorganization type, however, SAP highly recommends backing up the relevant data. Please follow the following recommendations:

Reorganization of a Single Object/List of Objects

Before a table/index reorganization, you should always back up the data in the table(s), for example, using the SAPDBA option *Export/import*. In case of error, repeat the reorganization operation, for example, by using the restart functionality. If this attempt also fails, use the backup of the data to restore the old status and repeat the reorganization. Pay special attention to the sequence of the actions that you perform. The actions listed above in case of error should be performed immediately, to guarantee consistency of the database.

Reorganization of a Tablespace (Without Data Files)

Before a tablespace reorganization without data files, you should make a complete backup of the database (online possible; offline recommended). To do this, use the BRBACKUP utility program, which you can start manually or with the SAPDBA option *Backup database*. When the last complete backup of the database was not performed too long ago, and all redo log files up to the point when the reorganization is started are complete and available (on disk, when possible), you do not have to back up the database again. In this case, the database could be restored from the last available backup, should this prove necessary.

After a successful reorganization, you should back up at least the relevant tablespaces, since the reorganization process is very time-intensive.

Reorganization of a Tablespace with Data Files

Before a tablespace reorganization with data files, you should make a complete backup of the database (online possible; offline recommended). To do this, use the BRBACKUP utility, which you can start with the SAPDBA option *Backup database*. When the last complete backup of the database was not performed too long ago, and all redo log files up to the point when the reorganization is started are complete and available (on disk, when possible), you do not have to back up the database again. In this case, the database could be restored from the last available backup, should this prove necessary.

After a tablespace reorganization with data files, you **must** at least back up the relevant tablespaces; any required recovery with SAPDBA is otherwise not possible after such structure changes. In this case, SAPDBA automatically displays the backup menu, to enable you to immediately start the backup of the relevant tablespaces.



A tablespace backup after reorganization is sufficient when you operate the database in ARCHIVELOG mode. If not, the required degree of security can only be achieved by backing up the entire database. If a large number of tablespaces require reorganization, we always recommend a complete offline backup of the database before reorganization of the first tablespace and after reorganization of the last tablespace.

Backup Concepts in Reorganizations**Move/Rename Data Files of a Tablespace**

After moving or renaming the data files of a tablespace, you **must** perform at least a backup of the corresponding tablespace, since otherwise you cannot implement a recovery of the database with SAPDBA after such a change if there should be an error. In this case SAPDBA automatically displays the backup menu, so that you can immediately start the backup of the corresponding tablespace.

Freespace Check

Before a reorganization, SAPDBA checks whether the relevant tablespace contains enough space for the object at hand.

The import check always takes into account:

1. All the extents to be generated according to the number of MINEXTENTS for all the tables to be reorganized (one INITIAL extent and (MINEXTENTS-1) NEXT extents).
2. All other NEXT extents of tables to be expected during the data import (this number is determined from the old total size allocated or from the total size of the tables determined by *Reduce object size: yes*).
3. All expected INITIAL and NEXT extents of all indexes to be reorganized (as in nos. 1 and 2).
4. Reserve space (the size of the largest NEXT extent or at least 120 KB).

The import checks are described in the following sections:

- [Freespace Check during Reorganization of a Single Object \[Page 598\]](#)
- [Freespace Check during Reorganization of a Tablespace \[Page 599\]](#)
- [Freespace Check during Reorganization of a Tablespace with Data Files \[Page 600\]](#)
- [Freespace Check During CREATE TABLE AS SELECT and INDEX REBUILD \[Page 601\]](#)

Freespace Check during Reorganization of a Single Object

Freespace Check during Reorganization of a Single Object

Import check when menu item *Reorganize table or index* is selected.

SAPDBA first checks whether the first extent (in the case of `MINEXTENTS = 1`, otherwise SAPDBA carries out a space check for all requested extents) of the table or index for reorganization fits in one of the data files in the tablespace. Afterwards a check is made to see if there is enough space for all the `NEXT` extents that are expected during the data import.

- SAPDBA determines the size of the freespace areas that will be available in the individual data files when the table or index is deleted.

When the result of the check is negative, SAPDBA does not perform the reorganization. In this case, you should extend the tablespace and/or use the option *Reduce object size: yes* and repeat the reorganization.

- SAPDBA additionally takes reserve space into account when calculating the space required for the reorganized table or index. This reserve prevents the import operation from failing in the rare case that the space requirements of the reorganized table or index are greater than before the reorganization. This can occur, for example, if the `PCTFREE` values are relatively high and the `PCTFREE` areas are quite full. These areas are freed during reorganization. Any additional `NEXT` extents are requested at this time:

The reserve space corresponds to the size of the `NEXT` extent of the affected table or index, but is always at least 120KB.

If the appropriate reserve space is not available, you can still start the reorganization. You can simply ignore the warning message or, alternatively, cancel the reorganization.

Reorganization of Indexes

When an index is reorganized (also as a consequence of a table reorganization), SAPDBA checks whether sufficient space is available for sorting the index in tablespace `PSAPTEMP`.

If tablespace `PSAPTEMP` has less space than twice the size of the largest index for reorganization, SAPDBA issues a warning message.

Freespace Check during Reorganization of a Tablespace

Import check when menu item *Reorganize tablespace* is selected.

SAPDBA checks whether enough space is available in the data file(s) of the tablespace for the extents of the tables/indexes which are to be reorganized (the check includes the entire storage of the tablespace).

SAPDBA performs this check in the following manner:

1. Sorts the tables/indexes for reorganization in ascending INITIAL value order. This sequence is stored in the SQL script containing the CREATE TABLE/INDEX commands.
2. Sorts the data files of the tablespace by size of the freespace that will be available after the objects are deleted, in descending order (at the beginning of the check, this value is identical to the size of the file; this is then reduced by the size of the mounted objects).
3. Attempts to fit the extents of the tables/indexes (in the given order) in a file. This process starts at the file with the most freespace. If a file with sufficient freespace is found, the table/index is mounted into it. This operation (e.g. step 2 and step 3) is repeated until space has been found in the tablespace for all the objects contained in the SQL script.

SAPDBA assumes that the extents for reorganization can be successfully reimported when the following condition has been met:

- The freespace available according to the simulated reorganization (estimated value) is greater than the reserve space.

The reserve space is a freespace requirement that SAPDBA uses in its calculations. This reserve ensures that the import of a reorganized tablespace only fails in the rarest of cases. The reserve corresponds to the size of the largest NEXT extent of the object, but is always at least 120KB.

If you ignore a warning message during reorganization and the import fails, you can extend the tablespace by adding a data file. Then select the SAPDBA restart function to continue with the reorganization.

- If SAPDBA reports at the very start that there is not enough space for the requested reorganization, we recommend extending the tablespace and/or using the option *Reduce object size: yes* and then restarting the reorganization. If the SAPDBA check fails, the tablespace probably would have had to be extended soon anyway.

Freespace Check during Reorganization of a Tablespace with Data Files

Freespace Check during Reorganization of a Tablespace with Data Files

Import check when menu item *Reorganize tablespace and data files* is selected.

SAPDBA displays the size of the data file(s) and asks whether you want to change the file size(s).

During the reorganization of a tablespace with data files, SAPDBA performs - in addition to the import checks (see [Freespace Check during Reorganization of a Tablespace \[Page 599\]](#)) - a file system check. SAPDBA checks whether the selected file system has sufficient freespace for the new files of the tablespace (the first suggestion from SAPDBA is the file system containing file `<tablespace name>.data1` of the tablespace).

The check considers the freespace that will be available once the tablespace is deleted.

SAPDBA specifies which directories the new data files for the tablespace will be stored in (file numbering starts again with the suffix "1") and asks whether you want to specify additional paths for the data files. If necessary, you can then specify a path in a file system with more space. Both the available `sapdata<n>` directories and the possible raw devices can be selected with a list which can be displayed by SAPDBA. In this case, SAPDBA repeats the import check and then performs the reorganization.

If the files are not in a standard SAP directory, SAPDBA proposes the directory `<ORACLE_HOME>/sapdata<n>/<tablespace name>_<file number>`, which is a symbolic link to the specified directory.

Please adhere to the SAP naming conventions whenever possible (see [Naming Convention \[Page 691\]](#)).

If a raw device should be used for the new data file, SAPDBA automatically generates a symbolic link to the specified raw device in directory `sapraw`. In most cases SAPDBA suggests the old distribution of the tablespace on the files (since the raw devices have a fixed size). You might have to take manual action to change this suggestion.

Freespace Check During CREATE TABLE AS SELECT, in Deferred Mode and During INDEX REBUILD

When reorganizing with CREATE TABLE AS SELECT or in *Deferred* mode the objects are reorganized one after the other. During reorganization sufficient memory space must be available both for the old and the new object. Each time SAPBDA checks whether there is enough space in the relevant tablespace in order to regenerate the largest object to be reorganized. The largest object is always reorganized first.

In exceptional cases, for example when the objects to be reorganized are very fragmented or free space is relatively short, errors can arise during the reorganization of a list of objects or a whole tablespace when generating the new objects. Such ORACLE errors, which SAPDBA registers as a warning, tell you that the object that caused the termination will be left out. The object is unchanged, however. You can reorganize it later as a single object or carry out a restart (if necessary after a tablespace enhancement).

See also:

[Reorganization with CREATE TABLE AS SELECT \[Page 636\]](#)

[SAPDBA: Reorganization Menu \[Page 615\]](#): Option: *Rebuild*.

SAPDBA: Reorganization

SAPDBA: Reorganization

	Reorganization
<i>a</i>	- Check extents and fragmentation
<i>b</i>	- Alter/Show table or index storage parameters
<i>c</i>	- Reorganize single table or index
<i>d</i>	- Reorganize list of tables or indexes
<i>e</i>	- Reorganize tablespace
<i>f</i>	- Reorganize tablespace and data files
<i>g</i>	- Move/rename data files of a tablespace
<i>r</i>	- Restart
<i>q</i>	- Return

For more information, see [Extent Check Functions \[Page 606\]](#), [Changing MAXEXTENTS and NEXT \[Page 612\]](#), [Carrying Out a Reorganization \[Page 614\]](#).

Reorganization

A reorganization is an extremely complicated process. We therefore recommend that you follow the following procedure:

1. Considerations [Before a Reorganization \[Page 604\]](#).
2. [Checking Statistics \[Page 605\]](#), to find out if reorganization is necessary.
3. [Changing MAXEXTENTS and NEXT \[Page 612\]](#), to prevent reorganization.
4. [Carrying out a Reorganization \[Page 614\]](#).

[Extent Check Functions \[Page 606\]](#)

[Number of Data Files \[Page 610\]](#)

[Analyzing the Statistics \[Page 611\]](#)

[Reorganization of Special Tablespaces \[Page 587\]](#)

[Command Option -analyze \[Page 608\]](#)

[Command Option -next \[Page 613\]](#)

[SAPDBA: Reorganization Menu \[Page 615\]](#)

[Freespace Check \[Page 597\]](#)

[Changing the Extent Size \[Page 620\]](#)

[Moving a Table or Index \[Page 622\]](#)

Before a Reorganization

Before a Reorganization

- **Shut down the R/3 System:** To reorganize the database, you must shut down the R/3 System. The database should be run in RESTRICTED-SESSION mode. See [Restricted Session \[Page 63\]](#).
- **Reorganize after starting the database:** We recommend carrying out reorganizations immediately after opening the database in RESTRICTED-SESSION mode, because no locks can then exist for database objects.
- **Time scheduling:** If you are reorganizing large quantities of data, the import/export and creation of indexes can be very time-intensive.

If you are reorganizing several objects specified in a list or tablespace, you should make sure that your file systems are available in sufficient quantity and of a sufficient size. Take advantage of the opportunity to run the export/import in parallel (if your host has several CPUs), which implies a great saving in time. See [Parallel Export/Import \[Page 643\]](#).

The reorganization of tablespace PSAPPOOLD may take up to 10 hours. You can speed up this process, however, if you use the possibilities for [Performance Maximization during Reorganization \[Page 623\]](#).

Also check the main log `reorg<DBSID>.log` to see whether a comparable operation has already run. This will enable you to better estimate the time requirements.

- **Bundle your reorganizations:** When possible, plan your reorganizations well in advance, to enable you to perform them together. This will keep disturbances in ongoing operations to a minimum.
- **Database backups:** Follow the [Backup Concepts in Reorganizations \[Page 595\]](#).
- **Database analysis:** [Checking Statistics \[Page 605\]](#). If the statistics check has revealed a large amount of free space in the objects, you can perform the reorganization in such a way that the storage parameters are adapted (see [Options for Changing and Checking the Storage Parameters \[Page 671\]](#)).
- **Freespace:** If you want to reorganize a table or tablespace without data files, make sure that enough free storage space is available in the relevant tablespace before the reorganization (around 5-10% of the tablespace size).
- **Naming conventions for files:** SAPDBA can only reorganize tablespaces whose data files conform to the SAP naming conventions for files (see [Naming Convention \[Page 691\]](#)). Note the special features when working with raw devices (see [SAPDBA Support for Raw Device Configuration \[Page 767\]](#)).

Checking Statistics

You can use the following tools to inspect the statistics relevant for a reorganization:

- Performance Monitor of the R/3 System (see online help of the *Computing Center Management System*)
- Command line option `sapdba -check`
See [SAPDBA Command Mode \[Page 69\]](#).
- Command line option `sapdba -analyze`
See [Command Option -analyze \[Page 608\]](#).
- SAPDBA option *Reorganization* → *Check extents and fragmentation*
See [Extent Check Functions \[Page 606\]](#).
- SAPDBA option *Tablespace administration*
See [Number of Data Files \[Page 610\]](#).

Please also read the notes in [Reorganization of Special Tablespaces \[Page 587\]](#).

Extent Check Functions

Extent Check Functions

With its menu item **Check extents and fragmentation** in the Reorganization menu, SAPDBA provides the following check functions:

- **Objects with at least n extents** displays objects that may need to be reorganized due to their large number of extents.

This function checks all objects in the database.

You can restrict the search by specifying a list of tablespaces, tables, indexes or other segment types to check.

- Lists must have the following format: Name,Name,... .
- Use the ORACLE placeholder “%” for generic specifications (corresponds to the ORACLE LIKE function). All objects with this LIKE pattern are checked.

Special features when selecting the tablespaces: You would generate a list of tablespaces (PSAPBTABD and PSAPBTABI) with PSAPBTAB%. You could select a tablespace from this list. If you want to check both the objects of PSAPBTABD and the objects of PSAPBTABI, you can do so by entering PSAPBTAB%.

- Entering “all” means that you select all objects.
- **Tablespace fragmentation (summary)** displays information on the size and fragmentation level of the specified tablespace.

The following information is displayed:

- Number of objects (tables and indexes) in the tablespace
- Number of allocated extents
- Average number of allocated extents per object
- Number of data files in the tablespaces
- Number of free fragments
- Number of freespace areas
- Average number of free fragments per data file

- Allocated space (total of all allocated extents)
- Freespace (total of all free fragments)
- Largest freespace area of the tablespace (adjacent free fragments)

- Available storage space in the tablespace
- Space occupied by ORACLE and operating system overhead
- Total size of the tablespace
- **Tablespace fragmentation (all extents)** displays detailed information about all segments (tables, indexes, ...) and their extents of the selected tablespace. The extents appear in the table in accordance with their physical sequence within the data files. If you

enter an object name or 'FREE', it is also possible to display the physical position and fragmentation of the object or only of the freespace.

You can use the first check function listed above to search the database, the second to display detailed information about the tablespaces that have a large number of extents, and the third to display an overview of the distribution of the files in the individual extents (e.g. the fragmentation).

- ***Estimate/Compute table***
Validate index
Estimate/Compute tablespace

These options enable you to contrast the allocated and actually occupied space, based on the ORACLE function ANALYZE TABLE/INDEX <name> ESTIMATE/COMPUTE STATISTICS or ANALYZE INDEX <name> VALIDATE STRUCTURE. The runtimes of an analysis (especially when COMPUTE is used for all tables/indexes of a tablespace) can be quite long. You can also limit the time with the menu (see: [Command Option - analyze \[Page 608\]](#)).

You can analyze an individual table (*Estimate/Compute table*), an index (*Validate index*) or all objects (tables and indexes) of a tablespace (*Estimate/Compute tablespace*).

You can generate current statistics each time or go back to statistics that already exist (*Get last analysis*).

The values can be used as a starting point for reducing storage sizes of tables/indexes and/or tablespaces during a reorganization. Please note, however, that a safety margin of 10-20% is always recommended. This safety margin is automatically taken into consideration when you reorganize with the *Reduce object size: yes* option.



The function *Estimate/Compute table/tablespace* locks only the table record just examined.

The function *Validate index* locks the entire table belonging to the index.

For information on logging all check functions, please refer to [Overview of the Logs \[Page 297\]](#) and [Setting the Log Checks Option \[Page 50\]](#).

Command Option -analyze

Command Option -analyze

The command option `sapdba -analyze <tablespace(s) | table>` involves an analysis of the allocated and actually used space for the specified table or tables and/or indexes in the specified tablespaces. (Normally, if you have kept to the SAP naming convention, there should be no indexes in a data tablespace and no tables in an index tablespace. In this case, statistics are only generated for the one object type. If there are tablespaces, which contain various object types, statistics are generated both for the tables and the indexes per tablespace).

Based on an estimate, the command:

- Specifies the estimated freespace per specified tablespace.

Tablespace structure: `tablespace_name, total_space, free, allocated, used`

- Creates a hit list of the 20 tables and/or indexes with the most unnecessarily allocated space.

Table structure: `tablespace_name, table_name, empty, never_been_used, used`

Index structure: `tablespace_name, index_name, total, used_by_btree, used`

When you call `sapdba -analyze` (without defining a method/option), the ORACLE command ESTIMATE is applied to the relevant tables. You can also perform analyses using COMPUTE or VALIDATE STRUCTURE (option `sapdba -analyze <TSP> -method <options>`).

Because this analysis is quite time-intensive (depending on the number of tables or indexes involved), it should be planned in the background and, if possible, only for selected tablespaces or tables. An analysis of all tablespaces for an SAP database with 5 GB of data can take up to 5 hours, for example, depending on the hardware used.

You can limit the number of records to be analyzed by using the option `sapdba -analyze <TSP> -option R<n> | P<n(*1000)>`:

`P<n>`: n percent of all the existing records are analyzed,

`R<n(*1000)>`: n*1000 records are analyzed.

You can decide whether you want to create current statistics with SAPDBA or use existing statistics which are not older than a certain number of days. Use the option `sapdba -analyze <TSP> -time <number of days>`.

To limit long analysis operations to a certain number of hours, use the option

`sapdba -analyze <TSP> -T <number of hours>`.

The values determined by the analysis can be used as the basis for manually reducing the storage sizes of tables/indexes or tablespaces. You can reduce their space requirements during the reorganization, for example by calling the reorganization with menu option *Change storage manually yes* to possibly reduce the value for INITIAL in accordance with the analysis. However, we always recommend a safety margin of 10-20%.

Normally you will not have to perform this complex, manual procedure. If you choose the *Reduce object size: yes* option during the reorganization, the analysis will be performed directly during the reorganization and the space requirements will be reduced automatically. SAPDBA takes the safety margin described above into consideration.

See [Reducing the Size of the Table or Index \[Page 676\]](#).

Command Option -analyze

When tables with a large data volume are involved, the temporary tablespace PSAPTEMP may overflow, since the analysis with `sapdba -analyze` involves a large number of sort operations. No statistic values are available for this table and the storage space which was assigned is not reduced.

The results of the analysis are stored in the log `sapcheck/<timestamp>.aly`.



The command option `-analyze` or a reorganization with the *Reduce object size: yes* option refreshes the database statistics of the particular object and thus has an effect on the behavior of the database optimizer (see [Support of the Cost-based Optimizer \[Page 91\]](#)). Also note that the statistics of all the reorganized objects are deleted during reorganization. You can recreate them with the command option `-analyze` or with the menu option *Check extents and fragmentation*.

See also: [SAPDBA Command Mode \[Page 69\]](#).

Number of Data Files

Number of Data Files

You can display the number of all data files in the database with the SAPDBA function

Tablespace administration → *Freespace and fragmentation of all tablespaces*.

See also [Freespace Statistics \[Page 683\]](#).

When you use the function *Tablespace administration* → *Alter tablespace Add Datafile* →

Display current files, the number of data files per tablespace is displayed.

See [Extending a Tablespace \[Page 686\]](#).

Also use the displays *Tablespace administration* → *Display all tablespaces and datafiles* and

Tablespace administration → *Display all character devices in /dsk* to obtain an overview of the file distribution. These lists also show which files are on raw devices.

This information can also be displayed in the Performance Monitor. See online help for the *Computing Center Management System*.

Analyzing the Statistics

Reorganization is not required as often as is commonly assumed. This can be briefly explained with the following considerations.

- **Number of data files:** A reorganization with data files is not necessary until the total number of data files in the database approaches the limit MAXDATAFILES. In such cases, reorganize tablespaces that consist of many files using the SAPDBA option *Reorganization* → *Reorganize tablespace and data files*.
- **Number of extents per table or index:**
SAP usually uses indexes to find entries. This means that full table scans are only rarely required. It is not necessary to limit the number of extents per table as long as ORACLE is capable of managing that number.

A reorganization should be considered when total system performance declines due to a large number of extents of a table or an index. If many of the objects consist of a large number of extents, reorganize the relevant table, index, or the entire tablespace.

- **Number of freespace fragments in the tablespace:**
The system monitor of the ORACLE database system is capable of automatically merging the **contiguous** freespace areas. This significantly reduces the probability that a reorganization will become necessary due to freespace problems.

If, however, none of the remaining fragments is large enough to create a new extent, you should extend the tablespace and possibly schedule a tablespace reorganization at a later time that is suitable.

Also see [Limitations of the Database System \[Page 579\]](#).

Changing MAXEXTENTS and NEXT

Changing MAXEXTENTS and NEXT

As described in [Limitations of the Database System \[Page 579\]](#), a table or an index has a software limit of 100 extents (parameter MAXEXTENTS) in the initial R/3 installation. If a table or an index reaches this limit, an extent overflow error occurs the next time an extent is requested.

In this case, you can increase the NEXT and/or MAXEXTENTS parameters, e.g. using the function *Reorganization* → *Alter/Show table or index storage parameters*. MAXEXTENTS can be increased to a value lower than or equal to 505 extents (for a block size of 8 KB). Therefore, when you reach the soft limit for MAXEXTENTS, you should not perform a reorganization of the object or tablespace immediately. However, this procedure should only be used in an emergency. The table or index should be reorganized as soon as possible in order to reset the MAXEXTENTS size to the SAP standard value.

MAXEXTENTS is automatically reset to the SAPDBA default value when there is an import or a reorganization of the relevant object. This value is defaulted even when you use one of the options for changing storage parameters manually.



As of ORACLE 7.3, you can set MAXEXTENTS to 'UNLIMITED'.

See also [Changing Storage Parameters \[Page 668\]](#).

Measures to Avoid Reaching the MAXEXTENTS Value:

- You can prevent the number of extents from growing too quickly by promptly increasing the NEXT parameter (larger extents are then allocated). You should therefore regularly use the SAPDBA command line option `sapdba -next`. See [Command Option -next \[Page 613\]](#).
If, despite the above considerations, a table still reaches a large number of extents, you can increase the MAXEXTENTS parameter (as described above) to allow more extents.
- Experienced database administrators can also influence the degree of occupancy of the data blocks by changing the parameters PCTFREE and PCTUSED. However, this is not normally necessary.

Command Option -next

If you are reorganizing a tablespace that grows quickly (e.g. tablespace with transaction data), we recommend using the SAPDBA command option `sapdba -next <tablespace/s>`. This option instructs SAPDBA to adjust the value of NEXT in accordance with the rapid growth of the tablespace. Adjusting NEXT prevents the tablespace from reaching the MAXEXTENTS limit too quickly.

The `-next` option involves adjusting the NEXT parameter according to the tables DD09L and TGORA/IGORA.

The new value for NEXT is calculated as the maximum of the following 2 values:

1. An SAP object is assigned a size category via table DD09L. The NEXT value is determined from table TGORA/IGORA based on that category. (If no category is assigned in DD09L, this value is not generated.)
2. SAPDBA calculates the maximum of the following values:
 - the current NEXT value.
 - 10% of the total allocated space

SAPDBA compares this chosen value (maximum of 1. and 2.) with the values for the extent size in TGORA/IGORA. SAPDBA chooses the next smaller value found in TGORA/IGORA as the new NEXT value. If TGORA/IGORA do not exist, SAPDBA uses an internal default value. See also [Default Value for NEXT \[Page 673\]](#).

This procedure prevents the number of extents from growing too quickly and - with repeated use - simulates an asynchronous, linear PCTINCREASE response. The changed tables are recorded along with their old and new values in the log `<timestamp>.nxt`.

See [SAPDBA Command Mode \[Page 69\]](#).

Carrying out a Reorganization

Carrying out a Reorganization

How to perform a reorganization with SAPDBA:

1. Pay attention to the [Backup Concepts in Reorganizations \[Page 595\]](#).
2. **Shut down the R/3 System.** Shut down the database (see [Shutting down the Database System \[Page 101\]](#)).
3. Note the following special feature:

If you want to use *Reorganize tablespace and data files* (reorganization of tablespaces with data files), you should make sure that no ORACLE background processes exist. Hanging background processes can disturb the course of a tablespace reorganization because they keep database files open. While these data files can be deleted during the reorganization, the disk space from such a file is not released by the operating system until the process is no longer keeping it open.
4. When necessary, change the initialization profile `init<DBSID>.ora` to accelerate the reorganization. Also read other remarks about performance maximizing.

See [Performance Maximization during Reorganization \[Page 623\]](#).
5. Start the database in RESTRICTED-SESSION mode (see [Starting up the Database System \[Page 100\]](#)).
6. Carry out the reorganizations using the options in the *Reorganization* menu. See [SAPDBA: Reorganization \[Page 602\]](#) and [SAPDBA: Reorganization Menu \[Page 615\]](#).
 - **Storage checks:** Before a reorganization, SAPDBA checks whether the object(s) for reorganization can be reimported - that is, whether enough space is available.

See [Freespace Check \[Page 597\]](#).
 - **Storage parameters:** you can define the storage parameters during a reorganization.

See [Options for Changing and Checking the Storage Parameters \[Page 671\]](#).
7. Shut down the database.
8. If necessary, reset the profile parameters of the database to their original values.
9. Pay attention to the [Backup Concepts in Reorganizations \[Page 595\]](#).

For complicated reorganizations, we recommend performing an offline backup of the complete database after the reorganization. **A backup of the database after the reorganization should be available** to enable you to quickly restore the database in case problems occur at a later time.
10. Start up the database in normal mode.

SAPDBA: Reorganization Menu

Select the required reorganization type (*Reorganize single table or index, Reorganize list of tables or indexes, Reorganize tablespace, Reorganize tablespace and data files or Move/rename data files of a tablespace*). Note the differences in the menu options depending on the selected type of reorganization.

The following menu is displayed:

a	- Tablespace		
d	- Working directory		
e	- Dump destination		
f	- Export/import	g	- Storage param.
	Compress		ComprExt
	Chop		SAP-NEXT
	CheckExp		CheckExt
	Commit		ReduceOb
	Parallel		Manually
	- Start		DropTab (only for single table)
s	- Return		

If you choose *Move/rename data files of a tablespace*, only menu options a, d, s and q appear.

If you want to use the SAP tool unload/load instead of Oracle export for exporting, you can set this with the menu option *-f*.

For more information, see [SAPDBA: Expert Menu \[Page 644\]](#), [SAPDBA unload/load, SQL*Loader \[Page 665\]](#), [Reorganization with Create Table...as Select \[Page 636\]](#).

The most important menu items are listed below:

- **Object names:** You can reorganize individual tables, indexes, a list of objects or all the objects of a specified tablespace (depending on the selected reorganization type).

Enter the name of the object/tablespace to be reorganized and if necessary, its owner or the name of an object list. You can generate lists of objects (tables, indexes, tablespaces) from which you can select the required object by entering *like* values (for example T%, T%_0, PSAP%).

If you want to edit a group of several objects, an ASCII file must have been created in the working directory with the required object list. There may only be one object name on each line. If SAPDBA finds several definitions in a line, the first value will be interpreted as the owner of the second object specified. All other entries in this line are ignored (valid separators between the owner and object: spaces or dots). If there is only one entry, it will be interpreted as an object name. The owner of this object is automatically **sapr3**.

- **Target tablespace (Current tablespace):** Default: The tablespace name in which the object currently resides is displayed.

SAPDBA: Reorganization Menu

If a different target tablespace is required, you can specify it here. This option only appears for the reorganization type *Reorganize single table or index*.

- **Working directory:** Default: <ORACLE_HOME>/sapreorg (init<DBSID>.dba parameter [exireo_dumpdir \[Page 271\]](#))

Working directory in which the scripts and logs are to be saved. You can change this working directory if required.

- **Export dump destination:** Default: <ORACLE_HOME>/sapreorg (init<DBSID>.dba parameter [exireo_dumpdir \[Page 271\]](#))

The target directories or tape devices for the export dump files; the space available in the directories or on the tapes must be at least as large as the objects to be reorganized (table(s), index(es), tablespace). All further scripts and logs are always written to the working directory.

The export or import can be run in parallel by defining the parameter [exp_imp_degree \[Page 274\]](#) > 1 (See [Parallel Export/Import \[Page 643\]](#)).

SAPDBA warns you if there is not enough space available. You might want to continue with the reorganization nevertheless. The data written to the directories on the disk can be compressed by a certain factor by using the option *Export/import method* → *Compress dump file(s): yes* (if ORACLE Export/Import is used). For security reasons, the space check does not take the affects of compression into account (the compression rate and degree of filling is not known exactly).

The menu option *Reduce object size: yes* is used, the export dump is computed more exactly or exactly (choose: estimate or compute statistics) since SAPDBA determines the storage space actually occupied by the object data and carries out the space checks using this value. The compression rate is, however, also not taken into account here.

When you export to tape, SAPDBA asks you to state the size of the tape. SAPDBA also checks in this case whether the data to be reorganized has space on the tape(s). SAPDBA can only use one tape drive for each export dump file. You are warned if the total amount of space required for a dump file is greater than the tape size.



If SAPDBA uses the Oracle program export for the export to tape, write errors are possible which Oracle export will not recognize. It is, therefore, essential that you back up the corresponding data before starting such a reorganization; otherwise you could lose your data during the reorganization. In addition, always use the SAPDBA option *CheckExp: yes*. In general, an export to disk will be less susceptible to errors than one to tape.

- **ORACLE exp/imp, Unload/load, Create table as select:** Default: **ORACLE export/import**

Here you can decide which methods you want to use for the data transport and set some parameters if required.

- *ORACLE export / ORACLE import* (default)
*SAP unload / ORACLE SQL*Loader* (as of ORACLE 7.2)
SAP unload / SAP load

SAPDBA: Reorganization Menu

Create table... as select (not for tables with *long* columns, or for reorganizations of tablespaces with data files).

For more information, see [SAP unload/ load, SQL*Loader \[Page 665\]](#).

- **Buffer size:** Default: **3000000**

To accelerate the reorganization, we recommend providing at least 3 MB of buffer space for the ORACLE Export and Import programs or for the load programs.
- **Compress (Compress dump file):** Default: **no**

Select **yes** if the data is to be compressed during export. In this case, SAPDBA sends the data with the UNIX command **compress** before writing it to the directories for the export dump files. By compressing the data, you can reduce the disk space required for reorganization by a certain amount (depending on the compression ratio).

Compression is only possible for an export to hard disk (with ORACLE Export). This option will be ignored if you export to tape. Therefore, use tape stations with hardware compression where possible.

Do not use compression for tablespaces containing objects already compressed by the SAP database interface (see Tablespaces with Compressed Objects), since there is no advantage to be gained (there are more likely to be disadvantages).
- **Chop (Chop dump file):** Default: **no**

This menu option only appears if the parameter [chop_util_name \[Page 269\]](#) is entered in the profile `init<DBSID>.dba`.

Select **yes** if the export dump files are larger than the maximum file size (usually 2 GB) for the particular operating system. In this case, SAPDBA sends the exported data to the chop tool via a named pipe. The chop tool splits the export dump files into several smaller files.

If both *Compress dump files* = **yes** and *Chop dump files* = **yes** are selected, compression takes place within R3chop (option `-c`). The chop option does not exist for SAP load/unload or for Windows NT.
- **CheckExp (Check dump files):** Default: **yes**

After the ORACLE export, SAPDBA performs a read check if you select *CheckExp: yes* (test import with the Oracle parameter *Indexfile*). The `inx<TSP>.sql` scripts are created. This option should be used in particular if you are exporting to tape.
- **Commit (Commit during import/load):** Default: **yes**

If you use the default setting **yes**, the COMMIT command is passed to the database once the data in the buffer has been imported.
- **Parallel (Parallel export and import):** Default: **degree 1**

The default value corresponds to the value of the parameter [exp_imp_degree \[Page 274\]](#) in the profile `init<DBSID>.dba`. The export/import can be performed in parallel by selecting a degree of parallelism >1 (see [Parallel Export/Import \[Page 643\]](#)).
- **ParTable (Parallel table creation):** Default: **degree 1**

Exists only when using the method *Create table... as select*.

SAPDBA: Reorganization Menu

By choosing a degree of parallelism >1 you can perform the creation of the tables in parallel. This option is identical to the option **ParIndex** in *Object handling*. (See Performing [Processes in Parallel \[Page 627\]](#)).

For more information, see [SAPDBA: Expert Menu \[Page 644\]](#).

- **Storage parameters:**

You can make various settings for the storage parameters here:

- **ComprExt (Compress extents):** Default: **yes**
- **SAP-NEXT (Use ABAP/4-DIC for NEXT):** Default: **yes**
- **CheckExt (Check space):** Default: **for all extents**
- **ReduceOb (Reduce object size):** Default: **no**
- **Manually (Change storage manually):** Default: **no**

If you reorganize tables, indexes or a tablespace, you can change the default SAPDBA storage parameters for the individual objects (see [Options for Changing and Checking the Storage Parameters \[Page 671\]](#) and [Changing the Extent Size \[Page 620\]](#)).

- **Object handling:**

- **SAP-Tsp (Use ABAP/4-Dic for tsp.):** Default: **no**

You can move an object to another tablespace (*Target tablespace*) during a reorganization. This option, however, can only be used if a single table or a single index is reorganized. You can use the option *Use ABAP-Dic for tablespaces: yes* to place the objects in the tablespace defined in the ABAP Dictionary (see [Moving a Table or Index \[Page 622\]](#)).

- **RebuildI (Alter index rebuild):** Default: **no**

During all pure index reorganizations, that is when an index is reorganized without its table or when not all the tablespace is reorganized with data files, the command *Alter Index Rebuild* can be carried out instead of deleting and setting up again with *Drop/Create Index*. The advantage is that the old index remains accessible, whilst the new index is set up in a temporary segment. The old index is only deleted when the new index has been completely set up. This accelerates the new setting up of the index. It is also unnecessary to delete any constraints which cause the index to be accessed when they are being checked. Note that, temporarily, there has to be enough space in the relevant tablespace for the old and the new index at the same time. In addition to this, the tablespace's free space cannot be combined optimally.



Indexes whose table is also being reorganized or whose tablespace, including data files, is being reorganized, (that is, those who are deleted together with the table by the command *Drop Table*, or deleted together with the tablespace by the command *Drop Tablespace... Including Contents*), are always recovered with *Create Index*. The option *RebuildI* is ignored in this case.

- **ReduceFi (Reduce data file size):** Default: **no**

If a tablespace with data files is reorganized, you can reduce the size of the data files using this option. SAPDBA determines the space actually required for the tablespace

SAPDBA: Reorganization Menu

objects after the reorganization. SAPDBA suggests a size per file which is computed from this value and a 10 percent safety margin. You can increase this value if you expect the tablespaces to grow considerably.

- **ParIndex (Parallel index creation):** Default: degree 1
You can create indexes in parallel (see [Performing Processes in Parallel \[Page 627\]](#)). This is advisable if you are using more than one CPU (see also *ParTable*).
- **DropTab (DropTable):** Default: defer
This option applies only to the reorganization of a single table (*Single Table HideTab (Hide tables during reorg.)*: Default: no
If you set this value to 'yes', all the tables which are reorganized or which are indirectly affected by the reorganization are temporarily renamed during the reorganization (that is, if their indexes are reorganized or referenced by foreign key from other tables to be reorganized).
- **Reorganization** The table is not deleted directly after the data export that precedes the creation of the new table, it is merely renamed. The renamed table is deleted after the data has been reimported. This procedure improves data security, however it does mean that there has to be space for the old and the new table in the tablespace at the same time. If SAPDBA realizes that there is not enough space in the tablespace, it switches automatically from *Deferred* mode to normal mode while it is creating the scripts. When a table is transported into another tablespace there are no space problems, despite the fact that the old and new tables exist at the same time. In this case, the *Deferred* mode cannot be changed.

Changing the Extent Size

Changing the Extent Size

With all reorganization types, you can change the storage parameters (i.e. influence the extent parameters, among others) with the menu items *Compress extents*, *Use ABAP/4 DIC for NEXT*, *Reduce object size* and *Change storage manually*.

INITIAL

It makes sense to change the size of the INITIAL extent in the following cases:

- **Larger extent:** If you expect a table or index to grow quickly, you can specify a larger value for INITIAL.

The additional space in the extent enables the database system to add data without having to immediately create a new extent. When possible, define the extent large enough to cover the requirements of the table or index for some time.

Remember that the extent will have to fit in the freespace expected after a reorganization. SAPDBA undertakes this check before the start of the reorganization.

Useful information for determining the extent size:

- Information about growth trends in the SAP Performance Monitor
See online help for the *Computing Center Management System*.
- Freespace information in SAPDBA (*Tablespace administration* → *Freespace and fragmentation of all tablespaces*, see [Freespace Statistics \[Page 683\]](#)).
- **Smaller extent:** When the following conditions apply, you can specify a smaller value for INITIAL:
 - The option *Reduce object size: yes* was not used.
 - You expect a table or index to shrink because data has been deleted.
 - Only little new data is expected.

SAPDBA always performs the space check both for the first extents according to the value of MINEXTENTS, and for all other NEXT extents expected during the data import (from the total size of the object).

MINEXTENTS

The minimum number of extents per object should normally be 1. SAPDBA however divides the given INITIAL extent (when *Compress extents: yes*) into several parts, if INITIAL would be bigger than the biggest data file or the largest available freespace fragment. In this case, SAPDBA also sets the MINEXTENTS to the corresponding number.

MAXEXTENTS

The maximum number of extents per object should be 100 (with a block size of 8 KB).

NEXT

The size of the NEXT parameter depends on the expected growth of the individual objects. Also consider the selected INITIAL value. If a tablespace contains many fast-growing objects, you should set NEXT large enough to cover storage requirements in the long term and avoid creating new extents whenever possible.

Changing the Extent Size

Also learn about the SAPDBA command options `-analyze` and `-next`, which can help you decide if and when to change the extent sizes, and even change the NEXT value automatically, when necessary.

See [Command Option -analyze \[Page 608\]](#) and [Command Option -next \[Page 613\]](#).

For more information on the other storage parameters, please refer to [Changing Storage Parameters \[Page 668\]](#).

Moving a Table or Index

Moving a Table or Index

Reorganize single table or index

By specifying a tablespace other than the *target tablespace*, you can move an individual object to a different tablespace during a reorganization.

In addition to the familiar objects and checks from the reorganization of a single object, SAPDBA also performs the following operations:

- Checks whether the specified tablespace corresponds to the SAP naming convention. If not, the entry is rejected.
- The specified tablespace cannot be one of the ORACLE tablespaces SYSTEM, PSAPROLL or PSAPTEMP (see [SAP Naming Convention: Tablespace Names \[Page 41\]](#)).
- When you move a table, the corresponding indexes are automatically moved to the corresponding tablespace. If the corresponding tablespace does not exist or is too small, the entry is rejected.

Example: when a table that is part of tablespace PSAPBSPD is reorganized, the corresponding indexes are stored in tablespace PSAPBSPI.

- If an index to be reorganized does not lie in the corresponding tablespace, the corresponding tablespace is the automatic default value for the target tablespace.

Example: the table of the index to be reorganized lies in tablespace PSAPBSPD. As a result, tablespace PSAPBSPI is suggested as the default target tablespace for the index.

- SAPDBA updates the data type of the table in SAP table DD09L.

Other Types of Reorganization

Using the option *Use ABAP/4-Dic for tablespaces: yes* you can create the objects to be reorganized in the tablespace which was defined for the objects by specifying a data type. See [Technical Configuration \[Page 25\]](#).

SAPDBA also performs the following operations in addition to the known actions and checks for reorganizing a single object:

- Check if there is an entry in table DD09L for the objects.
 - There is no entry: The objects are created in the original tablespace.
 - There is an entry: The objects are created in tablespaces in accordance with the information in tables DD09L and TAORA/IAORA.
- If you move a table, the corresponding indexes are automatically moved to the corresponding tablespace. If the corresponding tablespace does not exist or is too small, the objects are created in the original tablespaces.

Performance Maximization during Reorganization

[Changing init.ora Parameters \[Page 624\]](#)

[Archiving Mode during the Reorganization \[Page 625\]](#)

[Performing Processes in Parallel \[Page 627\]](#)

You will also find further information about this in the online help of SAPDBA and the SAPNet Notes database. See also [Reorganization of Special Tablespaces \[Page 587\]](#).

Speeding Up a Reorganization

Speeding Up a Reorganization

Use

You can speed up a reorganization by changing certain parameter values in the initialization profile **init<DBSID>.ora** before starting the reorganization.

Prerequisites

Only make these changes for the duration of the reorganization. Reset the old values after the reorganization.

Make sure that the R/3 System has been shut down before you start up the database.

Procedure

1. Write down the original values of parameters **sort_area_size** and **db_file_multi_block_read_count**.
2. Change the parameter **sort_area_size** to **8388608**: This increases the sort buffer to 1024 blocks of 8 KB each.
3. Change the parameter **db_file_multiblock_read_count** to **64**: This increases the number of blocks read or written for each I/O operation.
4. Start up the database in *DBA* mode (the R/3 System must be shut down) and perform your reorganization.
5. After the reorganization, return the parameters to their original values before you start up the database and the R/3 System for production use.

Archiving Mode during the Reorganization

Since a reorganization always involves the export and import of mass data, a large number of redo log files are written. If you deactivate the ARCHIVELOG mode, there is no significant acceleration of the reorganization. The archive process ARCH runs parallel to the remaining background database processes, this means that only the CPU is briefly more heavily loaded if you operate the database in ARCHIVELOG mode. Thus, the reorganization is not significantly slowed down.

Advantages of a Reorganization in NOARCHIVELOG Mode:

- You can safeguard against the risk of an archive stuck. In any case, you save space in the archiving directory of the offline redo log files `<ORACLE_HOME>/saparch`.
- BRARCHIVE does not have to run for the period of the reorganization, as no offline redo log files are created.
- Time can be saved in the event of a recovery, as you can resume from the backup performed after the reorganization (this is mandatory with this method).

Data Security for a Reorganization in NOARCHIVELOG Mode:

SAP only recommends deactivating ARCHIVELOG mode under certain conditions, since it interrupts the continuity of the redo log files. Once the reorganization is complete, it is essential that you reactivate the ARCHIVELOG mode and perform a complete offline backup of the database.

[Reorganizing in NOARCHIVELOG Mode \[Page 626\]](#)

Reorganizing in NOARCHIVELOG Mode

Reorganizing in NOARCHIVELOG Mode

1. We recommend performing an online or offline backup of the entire database.

A complete backup of the database is not required if all redo logs since the last backup are available. If this is the case, it is sufficient for security purposes to use the ORACLE command `ALTER SYSTEM SWITCH LOGFILE` to force redo log file switching. This means that the online redo log files, which were not yet backed up by ARCH (this normally only concerns the current online redo log group), can also be written in the archiving directory.
2. Shut down the database.
3. Deactivate the ARCHIVELOG mode. To do this, use the SAPDBA function or the ORACLE command `alter database noarchivelog` (see [SAPDBA: Archive Mode \[Page 31\]](#)).
4. You can now carry out the required reorganization.
5. Once the reorganization is complete, you **must** reactivate the ARCHIVELOG mode.
6. It is essential that you now perform a complete backup of the database, since recovering the database with an old backup would only be possible to the state before the reorganization (remember that the reorganization was not performed in ARCHIVELOG mode).

Only an **immediate offline backup** can offer complete security. It is therefore best to start an offline backup of the entire database.

Although an online backup is possible, it conceals a certain risk if every data file was not backed up. If the online backup - for any reason - terminates, a recovery can only proceed to the state before the reorganization (starting from the last backup before the start of the reorganization). All changes made in the database during the online backup would be lost and would have to be repeated after the reorganization.

Performing Processes in Parallel

You can speed up a reorganization by performing the following operations in parallel:

- Data transport (ORACLE exp/imp, SAP unload/load, SQL*Loader)

The export (or import) is performed in parallel by setting the degree of parallelism (parallel option or profile parameter `exp_imp_degree`) >1 and distributing the tables to be exported on several different export processes. A corresponding number of export dump files is generated. The different export processes can be processed either sequentially or in parallel.

The time for the data export/import (with the ORACLE tools Export/Import) depends on the amount of data. These procedures take a large part of the total organization time within the reorganization process. You should perform the export/import in parallel for this reason.

Information can be found in the ORACLE documentation and in: [Parallel Export/Import \[Page 643\]](#).

[exp_imp_degree \[Page 274\]](#).

- Creating objects (tables with CREATE TABLE AS SELECT, indices with CREATE INDEX)

The creation of large tables and their indices can take a very long time. You should, therefore, make these processes parallel. More information can be found in the ORACLE documentation.

SAPDBA supports the parallel creation of tables and indexes if you proceed as follows:

- Set the degree of parallelism with the parameter `index_degree` in the profile `init<DBSID>.dba` (also for tables).
See [index_degree \[Page 277\]](#).
- Interactive entry of the degree of parallelism in the corresponding reorganization menu
Select the menu option *Object handling* → *Parallel object creation* or (with CREATE TABLE AS SELECT) *Data transfer* → *Parallel table creation*. SAPDBA displays the default value set in `index_degree`. Enter the required degree of parallelism.

It does not make sense to perform the specified processes in parallel if you select a degree of parallelism which is significantly larger than the number of existing CPUs (the degree of parallelism should not be larger than two or three times the number of CPUs).

Restarting a Reorganization

Restarting a Reorganization

If an error occurs during the reorganization process, the system issues an appropriate warning message. If the error occurred during the main phase of the reorganization, you can continue the reorganization with the Restart option, once the error has been eliminated. **Pay attention** to both SAPDBA and ORACLE messages. If serious errors occur, the reorganization process is terminated at the appropriate point.



If you overwrite the default answer [N] to the question: **Error in... - Do you want to continue<Y/N> [N]** with Y(es) and continue the process despite the warnings, you alone are responsible for any and all consequences. SAPDBA continues the processing as if the error never occurred.

Examples of Error Situations

- Errors during the production of the reorganization scripts (preparation phase): Nothing has been altered in the database. The error can be ignored and the reorganization repeated (a restart is not possible because some of the scripts are still missing or may be incorrect).
- Errors during processing of scripts up until completion of exports (scripts have been completely created and processing has begun in the main phase):
 - Before the export, tables were renamed (`HideTab` option) or constraints were deleted (i.e. the database has been altered in this respect). A restart of the reorganization is possible without any further action after correcting the cause of the termination. If you want to completely repeat the reorganization, you must reproduce the deleted constraints, or cancel the table renaming. To do this, you can use the [Resetting a Reorganization \[Page 634\]](#) option in the restart menu (process script `con<TSP>.sql` or `ren<TSP>.sql`).
 - Before the export, no tables were renamed and no constraints were deleted (this means that the database is unchanged in this respect). A restart or a complete repeat of the reorganization is possible.
- Errors during processing of one of the scripts after the export script (in the main phase): tables, indexes or tablespaces have already been deleted. In this case you cannot repeat the reorganization. Instead, you must make a restart after fixing the error (or reproduce the initial status manually).

To restart the reorganization process after a termination, you can use the SAPDBA restart function.

To do this, proceed as follows:

- Examine the logs as *Preparing for the Restart*.
- Use the SAPDBA menu *Restart reorganization* to execute the *Restart*.

[Preparing for the Restart \[Page 630\]](#)

[Restart Procedure \[Page 631\]](#)

[Restart File for Reorganizations \[Page 632\]](#)

[Resetting a Reorganization \[Page 634\]](#)

Preparing for the Restart

Preparing for the Restart

In the SAPDBA menu, select *Show/Cleanup* → *Show log files/profiles* → *SAPDBA log files*. Select the log written during the operation that failed, and examine the process flow and (possibly) the cause of the error.

Name of log: <timestamp>.<ext>

The timestamp and function ID help you to find the right log. The timestamp indicates when the corresponding process was started.

With the default settings, SAPDBA saves these logs in <ORACLE_HOME>/sapreorg. This directory also contains the subdirectory <timestamp>, where SAPDBA places the scripts required for the process and the restart file.

See [Restart File for Reorganizations \[Page 632\]](#), [Overview of Script Types \[Page 289\]](#).



Always make sure you have the right log and the right restart file before you restart a reorganization.

Restart Procedure

Select *Restart reorganization* (similar to *Restart export/import* for restarting an export/import). The following menu items are displayed:

<i>a</i>	- <i>Working directory:</i>
<i>b</i>	- <i>Restart subdirectory:</i>
	- <i>function:</i>
<i>p</i>	- <i>Show restart plan</i>
<i>r</i>	- <i>Reset failed reorganization</i>
<i>s</i>	- <i>Restart</i>
<i>t</i>	- <i>Restart in batch</i>
<i>q</i>	- <i>Return</i>

During every reorganization (or during an export/import), SAPDBA generates a subdirectory <timestamp>, which contains both the scripts and the restart file. The SAPDBA default value for the *Restart subdirectory* is the <timestamp> directory of the last operation performed. SAPDBA displays a short description of the operation via *function*.

Before you restart the required operation, you should - for safety's sake - find out which scripts will be processed. To do this, select *Show restart plan*.

Select *Restart* to start the restart process immediately or *Restart in Batch* if you want the restart process to run in the background.

Select *Reset failed reorganization* to reset a reorganization to its original state if it terminated before or during the export (see [Resetting a Reorganization \[Page 634\]](#)).

Restart File for Reorganizations

Restart File for Reorganizations

Name of a restart file: <ORACLE_HOME>/sapreorg/<timestamp>/restart.<ext>

Function ID <ext>: Indicates the SAPDBA function for which the corresponding scripts were written to the file.

List of possible function IDs during a reorganization

rsi	Reorganization of single tables or indexes
rli	Reorganization of a list of tables or indexes
rtc	Reorganization of tablespaces
rtd	Reorganization of tablespaces with data files
rmv	Change the data file allocation for a tablespace (move/rename data files of a tablespace)

These values are identical with the extension of the corresponding log, <ORACLE_HOME>/sapreorg/<timestamp>.<ext>.

The restart file contains the scripts required for performing the corresponding reorganization - that is, a subset of the following scripts:

```
alt<TSP>.sql, drc<TSP>.sql, exp<TSP>.sh, exc<TSP>.sh, unl<TSP>.sh,
swi<TSP>.sql, dro<TSP>.sql, del<TSP>.sh, ren<TSP>.sql, tsp<TSP>.sql,
tab<TSP>.sql, imc<TSP>.sh, imp<TSP>.sh, ldr<TSP>.sh, ind<TSP>.sql,
imx<TSP>.sh, inx<TSP>.sql, con<TSP>.sql, grants.sql
```

For a complete overview of the scripts that are written and used by SAPDBA, please refer to [SAPDBA Scripts and Files \[Page 291\]](#). Also note the changes to the scripts when you are using the tools SAP unload/SAP load or SAP unload/SQL*Loader (see [SAP Unload/Load, SQL*Loader \[Page 665\]](#)).

See also: [Reorganization of a Single Object \[Page 583\]](#), [Reorganization of a List of Objects \[Page 584\]](#), [Reorganization of a Tablespace \[Page 586\]](#), [Reorganization of a Tablespace with Data Files \[Page 588\]](#) and [Moving/Renaming the Data Files of a Tablespace \[Page 590\]](#).

Script Indicators

When a reorganization is started, the scripts in the restart file are processed in sequence. The indicator before each script indicates whether it was successfully processed or not.

The following indicators are possible:

Y	Script processed successfully.
N	Script either not yet processed or failed. Before a reorganization is started, all scripts have this indicator.
S	The data import (import or loader script) has already been started but SAPDBA could not process the import script properly due to an external error (e.g. import interrupted by user or power failure). SAPDBA terminated. Scripts indicated with S will automatically be newly processed in the event of a restart.

Restart File for Reorganizations

When using SAP unload/load or ORACLE SQL*Loader the same indicators also appear in the parameter file `ldr<TSP>.par`.

During a reorganization, all the scripts indicated with N are processed, and are changed to Y once the respective operation has been completed (exceptions are the scripts: `imp<TSP>.sh`, `ldr<TSP>.sh : N → S → Y`, in error condition `N → S → N`).

If the import did not run correctly the indicator of the corresponding script remains S. When restarting the reorganization in this case the contents of all the tables which were already imported are deleted, the indicator is reset to N and the script `imp<TSP>.sh` is processed again. SAPDBA deletes the contents of tables found in the script `exp<TSP>.par` (with the ORACLE command TRUNCATE).

If several export dump files are generated during export/import, [SAP Unload/Load, SQL*Loader \[Page 665\]](#) SAPDBA only deletes the contents of the tables found in script `exp<TSP>_<n>.par` of the canceled import process (`imp<TSP>_<n>.sh`).

During a loader restart only the data in the table where the interruption occurred is deleted. During loading every table is marked with a restart indicator ('S') in the parameter file `ldr<TSP>.par`.

See also

“Entries in Export dump destination - Script Names” in [SAPDBA Scripts and Files \[Page 291\]](#).



Never change the sequence of scripts that appears in the restart file! It determines the flow of the corresponding reorganization.

Resetting a Reorganization

Resetting a Reorganization

The Reset option is intended for use in a reorganization which terminated due to an error. You should only use it if a reorganization already terminated before or during the export.

If tables were renamed before the reorganization (see the *HideTab* option in the [SAPDBA: Reorganization Menu \[Page 615\]](#)) and/or constraints were deleted (only during export to tape), resetting will cancel these actions. (SAPDBA deletes primary and unique keys before the export.) All the `con<TSP>.sql` and `ren<TSP>.sql` scripts are edited automatically during a reset. The database objects are set back to the state they had before the reorganization. You must check whether all the scripts were executed correctly using the logs and the restart plan.

Resetting a reorganization is usually only worthwhile when you do not want to carry out a restart of the same reorganization afterwards. If, despite this, you still want to implement a restart (after the reset), then before the restart you must manually set back all the 'Y' status indicators to 'N' in the restart plan.

It is not possible to reset if tables, indexes or the tablespace were already deleted after a successful export. The reorganization has to be followed by a restart once the error has been corrected.

See: [Restarting a Reorganization \[Page 628\]](#).

Data Transport During a Reorganization

To reorganize a table the contents must be saved and then entered into the newly created table. This process is decisive for the total length of time the reorganization takes. The exact procedure depends on the method used for data transport. The following methods or tools are available:

1. ORACLE export / ORACLE import
2. SAP unload / ORACLE SQL*Loader
3. SAP unload / SAP load
4. CREATE TABLE AS SELECT

In methods 1.-3. the data transport proceeds as follows: The data is first exported or unloaded from the table to be reorganized into a backup file. After reorganization the data is imported or loaded back into the table.

In the CREATE TABLE AS SELECT method the new table is created parallel to the old table and the contents of the table to be reorganized are copied into it. The old table is then deleted.

The length of time the reorganization (i.e. the data transport) takes is shortened significantly with the CREATE TABLE AS SELECT method in comparison to the others.



Note the restrictions and details of the various methods of data transport by reading the following chapters:

[SAP Unload/Load, SQL*Loader \[Page 665\]](#)

[SAPDBA: Reorganization Menu \[Page 615\]](#)

[Reorganization with CREATE TABLE AS SELECT \[Page 636\]](#)

Reorganization with Create table ... as select

Reorganization with `Create table ... as select`

Use

`Create table ... as select` (alongside ORACLE EXP/IMP, SAPDBA LOAD/UNLOAD and ORACLE SQL*Loader) is a method for making a data transport during a reorganization, so significantly reducing the total duration of the reorganization. You can use it for all types of reorganization except for table with long columns and tablespaces reorganizations with data files. This method is particularly time-saving when you transport a single table into another tablespace.

Prerequisites

If the command `Create table ... as select` is interrupted (for example, if there is sufficient free space, but too much fragmentation) the affected table is not deleted, so guaranteeing data security at all times. If this happens, the reorganization process is continued with the next table.

Note that the amount of memory space required by a table to be reorganized rises temporarily in the accompanying tablespace, as the old and the new tables coexist during the copying process. In addition to this, the free space of a tablespace is not combined optimally, since the tables are not deleted simultaneously.

Tables whose names are already 30 characters long are not reorganized.

Features

SAPDBA creates a new name for the table to be reorganized with the new parameters by appending the # character and copies the data directly from the old table to the new table. If this is completed successfully then the old table is deleted and the new table is renamed with the old name. In this way a single table, or all tables in an object list or tablespace, which are to be reorganized (reorganization without data files) are processed.

Exporting and Importing Database Objects

With export/import you can carry out backups of database objects in addition to other database backups. Export is especially suitable as a backup method when you only want to save particular tables. For example you can add to the data backup you carry out before a reorganization by exporting the objects that are to be reorganized. You can also backup the table and index definitions in your database.

The function *export/import* is only intended for the export of objects out of and the import of objects into a single database.

Note that these data backups are logical backups. This means that you cannot use them to carry out any ORACLE restore procedure. The exported objects are static and are only consistent with the database if it is not subjected to any changes. See

The SAPDBA function *export/import* carries out the export phase and, if required, the import phase as well. The function *export/import* generates scripts for both phases which contain the necessary commands. If necessary these scripts can also be started manually.

Export/import can be divided into 3 phases:

Phase 1:

Creation of scripts for the export and import. The export of tables and index definitions or other database objects (constraints, grants, views, synonyms and sequences) is all contained in this first phase.

Phase 2:

Processing of created export shell scripts (in the case of an export of tables with data and their indexes).

Phase 3:

If required, the data or the table and index definitions or the other database objects (constraints, grants, views, synonyms and sequences) can be imported.



Do not use export/import for the transport of database objects between databases

The logical structure of SAP data is so complex that data objects are often distributed across many tables and many tables are linked largely according to the relational database model. Attempting to transport data between systems using export/import will, therefore, lead to inconsistencies in the whole R/3 System.

You should create new SAP database with the help of the SAP installation procedure. Use the SAP correction and transport system to transport objects between R/3 Systems.

Exporting and Importing Database Objects**Data Consistency**

Please note that the exported objects are only saved logically. This means they cannot be used for an ORACLE recovery procedure. The exported objects are static, and are only consistent with the database as long as it is not changed.



If you schedule an export as a background process, tables which are only generated after the export scripts have been created are not taken into consideration when the export is started.

SAPDBA: Export/import

	Export/import
<i>a</i>	- Export - tables and indexes including data
<i>b</i>	- Export - structures of tables and indexes
<i>c</i>	- Export - constraints, synonyms, sequences, views, grants
<i>d</i>	- Import - tables and indexes including data
<i>e</i>	- Import - structures
<i>r</i>	- Restart
<i>q</i>	- Return

The SAPDBA function **Export/import** allows you to carry out the following actions:

- Export of database objects
You can export tables with their data, table and index definitions, and with other database objects such as constraints, grants, views, synonyms and sequences.
- Import of objects in the database which have earlier been exported.

You can find a description of the various types of export under

[Export Types \[Page 641\]](#).

In [SAPDBA: Export Menu \[Page 644\]](#) you can choose one of the following **methods** (tools) for the data transport:

- ORACLE export / ORACLE import (SAPDBA default)
- SAP unload / ORACLE SQL*Loader (from ORACLE 7.2)
- SAP unload / SAP load



One advantage of using the tool SAP unload/load, ORACLE SQL*Loader is the significant reduction in the length of time the data transport takes (up to 50%) in comparison to ORACLE export/import. For other advantages of this method see [SAP Unload/Load, SQL*Loader \[Page 665\]](#).

The so-called structure export (without data) of table definitions and index definitions, and of constraints, grants, views, synonyms and sequences, and their import (structure import) is not carried out with the tools described above, but with SAPDBA-internal methods.



In the following the terms export/import stand for the data transport procedure in general without specifying a particular method.

SAPDBA: Export/import

In order to accelerate the data transport process you can perform the export and the import processes in parallel:

[Parallel Export/Import \[Page 643\]](#)

Export Types

You can use the **Export/import** function to carry out three types of export procedure:

- Export of tables with their indexes and their data (rows). Other database objects, such as constraints and grants, are also contained in this export. For this export use the menu item *Export tables and indexes including data*. To import this data, use *Import tables and indexes including data*.
- Export of table and index definitions (structure export). For this export use the menu item *Export structures of tables and indexes*. To import this data, use *Import structures*.
- Export constraints, synonyms, sequences, views, and grants from a database (structure export). For this export, use the menu item *Export constraints, synonyms, sequences, views, grants*. To import this data, use *Import structures*.

The following table gives an overview of the individual data/application flows.

Description of Export Types

Objects for export	Procedure
Tables with indexes and data	<p>SAPDBA exports the table(s) you specify with their indexes and data. Indexes are exported even if they are not in the same tablespace as their table(s). SAPDBA exports all constraints, comments and grants of the selected tables.</p> <p>You can optionally modify the parameters INITIAL, NEXT, MINEXTENTS, MAXEXTENTS, PCTFREE, PCTUSED and FREELISTS or have SAPDBA automatically calculate the value for the INITIAL and NEXT parameters (PCTINCREASE is always set to 0).</p> <p>SAPDBA creates import scripts that are used by SAPDBA for importing the data.</p>
Tables/index definitions	<p>SAPDBA creates an import script that can be used to recreate definitions of tables and indexes that you specify. If you store table definitions, the index definitions referring to the tables are also stored, even if the indexes are stored in another tablespace than the tables.</p> <p>Definitions contain table and column comments, constraints, and grants.</p> <p>You can optionally modify the parameters INITIAL, NEXT, MINEXTENTS, MAXEXTENTS, PCTFREE, PCTUSED and FREELISTS or have SAPDBA automatically calculate the value for the INITIAL and NEXT parameters (PCTINCREASE is always set to 0).</p>

Export Types

Constraints, grants, synonyms, sequences, and views	SAPDBA creates an import script that can be used by SAPDBA to recreate these objects. The export is carried out globally: all constraints, grants, synonyms, sequences, and views of the database are included (you can optionally exclude objects of users SYS and SYSTEM from this process by using <i>incl.owners SYS/SYSTEM: no</i>).
---	--

Parallel Export/Import

The export as well as the import can be distributed to several parallel processes. This also applies to exports or import processes running during a reorganization. (Exception: *Reorganization of a single table or index*).

exp_imp_degree = 1

If the default value for the degree of parallelism ([exp_imp_degree \[Page 274\]](#) =1) is not changed, all export/import processes are executed sequentially. The <timestamp>-directory created in <ORACLE_HOME>/sapreorg is proposed as the target for the export dump file(s) (if the following remains unchanged: work directory = export dump file(s) directory = directory in [exireo_dumpdir \[Page 271\]](#)).

exp_imp_degree > 1

If several export dump files are being created, then the export/import can be executed in parallel (degree of parallelism no greater than the value of `exp_imp_degree`). It is possible to specify a different or further directories and/or tape devices for the export dump files (to specify the *Export dump destination* see [SAPDBA: Export Menu \[Page 644\]](#)). The same directories may be specified several times. SAPDBA always tries to achieve the maximum degree of parallelism.

One export dump file is generated from each of the parallel processes. SAPDBA distributes the tables to the individual export dump files on the basis of their size, to ensure that the files all have approximately the same size. However, SAPDBA also takes the size of the file systems or tapes into account.

If the number of parallel processes defined (by specifying directories or tape devices in *Export dump destination*) is greater than the number of tables that are to be exported, then only one process is used for each table. The remaining directories and tape devices are not used.

If the number of directories and tape devices specified (in *Export dump destination*) is greater than the maximum number of parallel processes defined in `exp_imp_degree`, then as many directories and tape devices are used in parallel as necessary to achieve this degree of parallelism. As soon as one of these processes ends, the other directories and tape devices available will be used by the next process.

The option *Compress export dump: yes* enables you to compress the export dump files in all the directories on disk. Parallel export/import is also possible when this option is used.

This option has no effect when exporting to tape; use hardware compression instead.

See also:

“Specifications in Export Dump Destination - Script Descriptions” in [SAPDBA Scripts and Files \[Page 291\]](#),
[Performing Processes in Parallel \[Page 627\]](#),
and [Logging of Parallel Export/Import Processes \[Page 304\]](#).

SAPDBA: Export Menu

SAPDBA: Export Menu

In data export, the following menu options appear for selection:

<i>a</i>	- <i>Tablespaces</i>		
<i>b</i>	- <i>Owners</i>		
<i>c</i>	- <i>Tables</i>		
<i>d</i>	- <i>Working directory</i>		
<i>e</i>	- <i>Dump destination</i>		
<i>f</i>	- <i>Export/Import</i>	<i>g</i>	- <i>Storage parameters</i>
	<i>Compress</i>		<i>ComprExt</i>
	<i>Chop</i>		<i>SAP-NEXT</i>
	<i>CheckExp</i>		
	<i>Commit</i>		<i>ReduceOb</i>
	<i>Parallel</i>		<i>Manually</i>
<i>s</i>	- <i>Start</i>		
<i>q</i>	- <i>Return</i>		

If you want to use a method other than the default method (ORACLE export/import) for the data transfer, choose menu option *f*. The menu option *f* in the above export menu will be adjusted accordingly, for example after choosing *SAPunload / ORACLE SQL *Loader* the following will appear in the export menu:

<i>f</i>	- <i>Unload/load</i>
	<i>SAP unload</i>
	<i>SQL *Loader</i>
	<i>Commit :</i>
	<i>Parallel :</i>

Specify the tables for export. You can select the tables by tablespace, owner, and name of the object. You can enter a single name, the identifier "all", a list of names, or a generic specification in the fields for the identification of the object (tablespaces, owners, tables) :

- Specifying "all" means that you select all objects.
- The lists must have the following format: Name, Name,...
- Make generic specifications with the Oracle placeholder "%" (corresponds to the LIKE function in Oracle). All the objects having this LIKE pattern would be exported.

Special feature when selecting the tablespaces: With *PSAPBTAB%* you would generate a list of tablespaces (*PSAPBTABD* and *PSAPBTABI*) from which you could select one tablespace. If you want to export both the objects of *PSAPBTABD* as well as the objects of *PSAPBTABI*, you can do so by entering *PSAPBTAB%%*.

SAPDBA: Export Menu

You can export ORACLE system objects if you select *Owners: "all" or "%" and then Incl. owners SYS and SYSTEM: yes.*

The most important menu items are listed below:

- **Working directory:** Default: <ORACLE_HOME>/sapreorg
You can change this default with the `init<DBSID>.dba` parameter [exireo_dumpdir \[Page 271\]](#).
Working directory in which you store the time stamp subdirectory with the scripts and parameter files. You can change this working directory if required. Logs are always stored in <ORACLE_HOME>/sapreorg.
- **Dump destination:** Default: <ORACLE_HOME>/sapreorg
You can change this default with the `init<DBSID>.dba` parameter [exireo_dumpdir \[Page 271\]](#).
Directory/ies (or tape device(s)) to which the export dump files are written (if a data export takes place). The total amount of space available in the directories or on the tapes should be big enough to at least store all the tables that are to be exported. All further scripts and logs are always written to the working directory.
The export or import can be run in parallel by defining the parameter [exp_imp_degree \[Page 274\]](#) > 1. See [Parallel Export/Import \[Page 643\]](#)
SAPDBA warns you if there is not enough space available. You may nevertheless wish to continue with the export. The data written to the directories on the disk can be compressed by a certain factor by using the option *Export/import method* → *Compress dump file(s): yes* (if ORACLE Export/Import is used). For security reasons, the space check does not take the affects of compression into account (the compression rate and degree of filling is not known exactly).
When you export to tape, SAPDBA asks you to state the size of the tape. SAPDABA then also checks whether there is enough space on the tape(s) for the data that you want to export. In contrast to an export to disk(s), where an export dump file is created for each tablespace, an SAPDBA export to tape creates only one export dump file for each tape for all tables of the specified tablespaces. SAPDBA can only use one tape drive for each export dump file. You are warned, if the total amount of space required for a dump file is greater than the tape size. If SAPDBA is used for exporting the ORACLE program Export, errors occurring when writing to tape during the ORACLE export are ignored.
The menu option *Reduce object size: yes* calculates the export dump accurately. SAPDBA determines the space actually occupied by the data of the object and carries out the space check on the basis of this value. The compression rate is, however, also not taken into account here.
You can specify the Null device **/dev/null**. You should do so for testing tables and indexes on corrupted ORACLE blocks. In this case the export is performed as usually but no export dump file is created. You can also make this check using the SAPDBA command option `sapdba -export <tablespaces/table>` (see [SAPDBA Command Mode \[Page 69\]](#)).
- **ORACLE export/import, SAP unload/ORACLE SQL*Loader or SAP load/unload:**
Default: `ORACLE export/import`
You can decide which tool you want to use for the data export/import here. You can also set some parameters for these programs if required.

SAPDBA: Export Menu

Using SAPunload/load or ORACLE SQL*Loader can save a significant amount of time in the data transfer in comparison with ORACLE export/import (such as with tables with LONG columns).

See [SAP unload/load, SQL*Loader \[Page 665\]](#).

- **Buffer size:** Default: 3000000

You can change this default with the `init<DBSID>.dba` parameter [exireo_dumpdir \[Page 272\]](#).

To accelerate the export/import, it is recommended that you make a buffer of at least 3 MB available for the ORACLE export and import programs or for the load programs.

- **Compress (Compress dump file(s)):** Default: no

Select `yes` if the data should be compressed during export.

In this case, SAPDBA sends the data with the UNIX command `compress` before writing it in the directory/ies for the export dump file(s). You can reduce your disk space requirements considerably if you compress your data.

Compression is only possible for an export (with ORACLE export) to disk. This option will be ignored if you export to tape. Therefore, use tape stations with hardware compression where possible.

Do not use compression for tablespaces containing objects already compressed by the SAP database interface, since there is no advantage to be gained (there are more likely to be disadvantages).

(If you set both *Compress* and *CHOP* to *Yes*, compression will use the R3chop option `-c` and not the UNIX command `compress`.)

- **Chop (Chop dump file(s)):** Default: no

If you want to export very large tables/tablespaces (with ORACLE export) and you expect that the export dump files which will be generated could exceed the maximum file size of the operating system, set option *Chop* to *YES*. In this case SAPDBA uses the program R3chop, which splits the dump files into smaller files and then combines them again. You are prompted for the maximum size of the split files. The default value corresponds to the value set in parameter [max_file_size \[Page 278\]](#) (whose default value is 2 GB).

This option is only available if parameter [chop_util_name \[Page 269\]](#) is entered in profile `init<DBSID>.dba`.

- **CheckExp (Check dump file(s)):** Default: yes

After the ORACLE export, SAPDBA performs a read check if you select *CheckExp: Yes* (test import with ORACLE parameter `INDEXFILE`). The `inx<TSP>.sql` scripts are created. This option should be used in particular if you are exporting to tape.

- **Commit (Commit during import(load)):** Default: yes

With *Commit:: yes*, the command `COMMIT` is passed to the database after the data contained in the buffer has been imported.

With *Commit:: no*, the `COMMIT` is only performed after all the records of a table have been imported.

- **Parallel (Parallel export and import):** Default: degree 1

The default value corresponds to the value of the parameter [exp_imp_degree \[Page 274\]](#) in the profile `init<DBSID>.dba`. The export/import can be performed in parallel by selecting a degree of parallelism >1 (see [Parallel Export/Import \[Page 643\]](#)).

- **Storage parameters:**

You can make various settings for the storage parameters here:

- **ComprExt (Compress extents):** Default: `yes`
SAP-NEXT (Use ABAP-Dic for NEXT): Default: `yes`
ReduceOb (Reduce object size): Default: `no`
Manually (Change storage manually): Default: `no`

Before you export objects, you can change the extent sizes and the other storage parameters proposed by SAPDBA. See [Options for Changing and Checking the Storage Parameters \[Page 671\]](#)

Exporting Tables with Indexes and Data

Exporting Tables with Indexes and Data

This section describes the menu item **Export tables and indexes including data** of the export/import menu.

With this function, you can export one or more tables. Tables are exported with their data, their indexes, and their constraints, comments and grants. The function creates an import script with which the data can be imported. For more information, see [Importing Tables with Indexes and Data \[Page 658\]](#).

[Requirements \[Page 649\]](#)

[Procedure \[Page 650\]](#)

[SAPDBA: Export Menu \[Page 644\]](#)

Prerequisites for Export and Import

To ensure a successful export and import, make sure that the following conditions are met:

- **Storage space in the dump directory/on tape:** The file systems or tapes to which the export dump files are written must have sufficient storage space to hold the objects for export. SAPDBA checks whether the contents of the objects to be exported fit in the directories or onto the tapes, and issues a warning if they are too large.

If the parameter [exp_imp_degree \[Page 274\]](#) is > 1, then this represents the number of processes that execute in parallel. SAPDBA tries to distribute the tables equally across the specified directories/tapes.



Tapes can only be used when implementing ORACLE export/ import, however not when implementing SAPunload/load and SQL*Loader.

- **Tablespace:** Before you import tables, you must make sure that the tablespaces to which the tables and the indexes belong exist.

The import script contains all commands for creating tables and the indexes, and for importing data, but not the commands for recreating any tablespaces.

SAP indexes are generally held in tablespaces - on separate disks - other than those of the tables to which they refer.

- **Storage space:** The tablespaces must have enough free storage space to hold the table and the indexes. In contrast to reorganization, SAPDBA does not check whether the import into the database will run successfully.

Export Options for Tables and Indexes with Data

Procedure

1. Start the database system in [Restricted Session \[Page 63\]](#) mode (recommended), if this has not been already done. The R/3 System must be shut down.
2. Select the SAPDBA export option for a table with its data (*Export tables and indexes including data*).
3. The following menu options are displayed (see [SAPDBA: Export Menu \[Page 644\]](#)):
4. Start the export. SAPDBA starts the ORACLE *Export* program.

Preliminary Result

- The necessary scripts and the restart file **restart.exd** are created.
- A log **<timestamp>.exd** is created.
- The table definitions (and comments), the definitions of the table indexes, all constraints and grants, and the data are exported with Oracle *Export* or SAP unload.

If an error occurs, you can repeat the data export with the restart function (see [Restarting the Export/Import \[Page 662\]](#)).

5. If necessary, you can import the table(s) and their corresponding objects again - use the import function of SAPDBA *Import tables and indexes including data* to start the import scripts. You can also start structure imports (SQL scripts) singly (SAPDBA function *Additional functions* → *Execute SQL script*).

Preliminary Result

Objects that exist in the database are not overwritten by an import. However, rows may be doubled if they already existed prior to the import and no *UNIQUE* index existed. Before you make the import you must make sure that the objects that are to be imported do not already exist, or that the tables are empty.

See [Data Consistency \[Page 638\]](#) and [Importing Tables with Indexes and Data \[Page 658\]](#).



When you use SAP unload, SQL scripts are always created for indexes and grants, since in contrast to an Oracle *Export*, the SAP unload only exports the tables with their data.

See [SAP unload/load, SQL*Loader \[Page 665\]](#).

Script Categories

During an export of a table with indexes and data, the following scripts and/or parameter files are used (if required) or are created for the later import or restart:

- **alt<TSP>.sql**
- **exp<TSP>.sh** (**exc<TSP>.sh** if necessary) or **unl<TSP>.sh**
- **exp<TSP>.par** (**exc<TSP>.par** if necessary) or **ldr<TSP>.par**

Export Options for Tables and Indexes with Data

- **tab<TSP>.sql**
- **imp<TSP>.sh** (**imc<TSP>.sh**, **imx<TSP>.sh** if necessary) or **ldr<TSP>.sh**
- **ind<TSP>.sql** (**inx<TSP>.sql** if necessary)
- **con<TSP>.sql**
- **restart.exe** (valid both for the export and for the possible import)

There are certain circumstances in which multiple scripts of the type *.sh and *.par are generated (see [SAPDBA Scripts and Logs \[Page 291\]](#)). The script **alt<TSP>.sql** is processed before tables with data are exported. It contains *ALTER* statements for those indexes that belong to a primary key. However, this applies only if one of the index parameters *NEXT*, *PCTINCREASE*, *MAXEXTENTS*, *PCTFREE* or *FREELISTS* has been changed (automatically or manually in the menu). For these indexes *MINEXTENTS* cannot be changed and *INITIAL* can only be changed using the option *Compress extents yes*.

When using Oracle *Export* the script **ind<TSP>.sql** is generated only if the index is exported without its table. When using SAP unload **ind<TSP>.sql** is always generated.

Grants are exported or imported with the Oracle programs *export/import*. Comments are always included in the script **tab<TSP>.sql**. SAPDBA creates the separate scripts **grants.sql** for grants when using the tools SAP unload/load, SQL*loader.

See [Restart File for Export/Import \[Page 663\]](#) and [SAPDBA Scripts and Files \[Page 291\]](#).

Exporting Table and Index Definitions

Exporting Table and Index Definitions

This section describes the menu item **Export structures of tables and indexes** of the export/import menu.

With this function, you can export the definitions of one or more tables or indexes. The function creates import scripts with which you can recreate the tables and the indexes. See [Structure Import \[Page 660\]](#).

A table definition contains table and column comments and the table constraints.

SAPDBA exports indexes that refer to a table even if they are stored in another tablespace. (Most of the indexes in the SAP database are stored in a different tablespace than that of the tables to which the indexes refer).

[Procedure \[Page 653\]](#)

Export Options for Tables and Indexes

Procedure

1. Start up the database system. You can make this export while the database and the R/3 System are active. This does not affect other users.
2. Choose the SAPDBA for export option for table and index definitions (*Export structures of tables and indexes*); the following menu appears:

a	- Tablespaces
b	- Owners
c	- Tables
d	- Working directory
g	- Storage parameters
	ComprExt
	SAP-NEXT
	ReduceOb
	Manually
s	- Start
q	- Return

1. Specify the tables and indexes that you want to export. Do this by the tablespace name, its owner or the table name. Make any other settings that you want. (see [SAPDBA: Export Menu \[Page 644\]](#)).
2. Start the export. SAPDBA starts the Oracle program *Export*. Import scripts are generated that contain the appropriate table and index definitions. SAPDBA creates the restart file `restart.exs` and the log `<timestamp>.exs`. If there is an error you must repeat the export of the structure. You can use the scripts in `restart.exs` to start an import of the objects later. If there are errors during the import you can repeat it using the restart function. (see [Restarting the Export/Import \[Page 662\]](#)).
3. If you need to re-import the definitions, use the SAPDBA function *Import structures*. You can start the appropriate scripts manually. Use the SAPDBA function *Additional functions* → *Execute SQL script*. for the SQL scripts.

The import does not overwrite any definitions that already exist in the database (see [Data Consistency \[Page 638\]](#) and [Structure Import \[Page 660\]](#)).

Script Categories

In an export of table and index definitions, the following scripts are used or created for a later import or restart:

- `tab<TSP>.sql`
- `ind<TSP>.sql`
- `con<TSP>.sql`
- `grants.sql`
- `restart.exs`

Export Options for Tables and Indexes

See [Restart File for Export/Import \[Page 663\]](#) and [SAPDBA Scripts and Files \[Page 291\]](#)

Exporting Constraints, Synonyms, Sequences, Views, and Grants

This section describes the menu item **Export constraints, synonyms, sequences, views, grants** of the export/import menu.

With this function, you can export the definitions of the database objects listed below.

- Constraints: Object requirements, such as interval of values allowed for a field
- Synonyms: Secondary identifications of defined database objects, such as official synonyms for DBA views (ORACLE Dictionary views)
- Sequences: Number intervals that, among other things, are used by SAP for certain network synchronization functions
- Views: Selections of data (can be from different tables) according to certain criteria
- Grants: Authorizations for using database objects

The function creates an import script that you can use to recreate the objects in the database, if this should become necessary. See [Structure Import \[Page 660\]](#).

All user objects (and the system objects, when necessary) are included in the export.

[Procedure \[Page 656\]](#)

Export Options for Constraints, Synonyms, Sequences, Views, and Grants

Procedure

1. Start the database system if it is not already active. You can carry out this export with an active database and active R/3 System. Other users are not affected.
2. Select the SAPDBA export option for constraints, synonyms, sequences, views, and grants (*Export constraints, synonyms, sequences, views, grants*). The following menu options appear:

a	- Incl. owners SYS,SYSTEM
d	- Working directory
s	- Start
q	- Return

See [SAPDBA: Export Menu \[Page 644\]](#)

1. Start the export. SAPDBA starts the ORACLE program *Export*. An import script containing the definitions of the exported objects is generated.

Preliminary Result

SAPDBA generates a restart file **restart.exg** and a log **<timestamp>.exg**. If an error occurs, you must repeat the structure export. You can later start an import of the objects using the scripts stored in **restart.exg**. If an error occurs, you can repeat the import process with the restart function (see [Restarting the Export/Import \[Page 662\]](#)).

If the objects have to be created in the database, use the SAPDBA function *Import structures*. You can also start the script individually (using the SAPDBA function *Additional functions* → *Execute SQL script*). Objects that exist in the database are not overwritten by the import (see [Data Consistency \[Page 638\]](#) and [Structure Import \[Page 660\]](#)).

Script Categories

During an export of the objects listed above, the following scripts are used (when required) or created for a later import or restart:

- **cssvg<DBSID>.sql**
- **restart.exg**



Constraint import command: Constraints are imported using commands with the form ALTER TABLE <owner>.<table name> ADD (<table constraint A>, <table constraint B>, ...

Sequence counter: Sequences are exported with their counters, the last number used in every sequence. Therefore, you should only export and import sequences along with the data that they refer to.

Export Options for Constraints, Synonyms, Sequences, Views, and Grants

Views: The complete specifications of all views are included in an export. The export includes view columns, comments, grants, and inspection options (constraints with category V).

See [Restart File for Export/Import \[Page 663\]](#) and [SAPDBA Scripts and Files \[Page 291\]](#).

Importing Tables with Indexes and Data

Importing Tables with Indexes and Data

This section describes the menu item *Import tables and indexes including data* of the export/import menu.

If you used the SAPDBA function *Export tables and indexes including data*, the scripts necessary for the import have already been created.

Important: Please read the section on [Data Consistency \[Page 638\]](#).

[Procedure \[Page 659\]](#)

Import Options for Tables with Data

Procedure

1. Select the SAPDBA import option for tables with data (*Import tables and indexes including data*).
2. The following menu options appear:

<i>a</i>	- <i>Working directory</i>
<i>b</i>	- <i>Import subdirectory</i>
<i>p</i>	- <i>Show import plan</i>
<i>s</i>	- <i>Start</i>
<i>t</i>	- <i>Start in batch</i>
<i>q</i>	- <i>Return</i>

SAPDBA displays the *Working directory* and a default value for the subdirectory **<timestamp>** that contains the import scripts (*Import subdirectory*). This information is based on the last export performed. Compare this information with that in the appropriate export log **<timestamp>.exd**. Before you start the import, check the import plan (*Show import plan*). If the SAPDBA default directories do not conform to your wishes, you can change the *Working directory* and *Import subdirectory*.

1. Select *Start* to start the import (or *Start in batch* to execute the process in the background).

Result

The tables, table indexes and all constraints and grants are imported with the Oracle program *import* (or SAP load with special SQL scripts) and the import log **<timestamp>.imd** is created

If an error occurs, you can repeat the process with the restart function (see [Restarting the Export/Import \[Page 662\]](#)).



You cannot set any parameters/options with the SAPDBA menu during the import. The settings which were made before the export and stored in the scripts are used

Structure Import

Structure Import

This section describes the menu item **Import structures** of the export/import menu.

If you used the SAPDBA function *Export structures of tables and indexes* or *Export constraints, synonyms, sequences, views, grants*, then the scripts necessary for the import have already been created.

Important: Please read the section on [Data Consistency \[Page 638\]](#).

[Procedure \[Page 661\]](#)

Import Options for Object Structures

Procedure

1. Select the SAPDBA import option for object structures (*Import structures*).
2. The following menu options appear:

<i>a</i>	- <i>Working directory</i>
<i>b</i>	- <i>Import subdirectory</i>
<i>p</i>	- <i>Show import plan</i>
<i>s</i>	- <i>Start</i>
<i>t</i>	- <i>Start in batch</i>
<i>q</i>	- <i>Return</i>

SAPDBA displays the *Working directory* and a default value for the subdirectory **<timestamp>** that contains the import scripts (*Import subdirectory*). This information is based on the last export performed. Compare this information with that in the appropriate export log **<timestamp>.exs** or **<timestamp>.exg**. Before you start the structure import, check the import plan (*Show import plan*). If the SAPDBA default directories do not conform to your wishes, you can change the *Working directory* and *Import subdirectory*.

3. Select *Start* to start the structure import (or *Start in batch* to execute the process in the background). SAPDBA processes the SQL scripts that contain the structure information.

Result

The objects are generated and an import log **<timestamp>.ims** is created.

If an error occurs, you can repeat the process with the restart function (see [Restarting the Export/Import \[Page 662\]](#)).

Restarting the Export/Import

Restarting the Export/Import

This section describes the menu item **Restart export/import** of the export/import menu.

If an error occurred during the export or import, an error message will be displayed. If an error occurred during the first phase of the export/import (script creation), the export must be repeated (a restart is not possible). The restart option only relates to the processing of created scripts (i.e. when an error occurs during the second or third phase). You can then use the restart option to continue a terminated export or import.

To do this, proceed as follows:

- Examine the logs, see [Preparing for the Restart \[Page 630\]](#).
- Use the SAPDBA menu *Restart export/import* to execute the *Restart*. Proceed as described in [Restart Procedure \[Page 631\]](#) (for the restart of an organization).

[Restart File for Export/Import \[Page 663\]](#)



The restart function can also be used in situations other than errors. You can also use it to postpone the start time of the export until you have created the scripts.

Restart File for Export/Import

The name of a restart file follows the naming convention:

<ORACLE_HOME>/sapreorg/<timestamp>/restart.<ext>.

Function ID <ext>: shows the SAPDBA function that the scripts in this file were written for.

Possible Function IDs for an Export/Import

exd	Export/import of tables with indexes and data
exs	Export/import of tables and index definitions (structure export)
exg	Export/import of constraints, synonyms, sequences, views and grants (structure export)

These values are identical with the extension of the corresponding export log

<ORACLE_HOME>/sapreorg/<timestamp>.<ext>.

Logs with the following extensions are also created during the import:

imd	Log for importing tables with indexes and data
ims	Log for structure import (table and index definitions, constraints, synonyms, sequences, views and grants)

The restart file contains the scripts that are processed during an export or an import, e.g. a selection of the following scripts:

```
alt<TSP>.sql, exp<TSP>.sh, exc<TSP>.sh, tab<TSP>.sql, imp<TSP>.sh,
imc<TSP>.sh, ind<TSP>.sql, con<TSP>.sql, grants.sql, cssvg<DBSID>.sql,
imx<TSP>.sh, inx<TSP>.sql, unl<TSP>.sh, ldr<TSP>.sh.
```

For an overview of the scripts used by SAPDBA please see the section [SAPDBA Scripts and Files \[Page 291\]](#). Also note the changes to the scripts when you use the tools SAP unload/SAP load and SAP unload/SQL*Loader (see [SAP Unload/Load, SQL*Loader \[Page 665\]](#)).

Script Indicators

When you start the respective operation (export or import), the scripts in the restart file are processed one by one. The flag by each script tells you whether the script was processed successfully or not. The following flags for the scripts in the restart file are possible:

Y	Script processed successfully
N	Script not yet processed / incorrectly processed. Before the export is started, all scripts either have this indicator or the indicator I.
I	Script is only processed for the import, not during the export. When the import is started for the very first time, these scripts are automatically marked with N by SAPDBA so that they can be included in the process (only applies to a data import, and not to a structure import).

Restart File for Export/Import

S	The data import was already started, but could not be correctly terminated due to an external error (e.g. power failure, or termination by a kill command). Scripts marked with S are automatically processed again when there is a restart.
----------	--

When using SAPunload/load or ORACLE SQL*Loader the same indicators (apart from 'I') also appear in the parameter file `ldr<TSP>.par`.

Scripts marked with N are always processed during an export/import and are assigned the indicator Y after the process has been finished successfully. The import shell scripts are assigned the indicator S during the import. When using the loader program the table that is being loaded is indicated by 'S' in the parameter file `ldr<TSP>.par`.

When restarting the loader (SAP Loader or ORACLE SQL*Loader) only the table indicated with 'S' in the parameter file will be deleted in which the loading process aborted. In contrast to this, all tables belonging to the script indicated with 'S' will be deleted before a restart of the ORACLE import.

During an export, all necessary import scripts are created and included in the corresponding restart file (function ID `exd`, `exs`, or `exg`). However, the respective import scripts are marked with I and are therefore not processed during the export. The indicator is not changed to N until the import process is started, so that SAPDBA includes these scripts in the process.



Never change the sequence of the scripts in the restart file! It defines the flow of the corresponding export/import operations.



In contrast to a reorganization, the command `DROP TABLE` and not the ORACLE command `TRUNCATE` is used to delete the contents of tables which were already incompletely imported (since indexes also have to be deleted here) when a data import is restarted (with ORACLE Import, SAP load or SQL*Loader).

This means that indicator Y has to be reset to N manually in the restart file for scripts `tab<TSP>.sql` before a data import is restarted.

It is no longer necessary to manually delete the incompletely imported tables as in earlier SAPDBA versions (up to and including 3.0C).

SAP unload/load, SQL*Loader

Use

SAPDBA provides the tools SAP unload/load and SAP unload/SQL*Loader (as of Oracle 7.2) for export/import processes, in addition to Oracle export/import (default). SAP load and SQL*Loader use the same data format created by SAP unload.



Before you start the SQL*Loader, the ORACLE script `catldr.sql` must be running in directory `<ORACLE_HOME>/rdbms/admin` under Unix or `<ORACLE_HOME>\rdbms<nr>\admin` under Windows NT.

Advantages of these Programs in Comparison with ORACLE Export/Import:

- Faster export/import of objects.
- The data is automatically unloaded as sorted according to the primary index and correspondingly reloaded. The option `NOSORT` can be used when regenerating the index, so shortening the index generation phase. Instead of using the index chosen automatically by SAPDBA when recovering an index with `NOSORT`, you can choose an index manually for each table (also *NONUNIQUE*). A prerequisite is that all index columns are *NOT NULL* columns. To do this, set the option *Unload/load* in the menu *Transfer method for table data* to `YES`, and the option *Index for sorted unload* to *MANUALLY*. The performance of some accesses to table data can also improve because the data is sorted.
- If the data loading procedure with SAP load or ORACLE SQL*Loader is terminated it is not necessary to repeat the whole import procedure when restarting (as with ORACLE import). You can restart with the table where the termination occurred.
- Simplified reorganization of tables with fields of type `LONG RAW`.

The DBA frequently had to take manual action (define large rollback segments, edit parameter `COMMIT` in script `imp<TSP>.sh`, see for example the SAPDBA online help) in order to reorganize tablespaces containing tables with `LONG RAW` fields (e.g. `PSAPPOOLD`). This is no longer required when using SAP unload/SAP load or SAP unload/SQL*Loader if the menu option `commit=Y` is selected (corresponds to default value).

- SAP unload identifies which tables have to be exported using the entries in script `ldr<TSP>.par`. If you start SAP unload, you can see on the screen how large the individual objects are and how fast they can be exported. SAPDBA determines the average rate of unloading for the objects in the corresponding script `ldr<TSP>.par`. This is a simple way to collect reference values for the times required for the export/import.



You do not have the `Chop` option for SAPDBA unload/SAP load or SQL*Loader, as you do for Export/Import. This means that you can unload/load only those tables whose data after unloading is less than the maximum permitted file size in the

SAP unload/load, SQL*Loader

operating system. More and more operating systems allow files > 2 GB (previously the usual limit).

Scripts/Logs

For a description of the scripts or logs when using Oracle export/import, see [SAPDBA Scripts and Files \[Page 291\]](#) or [Log Files and Their Contents \[Page 297\]](#). The following contains some notes on the differences in the scripts when you use SAP unload/SAP load or SAP unload/SQL*Loader.

Comparison of the Scripts/Logs when Using:

ORACLE export/import	SAP unload/SAP load or SAP unload/SQL*Loader
<code>exp/imp<TSP>.sh</code>	<code>unl/ldr<TSP>.sh</code> (Pseudo shell scripts)
<code>exp<TSP>.par</code>	<code>ldr<TSP>.par</code>
<code>exc/imc<TSP>.sh</code>	Do not exist, as compression and R3chop is not yet supported when using SAP unload
<code>exp<TSP>.dmp</code> (File)	<code>ldr<TSP>.dmp</code> (subdirectory)
<code>exp/imp<TSP>.log</code> (temporary logs, if)	<code>ldr<TSP>.log</code> (parallel processing)
<code>imx<TSP>.sh</code> <code>inx<TSP>.sql</code>	do not exist (no check option has yet been implemented for SAP unload)

The following files are also created in subdirectory `ldr<TSP>.dmp` for every table to be reorganized:

<code><OWNER>_<TABLENAME>.dat</code>	Unload dump file
<code><OWNER>_<TABLENAME>.ctl</code>	Control file for SAP load/SQL*Loader
<code><OWNER>_<TABLENAME>.log</code>	Detailed log of SAP unload/SAP load
<code><OWNER>_<TABLENAME>.ldr</code>	Log of the SQL*Loader
<code><OWNER>_<TABLENAME>.bad</code>	For SQL*Loader errors: data file with the first faulty record
<code><OWNER>_<TABLENAME>.dsc</code>	For SQL*Loader errors: data file with the first rejected record

The programs SAP unload/SAP load and SAP unload/SQL*Loader recognize which objects should be exported with SAP unload and imported with SAP load or SQL*Loader using the information in `ldr<TSP>.par`.

Unload Phase

The process of unloading with SAP unload is selected in the restart file with the pseudo-script `unl<TSP>.sh`. There are no explicit call commands in this script (in contrast to `exp<TSP>.sh`) because the commands for unloading are generated by SAPDBA. The script `unl<TSP>.sh` has indicator N in the corresponding restart file until all the objects in `ldr<TSP>.par` have been successfully processed, that is, all the tables were assigned indicator Y.

- Y/N

SAP unload/load, SQL*Loader

This indicator in **ldr<TSP>.par** in front of the table name specifies whether or not the object was successfully exported with SAP unload.

N	Table was not edited or could not be successfully exported.
Y	Table was successfully exported with SAP unload.

Script **unl<TSP>.sh** is assigned indicator Y and the indicators in **ldr<TSP>.par** are reset to N for all the tables when all the tables have been successfully edited with SAP unload.

Load Phase

The load process with SAP load or SQL*Loader is selected in the restart file with the pseudo-script **ldr<TSP>.sh**. There are no explicit call commands in this script (in contrast to **imp<TSP>.sh**) because the commands for loading are generated by SAPDBA. Script **ldr<TSP>.sh** contains indicator S in the corresponding restart file when the load process is started. The indicator for **ldr<TSP>.sh** is only set to Y once all the objects in **ldr<TSP>.par** have been successfully edited (that is, all the tables have indicator Y).

- Y/N/S

This indicator in **ldr<TSP>.par** in front of the table name specifies whether or not the object has been successfully imported with SAP load/SQL*Loader.

N	Table was not yet edited or could not be successfully loaded.
S	Editing of this table was started. However, the table could not be loaded successfully.
Y	Table was successfully imported with SAP load/SQL*Loader.

Script **ldr<TSP>.sh** is assigned indicator Y in the restart schedule once all the tables have been successfully edited with SAP load/SQL*Loader.

- */-

This indicator in **ldr<TSP>.par** specifies the tool to be used for loading. You can edit this indicator manually before a restart

*	SQL*Loader
-	SAP load

For information about restarting a reorganization, see [Restarting a Reorganization \[Page 628\]](#).

Changing Storage Parameters

Changing Storage Parameters

You can change the storage parameters of a table or an index for all the reorganization types and certain export types supported by SAPDBA. This was already mentioned in the sections [Changing the Extent Size \[Page 620\]](#) and [Export Types \[Page 641\]](#).

Remember that you can also change certain storage parameters online (without doing a reorganization). See [Changing MAXEXTENTS and NEXT \[Page 612\]](#) for more information.

[Options for Changing and Checking the Storage Parameters \[Page 671\]](#)

Overview of the Storage Parameters

During the reorganization or export, SAPDBA displays all values for the storage parameters in a menu, enabling you to see and change the respective current value, the default value and the new value that was manually changed if necessary. The parameters PCTINCREASE, PCTFREE, PCTUSED, FREELISTS and FREELISTS GROUP should only be changed after consulting SAP (e.g. if recommended by the EarlyWatch Service).

The following parameters exist and can be displayed and - if necessary - changed at the start of a reorganization or export using *Change storage manually: yes*:

- **INITIAL:** The size of the single extent into which a table or an index will be imported. The default SAPDBA size (*Compress extents: yes*) is the total of the occupied extents (INITIAL and NEXT) of a table or an index. If you select *Compress extents: no*, the old INITIAL value is used. If you choose the option *Reduce object size: yes*, a smaller value is defaulted if necessary.

Larger value for INITIAL: Specify a larger value for the INITIAL extent of tables/indexes that grow quickly. In this case, the database will not have to immediately assign a new extent.

Smaller value for INITIAL: Specify a smaller value for the INITIAL extent when you know that a table or index does not require the full space allocated.

- **NEXT:** The size of the second and all other extents assigned to a table or an index. Usually (*Use ABAP/4 Dic.for NEXT: yes*), this value is calculated using tables DD09L and TGORA/IGORA. See [Default Value for NEXT \[Page 673\]](#).

You can generally leave the NEXT value set to the SAPDBA default value. If you have made large changes to the INITIAL value, however, you should increase or decrease the value of NEXT accordingly.

- **MINEXTENTS:** Starting number of extents when the table is created (should be set to 1; Exception: The INITIAL extent must be split because of the free space available or the data file size. In this case, SAPDBA changes the value for MINEXTENTS independently).
- **MAXEXTENTS:** Maximum number of extents allowed for the table.
- **PCTINCREASE:** Is always set to zero (0) by SAPDBA (and cannot be changed manually with SAPDBA), to prevent exponential growth of extent sizes in all SAP tables. As a result, NEXT is the size for all new extents assigned after the first one.

You can simulate an asynchronous, linear PCTINCREASE response, however, by using the command option `sapdba -next`. See [Command Option -next \[Page 613\]](#).

- **PCTFREE:** This parameter specifies the percentage of storage in a data block that is held empty for possible changes to existing lines (default value: 10%).
- **PCTUSED:** When a data block is full (except for the space for PCTFREE), no new lines are added to that block. No more lines can be added to that block until the percentage of used storage falls below the value of PCTUSED (default value: 40%).

PCTFREE and PCTUSED are used to optimize use of the storage space in the extent data blocks. Both parameters can be specified for tables. For indexes only PCTFREE can be specified.

Overview of the Storage Parameters

- **FREELISTS:** Number of free lists for an object for managing the data blocks. The default value is 1. The maximum possible value depends on the database block size. Please only change this value after consulting with an expert or when recommended by the EarlyWatch Service.
- **FREELIST_GROUPS:** Defines the number of groups of freelists which is assigned to an object. The default value is 1. This parameter should only be set to a larger value if you are working with the Parallel Server Option of ORACLE.

The optimal value for FREELIST_GROUPS corresponds to the number of database instances for tables. If you are planning to increase the number of database instances, you should use a higher value right from the start. It is only possible to change parameters FREELISTS and FREELIST_GROUPS by reorganizing the tables or indexes. They cannot be changed at a later time with a corresponding ALTER command.



For values larger than 5 ORACLE blocks, the values for INITIAL and NEXT are always rounded off to a multiple of 5 ORACLE blocks. (Example: a NEXT value of 6 blocks was defined. In further operations, the system will use a NEXT value of 10 ORACLE blocks.) This procedure is determined by ORACLE and is taken into consideration by SAPDBA (see for example [Default Value for NEXT \[Page 673\]](#)).

Options for Changing and Checking the Storage Parameters

The following main options are available:

- *Compress extents*: default value *yes*
- *Use ABAP/4 Dic. for NEXT*: default value: *yes*
- *Check space*: default value: *for ALL extents*
- *Reduce object size*: default value *no* (yes for *Reorg list of tables or indexes*)
- *Change storage manually*: default value *no*

You can influence the storage parameters of an object in any of the following ways:

1. Restore the objects without any change (set all options except *Check space* to *no*)
2. Restore the objects by collecting all extents into one extent (Option *Compress extents*: *yes*)
3. Reduce the INITIAL extent determined by the option *Compress extents* so that there is as little freespace in the table as possible after restoration.
4. Manually change current values or SAPDBA defaults (Option *Change storage manually*: *yes*)

[Compress Extents \[Page 672\]](#)

[Default Value for NEXT \[Page 673\]](#)

[Checking Freespace \[Page 675\]](#)

[Reducing the Size of the Table or Index \[Page 676\]](#)

[Changing the Storage Parameters Manually \[Page 677\]](#)

Compress Extents

Compress Extents

Using **Compress extents yes** causes a logical compression of the extents into one extent. Each table and each index is merged in a single extent during the later import.

The size of the combined extent is calculated, on the SAPDBA side, from the sum of the extents allocated in total by the object.

If this sum exceeds the maximum amount of free space available (for reorganization of a table or an index) or the maximum data file size (for a tablespace reorganization), the extent is divided up into single extents so that the free space or data file size is not exceeded. In this case, the values for INITIAL and NEXT are equal; MINEXTENTS is set to the number of extents required.

SAPDBA will issue a recommendation that you reduce the value for NEXT after the reorganization.

Default Value for NEXT

When you carry out a reorganization or an export, SAPDBA recommends extent sizes for the object you are reorganizing or exporting. You can influence the calculation of the default value for extent parameter NEXT with the option **Use ABAP/4 Dic. for NEXT**. The basis for calculating the NEXT parameters are tables DD09L and TGORA/IGORA.

SAPDBA calculates a default value for the NEXT extent size as follows:

- *Use ABAP/4 Dic. for NEXT: yes*

The new value for NEXT is calculated as the maximum of the following two values:

- a) An SAP object is assigned a size category via table DD09L. The NEXT value is determined from table TGORA/IGORA based on that category. (This point does not apply if there is no category assigned in DD09L.)
- b) SAPDBA determines the maximum from the following values:
 - Current NEXT value
 - 10% of the total allocated space

SAPDBA compares this value (the maximum of 1. and 2.) with the values for the extent size in TGORA/IGORA. SAPDBA chooses the next smaller value found in TGORA/IGORA as the new NEXT value (the NEXT value therefore can become somewhat smaller than the current value under certain circumstances). NEXT values which are larger than the largest TGORA/IGORA value are *not* reduced. If TGORA/IGORA does not exist, SAPDBA uses internal default values which correspond to the values in TGORA/IGORA (see "*Values in TGORA/IGORA*").

- *Use ABAP/4 Dic. for NEXT: no*

No comparison with DD09L and TGORA/IGORA is performed; instead the greater of the two following values is used:

- 10% of the total allocated space.
- The current value for NEXT

Values in TGORA/IGORA

Defining the NEXT values to correspond to the permanently assigned values in the TGORA/IGORA tables results in less wastage in the tablespace. If the extent values are defined using a size category assigned in the ABAP/4 Dictionary (see [Technical Configuration \[Page 25\]](#)), incorrectly defined values can be corrected in the database without great difficulty.

NEXT values in TGORA/IGORA

Size category	NEXT value for table (in KB)	NEXT value for indexes (in KB)
0	16	16
1	160	80
2	640	160
3	2560	640
4	10240	2560

Default Value for NEXT

5	20480	5120
6	40960	10240
7	81920	20480
8	163840	40960
9	327680	81920
10	655360	163840
11	1310720	327680
12	2621440	655360
13	5242880	1310720
14	10485760	2621440

You can change this distribution manually, but you should note the following when doing so:

- Tables of size category 14 can reach a size of about 1000 GB and indexes a size of 250 GB for 100 assigned extents with the current default values. You should therefore reconsider whether larger values are really required.
- The extent values (except for category 0) are rounded to a multiple of `5*db_block_size` (above table: to a multiple of 40KB) in order to avoid additional ORACLE rounding. If you change values or insert new ones, it is best if you also specify only the corresponding multiple.

Checking Freespace

During each reorganization, SAPDBA checks whether all reorganized objects can be recovered with their data in the appropriate tablespaces. Besides the size of *initial* and *next*, the number of extents expected is also taken into account (according to *minextents* and the amount of data to be imported). This corresponds to the default of the option *Check space: for all extents*.

In certain exceptional cases (particularly when the option *Compress Extents: no* and/or *Reduce object size: no* is set), the memory space check may be too restrictive. SAPDBA states that there is not enough freespace, even though there is.

In such cases, you can set the option *Check space: only first minextents*. This means that only the *initial* extent or the extents according to *minextents* are included in the storage space check. Other *next* extents are not taken into consideration in the space check.



If additional *next* extents are generated during the import/load and the import/load terminates due to lack of space, the import/load can be repeated with restart after the tablespace has been increased.

Reducing the Size of the Table or Index

Reducing the Size of the Table or Index

You may want to reduce the size of a table/index (or the size of all objects in a tablespace) during a reorganization or for an export/import if a relatively large amount of memory space is required and occupied (allocated), but is not used by the relevant objects.

Choose **Reduce object size: yes**, if you want SAPDBA to automatically try to reduce the size of the objects (allocated storage space) during the reorganization or for an export/import. To determine the actual storage space occupied, SAPDBA uses the ORACLE command `ANALYZE TABLE <Name> ESTIMATE STATISTICS` or `ANALYZE INDEX <Name> VALIDATE STRUCTURE`. Using these values and the current storage parameters, a new value is determined for the space to be allocated. If this value is smaller than the storage space allocated before the reorganization or export/import, the size of the object is reduced to this value.

Changing the Storage Parameters Manually

You can also manually change all the SAPDBA default values. To do this, select **Change storage manually: yes**.

When you select all the objects to be reorganized or exported/imported, a screen mask appears in which you can change the storage parameters as required. SAPDBA will then use these changed values for the corresponding process.

- From the object list, you can select any objects for processing (*next object|table|index*, i.e. all objects or only tables or only indexes).
- Tables and indexes can be displayed selected in any order using any number of letters in their name (*name <name>|<part_of_name%>*, where % stands for the LIKE function).
- You can restrict the object type (*type <all>|<table>|<index>*).

You can display and/or manually change the storage parameters of the objects you select. If the name selection leaves out tables or indexes, these will automatically be assigned the SAPDBA default values (in accordance with the options selected for *Compress Extents*, *Reduce object size* or *Use ABAP/4 Dic.for NEXT*).

To exit the display, select *continue export/reorganization* (the configured values are used in further operations) or *quit* (the process is terminated).

[Overview of the Storage Parameters \[Page 669\]](#)

Tablespace Administration

[Tablespace Administration: Overview \[Page 679\]](#)

[Backup Recommended \[Page 680\]](#)

[SAPDBA: Tablespace Administration \[Page 681\]](#)

[Checking Tablespaces \[Page 682\]](#)

[Extending a Tablespace \[Page 686\]](#)

[Creating a Tablespace \[Page 694\]](#)

Tablespace Administration: Overview

The SAPDBA function **Tablespace administration** (see *SAPDBA: Tablespace administration*) can be used to perform the following tasks:

- Checking for space problems in tablespaces.
See [Checking Tablespaces \[Page 682\]](#).
- Adding a data file to a tablespace.
If the database system cannot assign any more extents in a tablespace (ORACLE error 1650, 1652-1655), you have to extend - add a new file to - the tablespace. By extending the tablespace, additional freespace exists for further tasks.
See [Extending a Tablespace \[Page 686\]](#).
- Creating a new tablespace
See [Creating a Tablespace \[Page 694\]](#).

Backup Recommended

Backup Recommended

Extending or creating a tablespace changes the structure of the database. The database can be restored more quickly when you back up the new data file/tablespace and the control file.

Backing Up a Newly Created Tablespace

Back up the newly created tablespace with the SAP utility program BRBACKUP.

Backup after Tablespace Extension

A backup after a tablespace extension is not required when you use the functions in the menu *Restore/Recovery*, since SAPDBA also creates new files, including the tablespace extension. You are given information about the last successful complete backup and about the necessary time period for any necessary recovery. You should always save the tablespace after an extension (for example, use the SAP utility program BRBACKUP) so that all the SAPDBA recovery functions are available (menu *Check (and repair) database* and *Restore/Recovery*). For this reason, after a tablespace extension, SAPDBA automatically branches to the *Backup database* menu, to enable you to start the appropriate backup immediately. As a result, you have a valid backup of the new control file and of the data files of the tablespace.

Independently of the above, SAPDBA backs up the old and new control files that do not yet exist in an old database backup, to the work directory `<ORACLE_HOME>/sapreorg/<timestamp>`. They are given the names `cntrl<SAPSID>.old` and `cntrl<SAPSID>.new`, respectively.

If you extend several tablespaces in a row, you should not back them up individually (this means that you have to exit the Backup menu), but instead carry out a complete backup of the database, once you are finished with all the extensions (an online backup is sufficient, however, an offline backup is better).

SAPDBA: Tablespace Administration

	Check/alter tablespace
<i>a</i>	- <i>Tablespace</i>
<i>b</i>	- <i>Log checks</i>
<i>c</i>	- <i>Freespace and fragmentation of all tablespaces</i>
<i>d</i>	- <i>Check free space of objects in all tablespaces</i>
<i>e</i>	- <i>Check free space of objects in tablespace</i>
<i>f</i>	- <i>Alter tablespace Add Datafile</i>
<i>g</i>	- <i>Create tablespace</i>
<i>h</i>	- <i>Display all tablespaces and datafiles</i>
<i>i</i>	- <i>Display raw device drivers in /dev</i>
<i>q</i>	- <i>Return</i>

SAPDBA records the performed checks in log **<timestamp>.ext**, when you have set the *Log checks* option to yes. See [Setting the Log Checks Option \[Page 50\]](#).

The extension of a tablespace or the creation of a new one is always logged.

Checking Tablespaces

Checking Tablespaces

You should check whether a tablespace needs to be extended, and distinguish between two cases:

- **The tablespace is highly fragmented.**

If the tablespace has sufficient free storage space, it is probably a fragmentation of free storage space which causes the storage problems. This will occur only if DROP operations are performed frequently. Database objects that are frequently deleted (dropped) and created anew should be separated.

You can solve the storage problems by reorganizing the fragmented tablespace or the fragmented tables and indexes which it contains. An extension of the tablespace would then not be necessary.

A reorganization is time-consuming and complicated, however. SAP therefore recommends that you extend the tablespace rather than reorganizing it. You should reorganize this tablespace only if one or more of the following conditions apply:

- The maximum number of data files for your system has been reached
- An extremely large number of extents has been assigned to very many objects in the corresponding tablespace
- Accessing the data in the tablespace causes performance problems

See [Limitations of the Database System \[Page 579\]](#).

- **The tablespace has too little free storage space.**

A full tablespace must always be extended.

If an extension of a tablespace causes your database to approach the limit value determined by the operating system for the number of files, it can be useful to reorganize this tablespace after an extension or to extend the tablespace during the reorganization. However, a reorganization is not very likely necessary because the limit value for the data files is very high (at least 254 files are available).

[Freespace Statistics \[Page 683\]](#)

[Check Functions \[Page 684\]](#)

[Files/Raw Devices of a Tablespace \[Page 685\]](#)

Freespace Statistics

Select *Freespace and fragmentation of all tablespaces*.

With these statistics you can see whether it is necessary to extend a tablespace or whether a reorganization should be performed instead or in addition. Check the following information:

- **Allocated** and **% Alloc.:** Reserved storage space in kilobytes and as a percentage of the tablespace size.

When the utilization is more than 90 percent, you should extend the tablespace by adding a new data file.

Please do not try to solve this problem by reorganizing the database. A reorganization defragments the complete storage space. This means that the fragmented used storage space and fragmented free storage space are put together, i.e., the storage space is reorganized, but no additional space is gained by this. In fact, additional space may even be required. Therefore, always extend a tablespace where there is little space left.

- **Free areas** (number of freespace areas) and **Largest** (size of the largest freespace area).

If the percentage of the storage space used is relatively low, you should check these columns to see whether the storage problems in a tablespace are due to the fragmentation of free storage space.

If the number of fragments is high and the largest fragment is small (smaller than 1 MB), the cause of the tablespace overflow is probably the fragmentation. It may be useful to reorganize the fragmented tablespace (or highly fragmented tables and indexes in it) instead of extending it. However, please note: Since enough data files are available, you can solve this problem by adding a new file. If possible, avoid reorganization because it is more complicated and time-consuming.

Check Functions

Check Functions

You can check whether (and which) tablespaces would overflow if an additional n extents were assigned.

- *Check free space for objects in all tablespaces*
Overflow check for all tablespaces. Displays a list of the tables for which assigning an additional n extents would cause the corresponding tablespace to overflow.
- *Check free space for objects in tablespace*
Overflow check for one tablespace. Select *Enter tablespace name* and enter the name of the tablespace. Displays a list of the tables for which assigning an additional n extents would cause the corresponding tablespace to overflow.

Which Table or Index Caused the Overflow?

Occurrences of the ORACLE error (tablespace overflow) are recorded in the R/3 System log (along with the table/index that was accessed).

You can then determine which tablespace the table/index belongs to - with SAPDBA, for example - using the function *Check free space for objects in all tablespaces* with test value 1.

Tablespace Overflow Threshold Values

- **Overflow with next extent**
Extend the tablespace as soon as possible. Exception: the tablespaces PSAPPROTD/I contain SAP logs (from background processes and spool requests). Delete old log files and requests instead of extending these tablespaces.
- **Overflow with the next two extents**
Monitor the tablespace.
- **No overflow with the next two extents**
The tablespace currently has sufficient free storage space. No intervention is necessary, provided the tablespace continues to be used normally. Important: if you change the storage parameters or perform operations in the R/3 System that will fill this tablespace more quickly, you should monitor it as well.

Database Monitor

The database performance monitor of the R/3 System also provides information on how urgent a tablespace extension is.

For more information, refer to the online help of the *Computing Center Management System*.

Files/Raw Devices of a Tablespace

The objects of a tablespace can be stored in files or in raw devices. SAPDBA can display this assignment for you:

- *Display all tablespaces and datafiles*
Display all tablespaces and datafiles displays the assignment between tablespace - data files or tablespace - raw devices and the size of the files.
- *Display all character devices in...*
Display all character devices in /dev (depending on the operating system) displays an overview of all the raw devices and their size. If data of your ORACLE database is in a raw device, the link to this raw device is also displayed.

See also [SAPDBA Support for Raw Device Configuration \[Page 767\]](#).

Extending a Tablespace

Extending a Tablespace

The R/3 System can be either active or closed when you extend a tablespace.

Do **not** shut down the database system if you want to add a data file to a tablespace.

1. Select *Tablespace* and enter the name of the tablespace you want to extend. Select *Alter tablespace <tablespace name> Add data file*. The following menu options are displayed, depending on whether you want to add a file (*in file system*) or a raw device (*on raw device*):

	File system	Raw devices
a -	File system or raw device	File system or raw device New data file name/Required raw device name
b -	File size	Enter raw device name
c -	Select path	Select raw device
d -	Alter suggested path	File size on raw device
e -	Display current files	Display current files
s -	Start (Add datafile)	Start (Add datafile)
q -	Return	Return

File system or raw device: SAPDBA displays whether you are working in a file system (*file system*) or on raw devices (*raw device*). At this time you can choose again between working in the file system or on raw devices. If you are working on raw devices, the ORACLE name of the raw device and the required raw device ID are displayed. See also [SAPDBA Support for Raw Device Configuration \[Page 767\]](#).

2. You can change the default values for the size and path of the new data file.

File system:

- **File size:** SAPDBA calculates the size of the new data file from the previous size of the tablespaces using one of the following percentages:
 - Percentage from the SAPDBA profile (`init<DBSID>.dba`) if parameter `tspadd_tspname` is set for this tablespace.
 - Default value: 30 percent of the tablespace size when no entry exists for this tablespace in the SAPDBA profile.

See [tspadd_tspname \[Page 287\]](#) and [Specifying the File Size \[Page 689\]](#).

- **Select path:** SAPDBA displays a list of all sapdata directories and highlights the directories that are already used by the tablespace and its corresponding D/I tablespace (`used (<n> x)`, `<n>` number of files).

You can decide which of the displayed sapdata directories you want to select. If you specify a different directory and this results in the files being moved to a different disk,

Extending a Tablespace

this may (when the data and index tablespace are on the same disk) cause performance losses due to competing I/O accesses.

See [New Path \[Page 692\]](#).

- **Alter suggested path:** SAPDBA suggests a path for the data files according to the SAP naming conventions or displays the path which you selected with *Select path*. Only the first part of the complete target directory is suggested (<ORACLE_HOME>/sapdata<n>). The other part of the directory and the file name itself are defined according to the SAP naming conventions.

See [Naming Convention \[Page 691\]](#).

You can define another file name (see [New Path \[Page 692\]](#)). In this case SAPDBA checks whether there is sufficient storage space available in the file system which was suggested to store the new data file.

Raw Device:

- **Enter raw device name:** Enter the name of the required raw device. You can enter the complete path here. If you do not do so, the path defined under *select raw device* will be used.

SAPDBA checks whether:

- the raw device exists
- ora<sapsid> is the owner
- an ORACLE file of <DB-name> resides there
- a file system (mounted) resides there.

SAPDBA does not check whether the raw device is used by another application and whether the disk is partitioned and the partitions are reserved.

- **Select raw device in path:** SAPDBA displays a list of the raw devices which can be used to extend the tablespace (owner: ora<sapsid>, CAUTION: It is up to the database administrator to check that there is no data from other applications in the selected raw device!).

You can decide which of the displayed raw devices you would like to select. SAPDBA informs you if the naming convention does not correspond to the SAP norm. If necessary, change the raw device ID as user `root`.

- **File size on raw device:** You can reduce the size after selecting a raw device, but the space which remains will not be used.

Display current files: SAPDBA displays the data files of the tablespaces and their sizes.

Depending on the operating system, the database may be limited to a certain number of data files (usually at least 254 files with maximum 2GB per file). If an extension moves you closer to this limit, you should reorganize the tablespaces that contain a large number of data files. This will occur only rarely, however, since the number of available data files (254) is almost always large enough. Therefore, do not reorganize tablespaces of the database until you actually reach this limit.

See [Limitations of the Database System \[Page 579\]](#).

3. Select *Start (Add datafile)* to accept the file specification.
4. At this point, SAPDBA checks the block sizes. The following values are displayed:

Extending a Tablespace

filesystem block	size:	<size>
filesystem fragment	size:	<size>
Oracle block	size:	<size>

If the sizes do not agree, an appropriate warning message appears and SAPDBA will ask whether you want to continue.

Successful tablespace extension: When SAPDBA has created the new data file, it automatically branches to the *Backup database* menu, to enable you to back up the extended tablespace.

See [Backup Recommended \[Page 680\]](#).

Failed tablespace extension: If the data file could not be added, SAPDBA deletes the corresponding directory or symbolic link and cancels the tablespace extension. Your system and database are unchanged.

See [SAPDBA Troubleshooting \[Page 693\]](#).

Use the ORACLE ALERT and trace files to analyze and correct the problem, then repeat the tablespace extension procedure.

5. SAPDBA generates a log: <timestamp>.ext. This log can be used to verify that the tablespace extension was successful.

[Specifying the File Size \[Page 689\]](#)

[Naming Convention \[Page 691\]](#)

[New Path \[Page 692\]](#)

[SAPDBA Troubleshooting \[Page 693\]](#)

Specifying the File Size

SAP has default percentages for the growth in tablespaces in the SAPDBA profile `init<DBSID>.dba` (See [tspadd tspname \[Page 287\]](#)). While these percentages are based on past experience, growth in tablespaces can vary widely, and you may want to change the default settings for your system. You can therefore enter a different file size than the default in the SAPDBA profile to reflect the actual growth you expect in the tablespace.

Guideline

You should size a new data file for that file to accommodate the growth of the tablespace for a long period. For a critical, rapidly growing tablespace, the file should be as large as possible. You also need to take the available storage space and the storage requirements of other tablespaces into account.

If you add many small files to a tablespace, you may need to reorganize the tablespace. This can be extremely time-consuming. Please also consider that you will have to shut down the R/3 System in order to reorganize the database. Reorganizing tablespaces with data files is the most complicated type of reorganization and should therefore be done as infrequently as possible. To avoid such situations, you should always add a sufficiently large file when extending a tablespace.

To minimize storage problems and in order not to reach the maximum number of files for the database system too quickly, SAP recommends that you do not set file sizes to values lower than those recommended below.

The following table contains tablespaces which should be monitored during productive operation:

Minimum File Sizes for Critical Tablespaces: Production

Tablespace	Minimum extension size (in MB) for data tablespace (D)	Minimum extension size (in MB) for index tablespace (I)
PSAPCLU	300	50
PSAPBTAB	300	200
PSAPSTAB	100	50
PSAPPOOL	100	50
PSAPPROT	100	50

Before transferring transaction data or starting production use of the system, each of these tablespaces must be extended with at least one file that is larger than the proposed minimum extension size.

If you work with client computers based on different hardware platforms, then you should also monitor PSAPLOADD/I or PSAPEL300D/I carefully.



Changes in these default values are entirely possible, especially - for example - when the EarlyWatch Service has determined other values to be more effective for your system.

Specifying the File Size

During an SAP update or upgrade, in addition to the tablespaces listed above, you should check and - when necessary - extend the tablespaces listed below.

Minimum Extension Sizes for Critical Tablespaces: Update/Upgrade

Tablespace	Minimum extension size (in MB) for data tablespace (D)	Minimum extension size (in MB) for index tablespace (I)
PSAPDOCU	100	50
PSAPDDIC	50	20
PSAPSOURCE or PSAPES300	50	20

If possible, store I/O-intensive tablespaces each on separate hard disks.

If possible, select different mount points for a data tablespace and its index tablespace when you set up the database.



The SYSTEM tablespace can also be extended by SAPDBA. However, the size of the SYSTEM tablespace cannot be reduced after you have extended it.

A tablespace extension to PSAPTEMP can be cancelled by SAPDBA with the corresponding reorganization means. However, this is only possible if there are no temporary segments in PSAPTEMP (that is, you have to log off all the other users).

You can undo a tablespace extension to PSAPROLL only by means of irregular procedures using ORACLE functions. Excellent knowledge of ORACLE procedures is required!

Naming Convention

File Systems

As a default, SAPDBA attempts to add a new file to the standard SAP directory in which the most recent data file of the tablespace was stored. The most recent file is the one with the highest file number.

- **Directory:** <ORACLE_HOME>/sapdata<n>/<tablespace name>_<file number>
- **File name:** <tablespace name>.data<file number>

Also see [Environment Variables \[Page 45\]](#).

<n> is the sequentially assigned number of the SAP directory in which the data file will be stored.

<file number> is the sequentially assigned number of the data file in the tablespace. The same number also appears in the subdirectory which is created for each new file in a tablespace.



You are adding a third data file to tablespace PSAPDOCUD in the R/3 System database C11. PSAPDOCUD consists of the following files:

```
/oracle/C11/sapdata1/docud_1/docud.data1  
/oracle/C11/sapdata2/docud_2/docud.data2
```

SAPDBA suggests `/oracle/C11/sapdata2` for the new data file, and creates a subdirectory `docud_3`, in which the file is created with name `docud.data3`.

Thus, the complete new file name is:

```
/oracle/C11/sapdata2/docud_3/docud.data3.
```

Raw Devices

See [SAPDBA Support for Raw Device Configuration \[Page 767\]](#).

New Path

New Path

You can enter a different directory path for the new data file if the standard directory does not have enough storage space for the file. You can create the file in a file system that has sufficient space.

To operate your database in a safer, more efficient manner, SAP recommends specifying a new path in certain cases. These files can be on the same hard disk as the redo log files if the redo log files are mirrored. However, SAP recommends for performance reasons that data files and redo log files always be stored on different hard disks.

See [Organizing Disk Storage \[Page 38\]](#).

SAPDBA does not allow you to change the name of the data file. You can, however, change the directory path as follows:

- **Other sapdata directory:** You can create the file in a different standard directory, `sapdata<n>`. The directory must already exist and should be stored in a file system others than the one suggested by SAPDBA. Use *Select path* to select the directory.

SAPDBA creates the new data file in the directory that you entered. It does not create a symbolic link from the directory it recommended.

Please note that specifying a different directory may result in competing I/O accesses, which can diminish the performance of your system.

- **Non-standard path:** You can create the file under a path which does not correspond to the SAP standard.

Example: You have set up and mounted a disk for data files. New files must therefore be added under path `/usr/newdisk` and not under the standard path.

SAPDBA performs the following tasks:

- It creates a subdirectory `<tablespace name>_<file number>` in the new directory `/usr/newdisk`, which you specified.
- It creates a symbolic link between the SAPDBA default directory (e.g. `/oracle/C11/sapdata2/<tablespace name>_<file number>`) and the new subdirectory `/usr/newdisk`.
- It creates a file `<tablespace name>.data<file number>` in the new subdirectory.

The symbolic link is transparent in the Data Dictionary of the database. The Data Dictionary recognizes only that the file exists in the standard SAP directory.

See [Naming Convention \[Page 691\]](#).

- **Raw device:** SAPDBA behaves as described in the section Non-standard path. The symbolic link is set from `<ORACLE_HOME>/sapraw` to the corresponding raw device. The file `<tablespace-name>.data<file number>` does not have to be created. See also [SAPDBA Support for Raw Device Configuration \[Page 767\]](#).

SAPDBA Troubleshooting

Before an additional file is added, SAPDBA carries out the following checks (if file systems are used):

- **Block size:** The following block sizes must be identical:

	Block size for
filesystem block size	the file system
filesystem fragment size	the file system fragments
ORACLE block size	the database system

SAPDBA verifies that the above values are identical. If not, it issues a warning message and suggests selecting a new file system. You can ignore the warning and continue, but this is not recommended, as system performance will be affected when the block sizes of the file system and database are different.

Please also note the following:

- The ORACLE database block size is always 8 KB in the R/3 System.
- SAPDBA does not perform the check under the AIX operating system, as the block sizes are always different:

ORACLE block size	4 KB
filesystem block size	8 KB

- For OSF1, the value for filesystem fragment size is always 1 KB, and is therefore not checked by SAPDBA.
- **Disk storage space:** SAPDBA checks that there is sufficient storage in the file system proposed for the new file. If not, SAPDBA does not add the file.

If these checks were successful, SAPDBA creates a directory or a symbolic link for the data file. The control file (old and new) is backed up under the directory (directory name: <timestamp>) belonging to the log file (<timestamp>.ext). SAPDBA then creates the data file with ALTER TABLESPACE ADD DATAFILE. If any of these operations fails, SAPDBA is terminated immediately.

Creating a Tablespace

Creating a Tablespace

The function *Create tablespace* is used to create new tablespaces. SAPDBA only supports the creation of tablespaces that correspond to the naming conventions for SAP tablespaces (PSAP<name>D/I). PSAPROLL and PSAPTEMP can also be created with SAPDBA, but not tablespace SYSTEM.

When you create a tablespace, SAPDBA verifies that the corresponding data and index tablespaces already exist. The results of the check are displayed, but any missing tablespaces are not created automatically. You must ensure that both the data tablespace and the index tablespace exist.

Enter the name of the tablespace. Then select *Create tablespace <tablespace name>*. The following menu options appear, depending on whether you want to assign a file (*in file system*) or a raw device (*on raw device*) to the tablespace.

Create new tablespace <tablespace name>

Corresponding <index/data> tablespace <tablespace name> exists/does not exist

	File system	Raw Devices
a -	<i>File system or raw device</i>	<i>File system or raw device New data file name/Required raw device name</i>
b -	<i>File size</i>	<i>Enter raw device name</i>
c -	<i>Select sapdata path</i>	<i>Select raw device in path</i>
d -	<i>Alter suggested path</i>	<i>File size on raw device</i>
e -	<i>Default storage parameters</i>	
s -	<i>Start (create tablespace)</i>	<i>Start (create tablespace)</i>
q -	<i>Return</i>	<i>Return</i>

File system or raw device: SAPDBA displays whether you are working in a file system (*file system*) or on raw devices (*raw device*). You can choose again here between working in the file system or on raw devices. If you are working on raw devices, the ORACLE name of the raw device and the required raw device ID are displayed. See also [Raw Devices and SAPDBA \[Page 767\]](#).

File system:

- **File size:** Enter the desired size of the data file of the tablespace. If you do not specify a value, the double block size (generally 16 KB) will be selected. Do not use the default, but instead enter a file size appropriate for your requirements.
- **Select sapdata path:** SAPDBA displays a list of all sapdata directories and highlights the directories that are already used by the tablespace and its corresponding D/I tablespace (*used (<n> x)*, where <n> is the number of files).

You can decide which of the displayed sapdata directories you want to create the new tablespace in. Make sure that a data tablespace and its corresponding index tablespace lie in different sapdata directories.

Always use *Select path* to define the name of the new data file, as this will provide you with additional important information: which directories are available, how much free

Creating a Tablespace

storage space is available, which directories contain the files of the corresponding tablespaces (if they exist).

- **Alter suggested path:** SAPDBA suggests a path according to the SAP naming conventions for data files (the suggestion is a path which does not contain a file of the corresponding tablespace) or displays the one you selected with *Select path*. Only the first part of the complete target directory is suggested (<ORACLE_HOME>/sapdata<n>). The other part of the directory and the file name itself are defined according to the SAP naming conventions.

See [Naming Convention \[Page 691\]](#).

You can also specify a different file name. In this case, SAPDBA also checks whether there is sufficient storage space in the file system which was suggested to store the new data file.

- **Default storage parameters:** The storage parameters for the tablespace are displayed here: INITIAL 16 KB, NEXT 40 KB, MINEXTENTS 1, MAXEXTENTS 300, PCTINCREASE 0. You can change these values. If a table or an index is created in this new tablespace later on without explicitly specifying the storage parameters, the values set for this object with *Alter default storage* are assumed.

Raw Devices:

- **Enter raw device name:** Define the name of the required raw device. You can enter the complete path here; if you do not do so, the path definition under *select raw device* will be automatically used.

SAPDBA checks whether:

- the raw device exists
- ora<sapsid> is the owner
- an ORACLE file of <DB-name> resides there
- a file system (mounted) resides there.

SAPDBA does not check whether the raw device is used by another application or whether the disk is partitioned and the partitions are reserved.

- **Select raw device in path:** SAPDBA displays a list of the raw devices which can be used to create the tablespace (owner: ora<sapsid>. Caution: It is the responsibility of the database administrator to check that no data from other applications is residing on this raw device.)

You can decide which of the displayed raw devices you would like to select. SAPDBA informs you if the naming convention does not correspond to the SAP norm. If necessary, change the raw device ID as user *root*. SAPDBA displays the size of the selected raw device, but it cannot be changed.

- **File size on raw device:** You can change the size of a raw device that you have selected. However, you can no longer use the remaining space.

Start (create tablespace): Creates the tablespace. SAPDBA generates a log <timestamp>.ext.

SAP lets you create more of your own tablespaces (PSAPUSER<n>D/I, n<=9) in addition to the delivered tablespaces (see [Tablespaces and SAP Naming Conventions \[Page 41\]](#)). SAPDBA

Creating a Tablespace

updates the database tables TAORA and IAORA to enable you to create tables and indexes for the new tablespaces you defined from within the R/3 System.

SAPDBA Database Recovery

Use

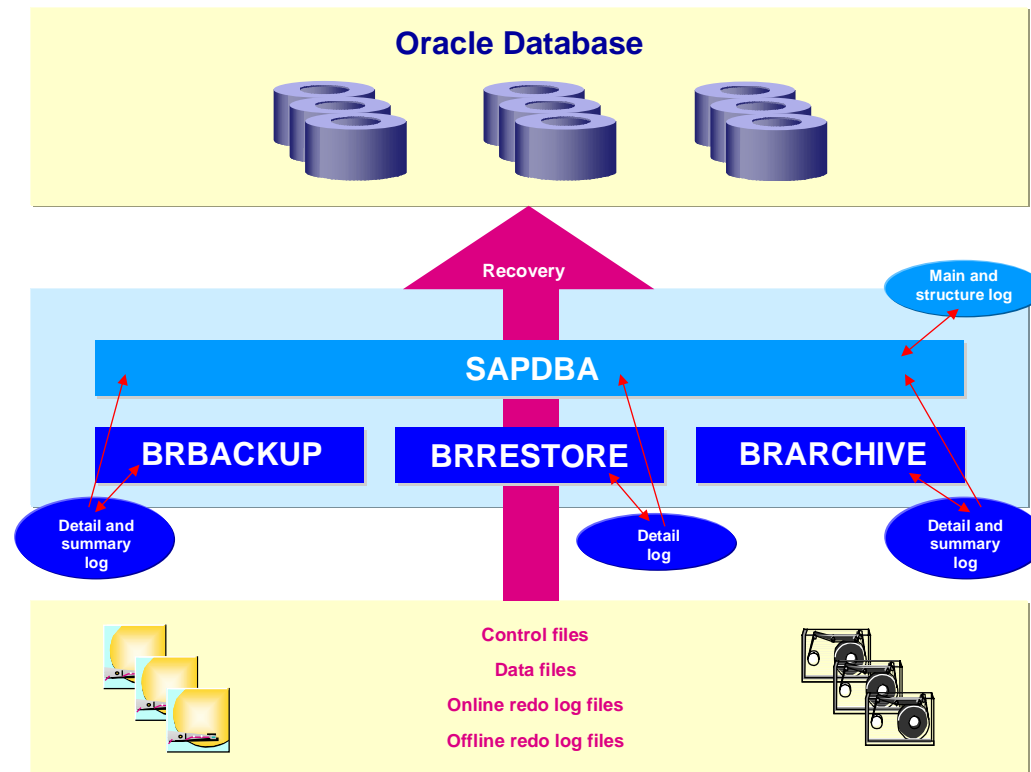
SAPDBA supports the recovery of the database after:

- *physical failures*, caused by hardware errors (for example, a hard disk fault)
- *design failures*, caused by software errors (for example, faulty software)
- *failures caused by user errors*

The SAPDBA restore and recovery functions are based on use of the backup tools BRBACKUP, BRARCHIVE and BRRESTORE. In a recovery, the backups created with BRBACKUP/BRARCHIVE are restored with BRRESTORE. The operating system logs of BRBACKUP and BRARCHIVE are highly important for the recovery since they contain all the information about the backups (such as directories, volumes and timestamps). You may need to restore these logs from existing backups.



The recovery with SAPDBA function is not based on BRBACKUP or BRARCHIVE logs in database tables, since these tables are not available when you recover the database.



You can recover the database from the SAPDBA *Restore/Recovery* menu.

SAPDBA Database Recovery



Only experienced database administrators working in expert mode ([SAPDBA: Expert Mode \[Page 60\]](#)) can use the SAPDBA *Restore/Recovery* function.

Prerequisites

The following prerequisites must be met for a successful recovery of the database:

- The BRBACKUP and BRARCHIVE logs must be available. SAPDBA can use the logs to find out where the backups of the data files and the offline redo log files are located.
- The backups of the missing or faulty data files made with BRBACKUP or BRARCHIVE must be available.



Other prerequisites may also need to be met, depending on the particular context of the recovery.

Features

SAPDBA offers various functions for recovering the database that are adjusted to particular situations:

- [Partial Restore and Complete Recovery \[Page 699\]](#)
- [Full Restore and Recovery \[Page 714\]](#)
- [Reset Database \[Page 721\]](#)
- [Restore One Tablespace \[Page 722\]](#)
- [Restore Individual File\(s\) \[Page 723\]](#)

	<i>Restore/Recovery</i>
<i>a</i>	- <i>Partial restore and complete recovery</i>
<i>b</i>	- <i>Full restore and recovery</i>
<i>c</i>	- <i>Reset database</i>
<i>d</i>	- <i>Restore one tablespace</i>
<i>e</i>	- <i>Restore individual file(s)</i>
<i>q</i>	- <i>Return</i>



Familiarize yourself with the differences between these functions before you start the recovery of the database. If you choose a function that does not suit your situation, you may not be able to recover the database, or you may lengthen its downtime.

SAPDBA: Check (and Repair) Database

Different types of errors can interfere with the operation of an ORACLE database recovery. See [Types of Database Errors \[Page 728\]](#).

The task of the SAPDBA recovery function is to remedy certain media and user errors. When such an error occurs, it usually involves the loss of database files, which can contain many different types of objects: ORACLE Dictionary segments, temporary segments, rollback segments, or user segments (tables and indexes).

SAPDBA supports restoring the database after the loss of the following files:

- Data files of the SAP tablespaces (PSAP<name>D/I)
- Files of the system tablespace (SYSTEM)
- Files of the rollback tablespace (PSAPROLL)
- Files of the temporary tablespace (PSAPTEMP)

The menu item *Check (and repair) database* only enables recovery of the database up to the current time. If you want to restore an older version of the database or perform a point in time recovery, please use the *Restore/Recovery* function (see [SAPDBA: Restore/Recovery \[Page 714\]](#)).

Also read the corresponding ORACLE documentation.

[Requirements \[Ext.\]](#)

[Recovery Options \[Ext.\]](#)

[Automatic Recovery \[Page 700\]](#)

[Recovery Procedure \[Ext.\]](#)

[Checking the Database Files \[Page 701\]](#)

[Finding the Backup Files \[Page 703\]](#)

[Restoring the Backup Files \[Page 705\]](#)

[Changing the Restore Parameters \[Ext.\]](#)

[Finding the Archived Redo Logs \[Page 710\]](#)

[Restoring Archived Redo Logs \[Page 712\]](#)

[Database Recovery \[Page 713\]](#)

Recovering the Database Automatically

Recovering the Database Automatically

Use

After you have lost a data file or discovered a faulty data files, first try and recover the database with *Automatic recovery*. SAPDBA automatically selects the last valid backup of the relevant data files and the appropriate log sequence numbers.

Procedure

Choose *Automatic recovery*, to run the function *Partial restore and complete recovery*. After each step you are asked whether you want to continue with the recovery. If you use backups to tape for the recovery, SAPDBA prompts you to mount the tapes with the correct labels.

If the automatic recovery terminates, the SAPDBA menu *Restore/Recovery* appears again. The status display tells you which steps ran successfully (status *finished*). You must perform any unsuccessful steps (status *not finished* manually after you have solved the problem that led to the termination.



The function *Automatic recovery* does not support a recovery of the database with incremental backups.

Checking the Database Files

Use

Choose **Check database** to check for critical errors in the database that prevent further operations.

Procedure

The database is checked by analyzing the appropriate V\$ tables. Because some of these tables are only updated when the database is restarted, SAPDBA differentiates between:

- **Online quick check:** This check is performed with the database open. Any attempt to recover the database with SAPDBA is rejected, however, because the information obtained from this check is not necessarily current. As a result, if you have only performed the online short check, the *Check database* function is assigned the status *not finished*.

In addition you can use the SAPDBA menu option *Archive mode* → *Show complete archive history* to determine initially which redo log files are necessary for a recovery, starting from the last backup. You can perform the online quick check at any time, in order to check the status of the control files, data files, and the online redo log files.

- **Offline safe check:** If the database has not been closed yet, SAPDBA closes the database (with `shutdown abort`, if necessary). The database is then mounted. This guarantees that the information in the V\$ tables is up-to-date. When the offline safe check is finished, and as long as it resulted in no errors, the database is opened again.

SAPDBA checks the condition of:

- Data files (existence; Status: OK, RECOVERY; Status: ONLINE, OFFLINE)
- Tablespaces (Status: ONLINE, OFFLINE; Status: BEGIN BACKUP, END BACKUP)
- Control files (existence, validity if necessary)
- Online redo log files (Existence; Status: OK, STALE, DAMAGED)



These checks do not find out, for example, whether table indexes are missing. To do this, use the database monitor of the R/3 System.

If the control files or redo log files are in a critical condition for the recovery, the database check ends. You cannot solve this type of error with SAPDBA. You need to take appropriate action with the right Oracle tools. For more information, see the Oracle documentation and [Recovery with SVRMGR \[Page 726\]](#). Once you have removed the error, you can begin the database check again.

If defective data files are found, SAPDBA creates an internal list of the damaged files. Based on this list, SAPDBA determines the further recovery procedure.

Once the database check has been completed successfully (a safe check was performed, no errors found in the control or online redo log files), the **Status** is set to **finished**. If the check did not discover any errors in the database data files, all the other functions are assigned the status

Checking the Database Files

not needed, since no recovery measures are necessary. Otherwise, continue the recovery with the next menu item flagged as *not finished*.

Finding the Backup Files

Select *Find backup files* to locate the backup files. The existing BRBACKUP logs (for a specified period) are scanned for valid backups of the damaged data files. If the backup was performed with an external backup program, SAPDBA attempts to establish a link to that backup program and find the files in this manner. The information on the location of these files is important for the subsequent phases of the recovery process. The following menu items are displayed:

<i>Find backup files</i>	
<i>S</i>	- Start finding backup files
<i>a</i>	- Show the list of damaged files
<i>b</i>	- Show the list of backup files
<i>c</i>	- Select a backup file for restore
<i>d</i>	- Select a BRBACKUP run for restore
<i>q</i>	- Return

Start Finding the Backup Files

For SAPDBA to be able to make copies of the lost files which were backed up with BRBACKUP available, the menu option

- *Start finding backup files*

must be called first. SAPDBA uses BRBACKUP logs to find the backup files.

Define the period for which you want SAPDBA to search for the backup files (default value: the last 30 days).

If you used a non-SAP backup program for the backup, SAPDBA will also attempt to obtain the corresponding information from this program. SAPDBA prompts you to enter the parameter file.

If SAPDBA finds backup files, the necessary log sequence number is determined by SAPDBA as follows: SAPDBA searches for the most recent BRBACKUP file for each lost file and then selects the smallest of the respective log sequence numbers.

In the case of an OPS database system, the smallest log sequence number is provided for each instance ('thread').

Display Functions

- *Show the list of damaged files*

Lists the lost files and their backup files.

Each of the files shown in the list contains one of the following comments:

- Backup file: <name> on <tape/disk> - The file must be restored from tape/disk.

Finding the Backup Files

- backed up by <name of external backup program> - The file was backed up using the specified program. This comment appears when the parameter `backup_util_name` of profile `init<DBSID>.dba` contains the name of the external backup program. Otherwise, the following comment is displayed:
`ext. backup utility`
- no restore of a backup file required - The existing file can be used; it is simply an older version and must be recovered.
- No backup file found - No backup was found for this file in the specified period.
- *Show the list of backup files*
Specify the lost files for which you would like to see the available backup files. Each file that has been lost can have several backup files.

Select Options

- *Select a backup file for restore*
You can change SAPDBA's proposal for which backup file should be restored by using this menu option. The file that is finally selected for the restore process is flagged with `(SELECTED FOR RESTORE)`.
- *Select a BRBACKUP run for restore*
SAPDBA suggests the newest found backup for each individual file from which the requested files `(SELECTED FOR RESTORE)` can be restored. You can change this setting, for example, if all the files for restoring were backed up in one backup run and you only want to specify that one backup run (perhaps to avoid changing tape, which is often necessary if the files belong to different backup runs). The following information is listed:
 - Sequential number of the backup file found (*No.*)
 - Coded time stamp (*coded timestamp*), date (*date*), and time (*time*) of the backup
 - The backup *medium*
 - The number of files found in this backup which are to be restored (*backup files found*): `<files found>/<total files for restoring>`. When `n/n` is displayed, this means that all the selected files were found in this backup.

The log sequence number(s) is not specified as from R/3 Release 31G since this would be too unclear in OPS database systems.

Select *Return* to continue with the recovery process. The function *Find backup files* is assigned the **status finished**.

Restoring the Backup Files

Select the menu item *Restore backup files*.



Restoring will overwrite any existing database files, since files are restored at their original location.

At this point, the restore option is only implemented in connection with a recovery situation. The individual restoring of old data files, tablespaces or full backups for other reasons is supported by the SAPDBA menu item *Restore/Recovery*.

	<i>Restore backup files</i>
<i>a</i>	- <i>Specify restore parameters</i>
<i>S</i>	- <i>Start restore of backup files</i>
<i>q</i>	- <i>Return</i>

The first menu item can be used to change the restore parameters; see [Changing the Restore Parameters \[Ext.\]](#). This is not usually necessary, however, since the parameters are determined based on the backup logs.

Start Restore of Backup Files

During a restore, backups are physically restored with the SAP utility BRRESTORE. If space problems occur during a restore of data files, you will have to use operating system utilities to provide the required space.

Select *Start restore of backup files*.

SAPDBA checks whether the files which are to be restored are still available. If they are, an error message is displayed and you must confirm that SAPDBA may overwrite these files. If you do not allow SAPDBA to overwrite the files and the restore process is terminated at this point.

SAPDBA checks whether there is a backup file for each data file that was lost. If a backup file is missing, the restore process is terminated at this point.

SAPDBA displays the restore parameters. Verify the correctness of the specifications. You can change the restore parameters using the option *Specify restore parameters* (see [Changing the Restore Parameters \[Ext.\]](#)) and then restart the restore process for the backup files.

The SAP utility BRRESTORE is started in order to restore the files. Once the backup files have been successfully restored, the function *Restore backup files* is assigned **Status finished**.

Changing the Restore Parameters

Changing the Restore Parameters

Procedure

Choose **Specify restore parameters** to change the restore parameters. The preset values for BRRESTORE are determined from the initialization profile `init<DBSID>.sap`. It is important that you adhere to the SAP standard for this profile.

The restore parameters depend on the particular situation. For example, this means that the parameter *Archive directory* is not needed for restoring the data files.

<i>a</i>	- BRBACKUP profile	<i>init<SID>.sap</i>
<i>b</i>	- Use (choose) former restore	
<i>c</i>	- Clear list of former restores	
<i>d</i>	- Archive directory	<i><ORACLE_HOME>/saparch</i>
<i>e</i>	- First/second copy of archives	<i>1 2</i>
<i>f</i>	- Device type for tapes	<i>tape tape_auto tape_box pipe pipe_auto pipe_box</i>
<i>g</i>	- Backup utility parameter file	<i><Name></i>
<i>h</i>	- Confirm tape mount requests	<i>yes no</i>
<i>i</i>	- Language	<i>Deutsch English</i>
<i>q</i>	- Return to ... and continue	
<i>r</i>	- Cancel ...	

Check the proposed parameters.

Set the parameter *Confirm tape mount requests* to *no* to activate the unattended mode. This suppresses the mount requests. You must respond to all other confirmation messages.

Choose *Cancel...* to cancel the current restore action.

Choose *Return to ... and continue* to continue with the recovery.

Finding Incremental Backups

Use

Choose ***Find incr. backup*** to search for incremental backups based on one or more full backups. You need to do this if you have just restored corrupt or missing files from one or more full backups.



Use incremental backups to reduce the amount of redo log files you need to import when you recover the database. After you restore the database you only have to import the redo log files that have been written since the incremental backup. This speeds up the recovery of the database.

	<i>Find incr. backup</i>
S	- <i>Find appropriate incremental backup runs</i>
a	- <i>Select an incr. backup run for restore</i>
b	- <i>Show list of found incr. backup runs</i>
c	- <i>Show selected incr. backup run</i>
q	- <i>Return</i>

You cannot use incremental backups to recover the database when:

- Files were restored from partial backups or whole backups. Incremental backups use full backups (level 0) as their reference.
- Backups were restored earlier in another *Partial restore and complete recovery* operation, however the operation was never completed as a recovery. This means that the information about the origin of the files has been lost. This also applies when corrupt or missing files were restored with a tool other than SAPDBA.



If you are sure that the corrupt or missing data files have been restored from a full backup then you can still use SAPDBA to search for incremental backups (you search in a restricted interval of time) and then restore them. If the restoration fails then you either chose an incorrect incremental backup, or a recovery with incremental backups is not possible.

Procedure

Choose:

1. *Find appropriate incremental backup runs*, to search for incremental backups.
2. *Show list of found incr. backup runs* to display a list of the incremental backups that SAPDBA has found.
3. *Select an incr. backup run for restore* to select an incremental backup from the list.
4. *Show selected incr. backup run* to display detailed information on the incremental backup that you have chosen.

Finding Incremental Backups



You can skip the steps *Find incr. backup* and *Restore incr. backup* if you want. To do this, select the next step (for example *Find offline redo logs*) that has the status *not finished*.

Choose *Return* to continue with the recovery. The function *Find incr. backup* is assigned the status *finished* when the process is complete.

Restoring Incremental Backups

Use

Choose **Restore incr. backup** to restore incremental backups.

	Restore incr. backup
a	- Specify restore parameters
S	- Start restore incr. backup run
q	- Return

Procedure

Choose:

- *Specify restore parameters* to change the default restore parameters (for more information, see [Changing the Restore Parameters \[Page 706\]](#)).
- *Start restore incr. backup run* to physically restore the incremental backup.

Choose *Return* to continue with the recovery. The function *Restore incr. backup* is assigned the status *finished* when the process is complete.

Finding the Archived Redo Logs

Finding the Archived Redo Logs

Select **Find archive files** to locate the archived redo log files. The summary BRARCHIVE log (arch<DBSID>.log) is scanned for valid backups of the redo log files required to recover the damaged data files. If the backup was performed with an external backup program, SAPDBA attempts to establish a link to that backup program and find the archived redo logs in this manner. The following menu items are displayed:

	Find archive files
a	- Specify first archive file to be found / restored
S	- Start finding archive files
b	- Show the list of archive files
q	- Return

Determining the Start Point for Restoring Logs

Select *Specify first archive file to be found / restored*.

SAPDBA displays the following log sequence numbers for each instance (thread):

- The fixed number in the BRBACKUP backup used (if no backup was restored, a value is determined from V\$RECOVER_FILE and V\$RECOVERY_LOG)
- The number of the first offline redo log required (compare ORACLE: V\$RECOVERY_LOG)
- The number of the last (online) redo log required

The number SAPDBA suggests

These numbers will normally tally. If necessary, you can change the log sequence number of the first archive redo log file for restoring.

a	Select instance and change current first required log
q	Return



The log sequence number proposed by SAPDBA should only be changed in exceptional cases and by an experienced database administrator. Normally, finding the *archive files* can be started directly with the menu option S described below.

Start Finding the Archived Redo Logs

You must select menu item *Start finding archive files* so that SAPDBA can make available the archived redo log files and offline redo log files belonging to the lost data files. To find the archived redo log files, SAPDBA uses the BRARCHIVE summary log; to find the offline redo logs, it searches the directory where ORACLE archives its online redo log files.

Finding the Archived Redo Logs

If you used an external backup program for the backup, SAPDBA attempts to obtain the appropriate information from that program. You are prompted to enter the required parameter file. SAPDBA automatically suggests the parameter file in the BRARCHIVE profile BRARCHIVE (`init<DBSID>.sap`).

Any gaps in the archived redo logs are displayed. If the sequence of the redo log files required for recovery is interrupted, recovery using the SAPDBA function *Check (and repair) database* is impossible. A partial recovery with the SAPDBA function *Restore/Recovery* or with ORACLE tools is possible, but only with loss of data.

Display Functions for the Archived Redo Logs

Select the option *Show the list of archive files* to display the list of archived redo logs for the respective instance (that is, the logs beginning with the determined log sequence number) that are required for the recovery process. SAPDBA highlights any gaps in the redo log sequence.

Select *Return* to continue with the recovery process. The function *Find archive files* is assigned the **status finished** when the process is complete.



You must get offline redo log files of other instances which were not found from the archiving directories and store them manually or with BRARCHIVE in the local archiving directory. SAPDBA does not perform this action automatically.

Restoring Archived Redo Logs

Restoring Archived Redo Logs

Select the menu item **Restore archive files**.

	<i>Restore archive files</i>
<i>a</i>	<i>- Specify restore parameters</i>
<i>S</i>	<i>- Start restore of archive files</i>
<i>q</i>	<i>- Return</i>

Start Restoring the Archived Redo Logs

During a restore, archives are physically restored with the SAP utility BRRESTORE. If SAPDBA determines that the archiving directory <ORACLE_HOME>/saparch does not have enough space to restore all the necessary redo log files, the redo logs that have already been used will be deleted and the next required redo logs restored during the subsequent recovery.

Select *Start restore of archive files*.

This option is mandatory when the recovery requires offline redo log files that are no longer in the archiving directory. The recovery cannot be started until the necessary archived redo logs are restored (now or continuously during the recovery).



If you have an OPS configuration, the redo log files of all the instances must be restored to the local archiving directory.

SAPDBA displays the following information on the screen:

- The log sequence number of the first archived file to be restored
- The archived files that were found
- The maximum size of the archived redo log files
- The configured restore parameters which you can change, if required, using the function *Specify restore parameters* (see [Changing the Restore Parameters \[Ext.\]](#))

The SAP utility BRRESTORE restores the required files.

If the redo logs are still available on disk, they do not have to be restored. SAPDBA flags such files with the comment (*still on disk - not restored*).

Select *Return* to continue with the recovery process. The function *Restore archive files* is assigned the **status finished** when the process is complete.

Recovering the Database

Procedure

Select *Recover database* to recover the database.

SAPDBA tells you:

- when the recovery started.
- whether the database can be recovered in one step.
- which scripts were recovered to start the recovery.

If sufficient disk space was not available when you restored the archived redo logs, the archived redo log files must be loaded later. SAPDBA automatically deletes the redo logs that have been retrieved from tape and processed in order to make space available. The next redo logs are then restored and the recovery process is continued. You can monitor this process by watching the screen displays.

After successful recovery, the database is opened again.

The recovery process is now complete. The *Recover database* function is assigned the status **finished** if the process is successful.

Afterwards, use the SAPDBA function *Show/Cleanup* → *Show log files/profiles* to check the SAPDBA log file `<timestamp>.rcv` (Directory `<ORACLE_HOME>/sapreorg`).

For more information, see [SAPDBA Logs \[Page 296\]](#).

Full Restore and Recovery

Full Restore and Recovery

Usage

The function *Full restore and recovery* lets you reset the database to a particular *Point in Time*. You can use this function after:

- Failures caused by user errors (such as logical errors)
- Failures during maintenance (such as upgrade errors or data transfer errors)

Prerequisites

As well as the availability of the BRBACKUP/BRARCHIVE logs and the data file backups, the following prerequisite must also be met:

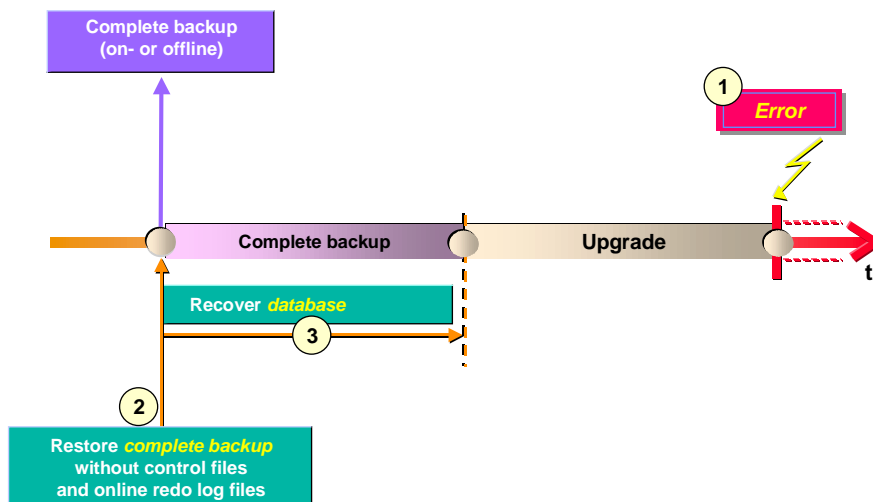
- All offline redo log files between the data backup and the reset point in time must be available.

Features

The function *Full restore and recovery* supports you in the following scenarios:

Scenario 1

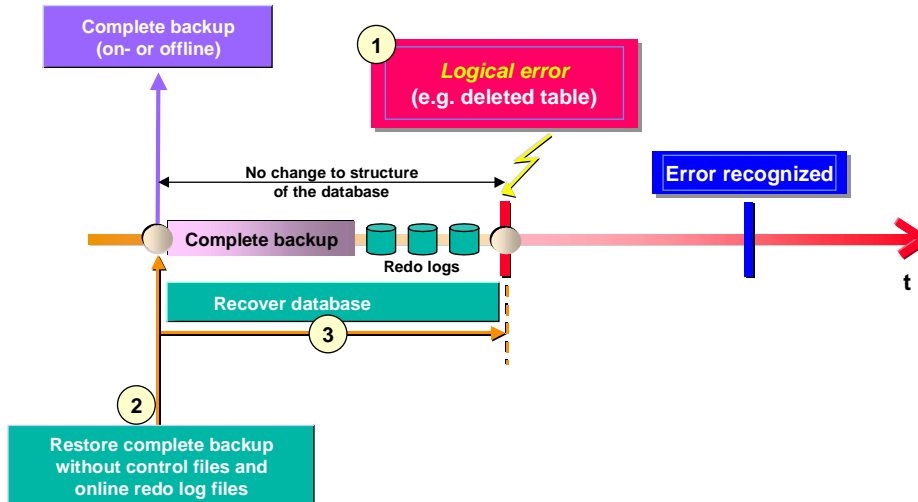
There is an error during an upgrade. You want to recover the database to the point in time before the upgrade.



In the first step, SAPDBA restores the last complete backup without control files and online redo log files, and in the second step recovers the database.

Scenario 2

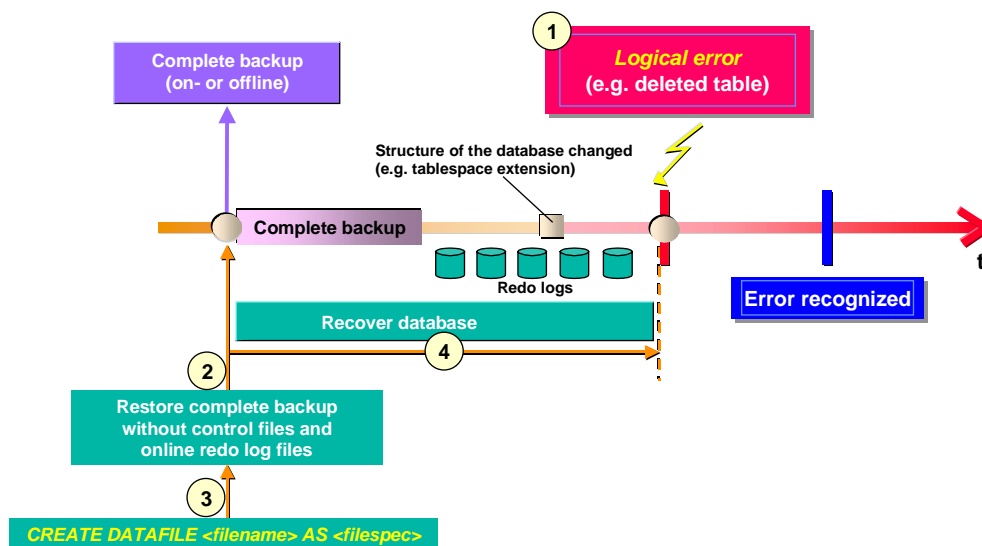
A logical error occurred during normal database operations that was only recognized later. You want to recover the database to the point in time before the error.



In the first step, SAPDBA restores the last complete backup without control files and online redo log files, and in the second step imports the redo log files and recovers the database.

Scenario 3

A logical error occurred during normal database operations that was only recognized later. The structure of the database was changed between the error and the last complete backup. You want to recover the database to the point in time before the error.

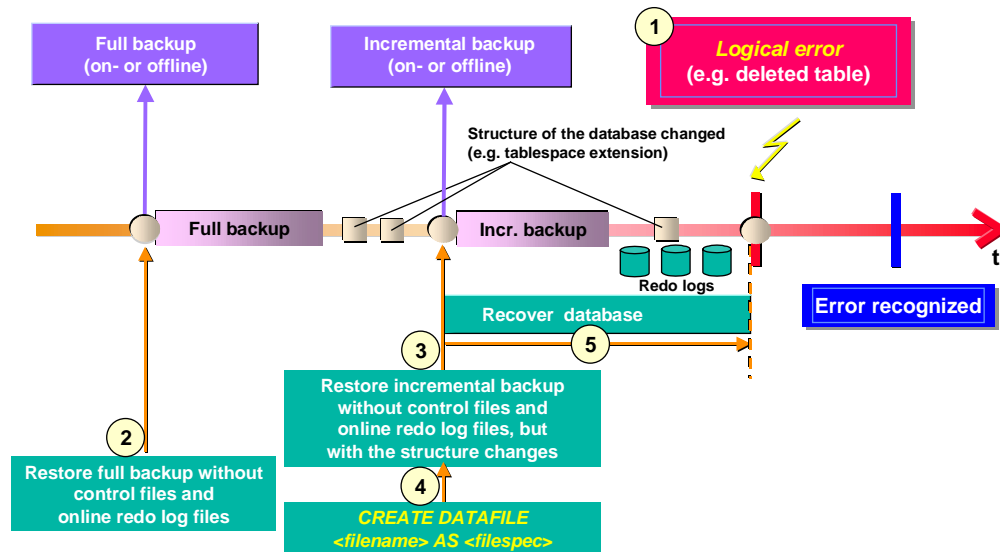


In the first step, SAPDBA restores the last complete backup without control files and online redo log files, and in the second step recovers the structural changes (`CREATE DATA <filename> AS <filespec>`). In the third, the redo log files are imported and the database recovered.

Full Restore and Recovery

Scenario 4

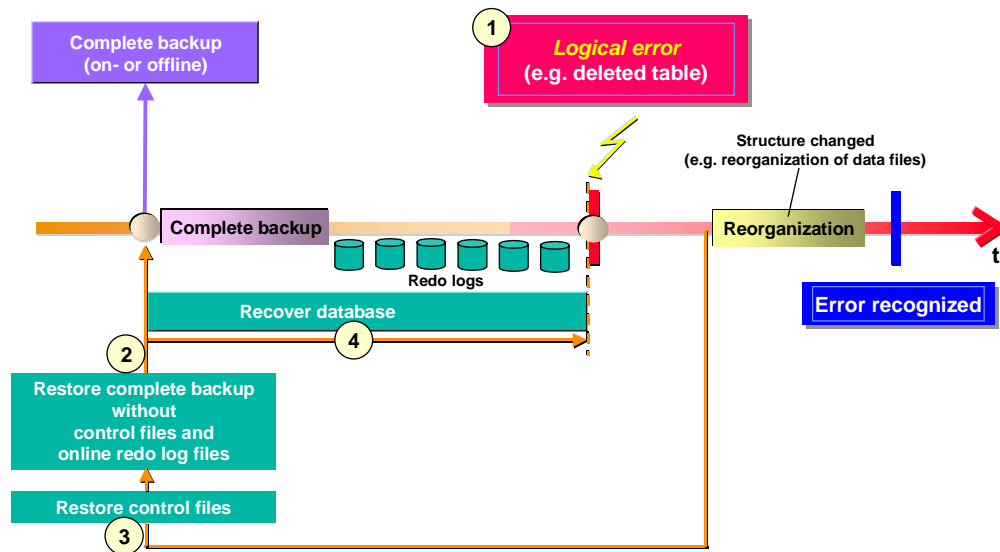
A logical error occurred during normal database operations that was only recognized later. The structure of the database was changed several times between the error and the last complete backup. You want to recover the database to the point in time before the error.



In the first step, SAPDBA restores the last complete backup without control files and online redo log files, and in the second step restores the last incremental backup before the error. The changes to the structure are included in the incremental backup. In the third step, the structural changes made between the incremental backup and the logical error are recovered (`CREATE DATA <filename> AS <filespec>`). In the fourth step, the redo log files are imported and the database recovered.

Scenario 5

A logical error occurred during normal database operations that was only recognized later. The database was reorganized between the error occurring and its discovery. You want to recover the database to the point in time before the error.



In the first step, SAPDBA restores the last complete backup without control files and online redo log files, in the second step restores the control files as they were before the reorganization. During the reorganization the control files were backed up in the directory `<ORACLE_HOME>/sapreorg/`. In the third step, the redo log files are imported and the database recovered. (Recovery with the option and *USING BACKUP CONTROLFILE*).

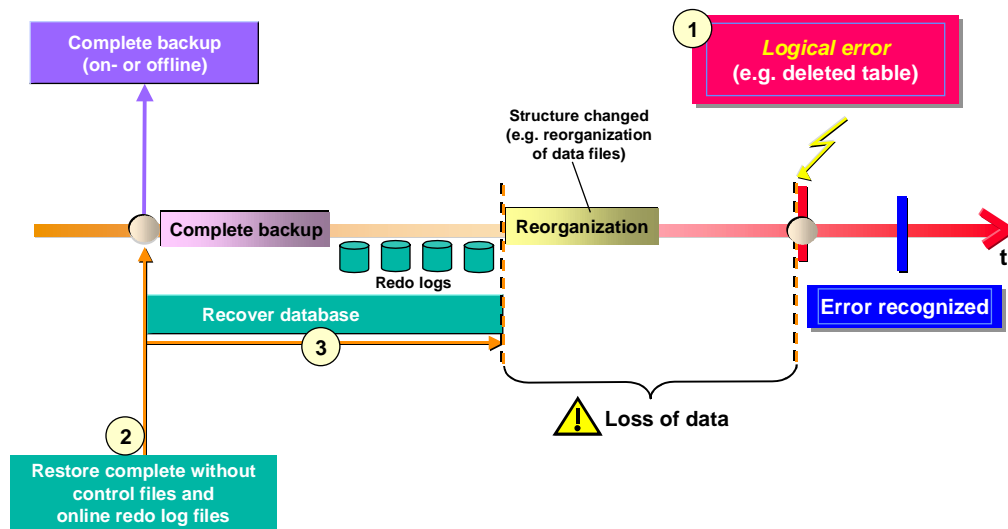
Scenario 6 (not supported by SAPDBA)

A logical error occurred during normal database operations that was only recognized later. The database was reorganized before the error occurred. You want to recover the database to the point in time before the error.



SAPDBA does not support this scenario. The most you can do is to recover the database to the point in time before the reorganization, therefore losing data. Always make a complete backup of the database after you reorganize it.

Full Restore and Recovery

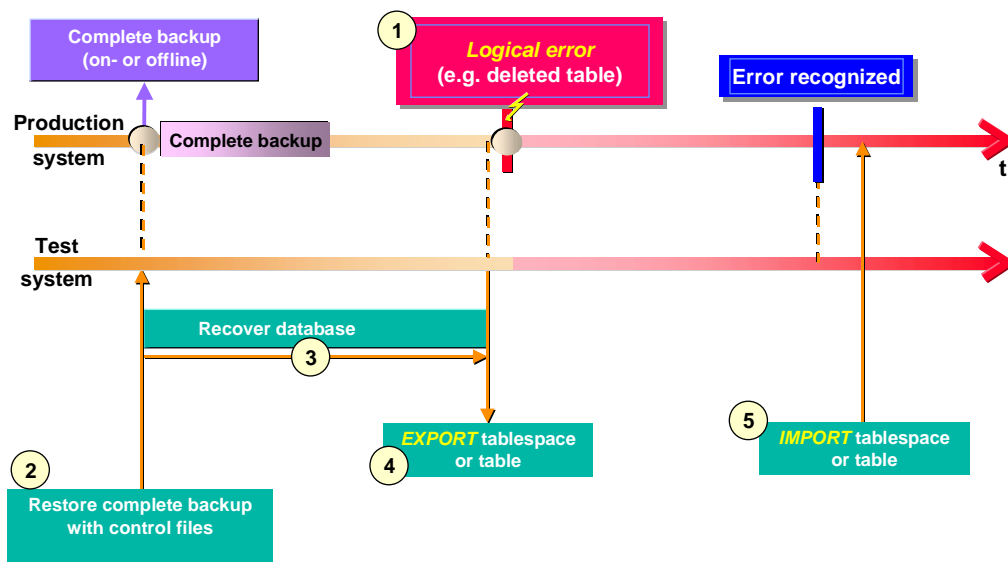


Scenario 7 (risky, not supported by SAPDBA)

In the production system, a logical error occurred during normal database operations that was only recognized later. You want to recover the database to a point in time before the error in a test system. You then want to export the affected tables from the test system into the production system.



This scenario is highly risky and may destroy the database beyond recovery, since the isolated export and import of individual R/3 tables makes the R/3 Data Dictionary inconsistent. You may not be able to start the R/3 System. SAPDBA does not support this scenario.



Activities

Start the function *Full restore and recovery* from the SAPDBA menu entries:

	Full restore and recovery	
	DATABASE STATE	NOMOUNT MOUNT OPEN
	RESTORE/RECOVER	allowed allowed
		Current setting
A	- Select a backup of type	Full backup whole backup
b	- Select incremental backup run	(Only for selected full backup)
c	- Recover until	now point in time
d	- Show status	
e	- Restore and recover	
q	- Return	

Choose *Select a backup of type* to display a list of all possible whole and full backups (on- and offline). Choose the appropriate backup for your needs. If you choose a full backup you can choose the accompanying incremental backup with *Select incremental backup run*.

Choose *Recover until* to enter the point in time to which you want to recover the database. You can choose between *Recover until now* and a point in time of your choice (*Recover until YYYY-MM-DD HH.MM.SS*).

Choose *Show status* to display:

- the backup you want to use.
- the point in time to which you want to recover the database and which redo log files are needed for this.
- whether changes to the structure have been made between the chosen backup and the recovery point in time (for example, after a reorganization with data files or a tablespace extension).
- whether an operation with the chosen settings is allowed.



Always choose *Show status* if the function *Full restore and recovery* is restricted or not possible.

Choose *Options* if you want to use a control file from a backup for the recovery.

Choose *Restore and recover* to start the function *Full restore and recovery*.

After the recovery, check the log file written by SAPDBA

<ORACLE_HOME>/sapreorg/<timestamp>.rsn (for *Recover until now*) or

<ORACLE_HOME>/sapreorg/<timestamp>.rsp (for *Recover until point in time*). To do this, choose the SAPDBA function *Show/Cleanup* → *Show log files/profiles*. For more information, see [SAPDBA Logs \[Page 296\]](#).

Full Restore and Recovery



After the *Full restore and recovery*, the command `ALTER DATABASE OPEN RESETLOGS` is always executed. For security reasons, make a backup before you open the database again. After you open the database, the current *LOG SEQUENCE NUMBER* = 1 and its operations overwrite the old redo log files. Back up the offline redo log files before the function is executed.

Resetting the Database

Select *Full restore (incl. redo logs and control files)*.

This option is intended without recovery and must be based on offline backups because only these have a consistent state and contain all the objects, including the online redo logs.

SAPDBA calls the SAP utility BRRESTORE. BRRESTORE restores all the data files, the control file, and the online redo logs from a valid (completed with return code 0 or 1) offline backup of the entire database. The control file and the redo log files are copied to the appropriate places in the system, so that the original mirrors of these files are available again. Depending on the restore type you select, you can then subsequently recover the database or not.

- *Restore database and startup open (no recovery possible)*

Restore and open the database. No recovery is possible. The database now has the status from the time of the offline backup from which it was loaded. Also see [Partial Recovery \[Page 754\]](#) and [Actions after a Partial Recovery \[Page 757\]](#).

- *Restore database and startup mount (for manual recovery (using backup control file))*

Restore the database. The database is then mounted, and you can perform a recovery using SAPDBA.

The SQLDBA command for recovering the data is as follows (be sure to read the corresponding ORACLE documentation):

```
sqldba> recover database [until cancel] [using backup  
controlfile];
```

This SAPDBA function will probably be seldom used since normally the SAPDBA functionality for restore and recovery of the database can be used.

The log file written by SAPDBA is contained in file `<ORACLE_HOME>/sapreorg/<timestamp>.rst`. To access the file, select the SAPDBA function *Show/Cleanup* → *Show log files/profiles*.

See [SAPDBA Logs \[Page 296\]](#).

Restoring One Tablespace

Restoring One Tablespace

In certain situations, it may be necessary to restore one tablespace. In this case, however, SAPDBA only provides limited support for the recovery. This action should only be used by experienced database administrators.

Select *Restore one tablespace*.

Enter the name of the tablespace. You can selectively restore objects that were backed up using BRBACKUP (or an external backup program). SAPDBA displays the backups it finds. Select the required backup.

Restoring the Files of a Tablespace

Depending on the situation, the program displays important parameters and option values used by BRRESTORE. You can change these values if necessary.

See [Changing the Restore Parameters \[Ext.\]](#).

If the current settings are correct, select the option *Return to restore process and continue* to start the restore process.

During the restore process, the backups are physically restored using the SAP utility BRRESTORE. SAP asks whether you really want to restore the files of the specified tablespace. The database is closed and the files are restored. The database remains closed once the restore process is complete.

Once you have restored the files of the database, you will probably have to recover the database. SAP displays the log sequence number of the first redo log file required for the recovery.

If you have merely restored old data files, and the redo log and control files are current, you can use the SAPDBA menu *Check (and repair) database* to update these files. The check will determine that some of the data files are outdated, and display the period necessary for recovery (see [SAPDBA: Check \(and Repair\) Database \[Page 699\]](#)).

Check the log file written by SAPDBA `<ORACLE_HOME>/sapreorg/<timestamp>.dba`. To do this, select the SAPDBA function *Show/Cleanup* → *Show log files/profiles*.

See [SAPDBA Logs \[Page 296\]](#).

Restoring Individual Files

In certain situations, it may be necessary to restore individual database files (data files, control files, or online redo log files) or non-database files.



Please note that SAPDBA does not check whether restoring the files is sensible. If the restore process causes inconsistencies in the database, you - the database administrator - are responsible for implementing the required recovery measures.

A recovery with SAPDBA (updating the data files) is only partially supported (see [Restoring One Tablespace \[Page 722\]](#)). The restore is performed by calling the SAP utility BRRESTORE.

Select *Restore individual file(s)*.

	Restore individual file(s)	
A	- File type:	<no selection>
b	- BRBACKUP run:	<no selection>
c	- Instance name:	<no need for a selection>
d	- Enter/show files:	<no selection>
S	- Start BRRESTORE	
q	- Return	

Defining the Type of File

Select menu item *File type* to define the type of the files to be restored.

The following files can be restored:

1. Non-database files
2. Database data files
3. Control files
4. Online redo log files
5. Offline redo log files
6. ORACLE profiles `init<SID>.ora` and `init<DBSID>.dba` (OPS: `init.ora/init.dba` profiles of all the database instances)
7. BRBACKUP/BRARCHIVE profile `init<SID>.sap`
8. Main SAPDBA log `reorg<SID>.log` and the structure log `struct<DBSID>.log` (if it exists, also the script `create_raw_links`)
9. Summary BRBACKUP/BRARCHIVE log (OPS: the summary of the BRARCHIVE logs of all the database instances)
10. Detailed BRBACKUP/BRARCHIVE log

Select the required file type. Your selection is indicated by an arrow and displayed in the status line `Current selection: <file type>`.

Restoring Individual Files



Note that all the specified profiles and logs are always reloaded for options 6, 8 and 9. No further selection is possible.

See: command option for BRRESTORE: [-n|-number \[Page 380\]](#).



Note that logs and profiles can only be restored from certain types of backup (device type: `tape`, `pipe`, `tape auto`, `pipe auto`) (see: [-d|-device \[Page 371\]](#)). A restore from disk, for example, is not supported. Please note the appropriate warning messages from SAPDBA.

Select *Return* to continue with the process.

Finding the Backup Files

The selected file type is shown in the *Restore individual file(s)* menu. You can now use menu item *BRBACKUP run* to specify the backups to be used for restoring the files.

- For offline redo log files, it is not necessary to select menu item *BRBACKUP run*, because BRRESTORE can specify the required archives from the summary log `arch<DBSID>.log` (no need for selection). If you have an OPS configuration, you must specify the instance name for the offline redo log files (menu option c).
- Profiles and logs are saved during every backup run. BRRESTORE will load a profile or log from any tape without checking the BRBACKUP/BRARCHIVE run that wrote the tape. As a result, no entry is necessary (no need for selection).
- For the other file types, you can use menu item *BRBACKUP run* to enter the backup runs that were used to save the files of the selected type. The log names and timestamps of the BRBACKUP files found are listed. Select the required backup. SAPDBA always suggests the newest backup.

Selecting/Displaying the Files to Restore

Select *Enter/show files*, if required.

- For control files, it is not necessary to select this menu item, because BRRESTORE can detect the control file in the selected backup automatically (no need for selection).
- To specify online redo log files, enter the group numbers.
- To specify offline redo log files, enter the log sequence numbers.
- Profiles and logs are recognized by BRRESTORE automatically. No entries are necessary (no need for selection).
- The following applies to the other file types:
Select *Enter/show files*. The relevant files and directories are displayed. You can decide which of the files you want to restore from this backup. The following information is displayed:
 - Sequential number of the backup file found (*No.*)
 - File type: directory (*dir*), file (*file*),
 - Flag whether the file was selected for restore (*Res yes*) or not (*Res no*)

Restoring Individual Files

- Complete file name of the file at the time of the backup (*Name <file name>*)

You can flag a file for restoring (*Res yes*) by entering its number or by entering `+<file number>`. Enter `-<file number>` to change the flag back to *no*.

In some cases, the list of files found may be quite long. Use RETURN to scroll through the list.

Only those files assigned the flag *Res yes* are restored. If the selection is correct, press RETURN to continue processing.

Restoring the Files

Select the menu item *Start BRRESTORE*. The important parameters and options used by BRRESTORE are displayed for the specific situation. You can change these values if necessary.

See [Changing the Restore Parameters \[Ext.\]](#).

If the information displayed is correct, select the option *Return to Restore individual file(s) and continue* to start the restore process.

SAPDBA does not check the database state or whether the restore operation is logical. You will probably have to recover the database once you have restored the database files.

During a restore, backups are physically restored with the SAP utility BRRESTORE. BRRESTORE checks whether the files to be restored still exist. If so, a warning message is issued, and you will have to confirm (by entering `cont`) that you want to overwrite these files.

Depending on the data medium selected, other prompts may be displayed involving the type of backup device, the backup program, etc.

Once the backup has been restored, the menu *Restore individual file(s)* menu is displayed again.

Inspect the log file written by SAPDBA `<ORACLE_HOME>/sapreorg/<timestamp>.dba`.

Select the SAPDBA function *Show/Cleanup* → *Show log files/profiles*.

See [SAPDBA Logs \[Page 296\]](#).

Recovery with SVRMGR

[Recovery with SVRMGR: Overview \[Page 727\]](#)

[Types of Database Errors \[Page 728\]](#)

[Recovery after User Errors \[Page 731\]](#)

[Recovery after Statement Errors \[Page 732\]](#)

[Recovery after Process Errors \[Page 733\]](#)

[Recovery after an Instance Error \[Page 734\]](#)

[Recovery after Media Errors \[Page 735\]](#)

Recovery with SVRMGR: Overview

Although SAPDBA recovery functions can in most cases be used to recover a failed database, there are exceptions to this. Therefore, this chapter contains a description of several general procedures for recovering the ORACLE database system.

It provides assistance for database recovery after an error in one of the database components:

- Data files
- Online redo log files
- Control files

Please note that an incorrectly performed recovery may lead to irreparable loss of data in your database. You bear the sole responsibility for any damage caused to the database by an incorrect recovery. Please avoid such situations by talking to an experienced co-worker, your SAP consultant, or an ORACLE specialist before starting a recovery.

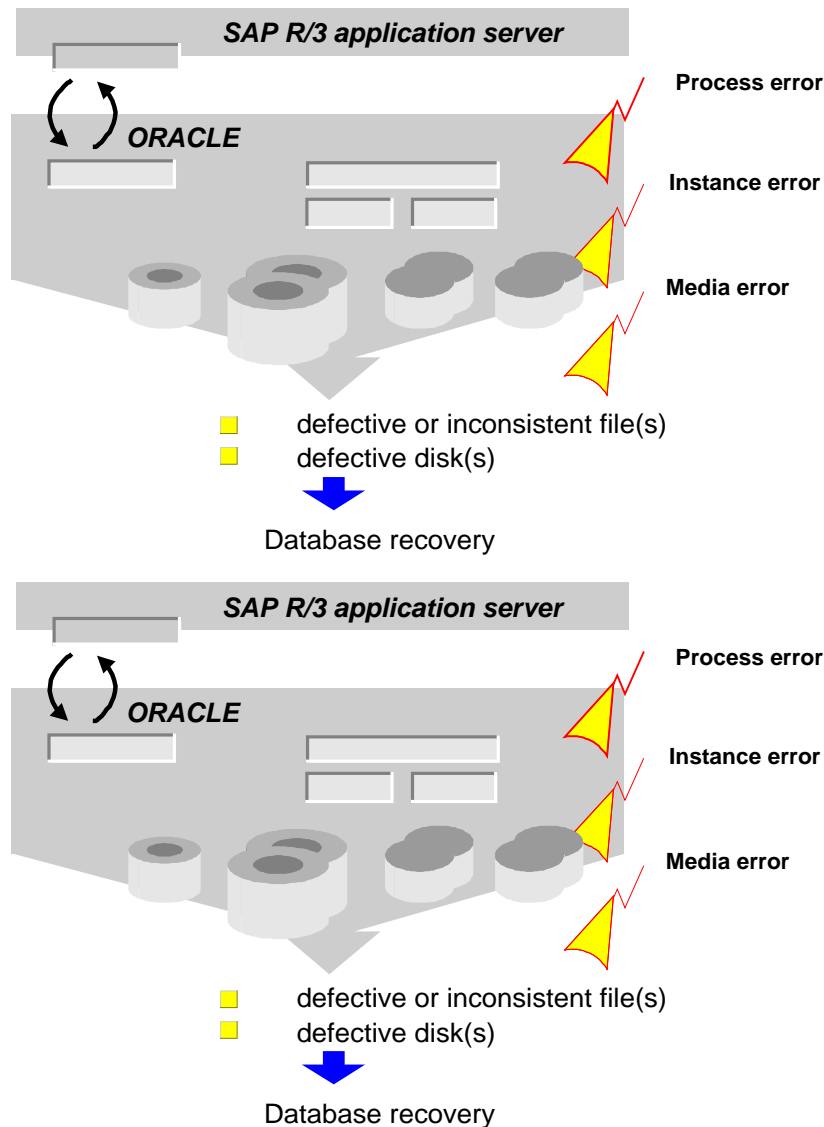
Types of Database Errors

Types of Database Errors

Use

There are a number of errors that hinder the operations of an Oracle database or can cause the database to crash. Depending on the type of error, a recovery can either be carried out automatically or must be performed by the user or database administrator.

It is therefore important for you to find out the exact type of error before intervening in any way, to be able to act correctly and adequately.



Errors can generally be divided into five categories:

- User errors, see [Recovery after User Errors \[Page 731\]](#)
- Statement errors, see [Recovery after Statement Errors \[Page 732\]](#)

Types of Database Errors

- Process errors, see [Recovery after Process Errors \[Page 733\]](#)
- Instance errors, see [Recovery after an Instance Error \[Page 734\]](#)
 - Media errors, see [Recovery after Media Errors \[Page 735\]](#)

[Error Analysis \[Page 730\]](#)

Error Analysis

Error Analysis

Information on all types of database problems can be found in the trace and ALERT files, where the actions in the database are recorded.

- In the ALERT file, important system events and significant error messages are continually collected for the entire system. The ALERT file can inform you, for example, which of the online redo log groups is the current one.

ALERT file: `<ORACLE_HOME>/saptrace/background/alert_<DBSID>.log`

- Trace files are created by the individual background and user processes of the database system. Trace files thus offer more detailed information for error analysis than the ALERT file.

Trace files of the background processes:

`<ORACLE_HOME>/saptrace/background/<process>_<number>.trc`

Trace files of the user processes:

`<ORACLE_HOME>/saptrace/usertrace/<process>.trc`

The directories where the ALERT and trace files are stored are defined by the `init.ora` parameters `background_dump_destination` and `user_dump_destination`.

The naming conventions for the trace files depend on the operating system. Under UNIX, the name of the trace file contains the operating process number, the ORACLE process number and the ORACLE system ID.

A trace file can contain the following information:

- Date and timestamp
- Version number
- Process information
- Error messages
- Calling stack trace
- Dumps from the System Global Area (SGA) and the Program Global Area (PGA).

Recovery after User Errors

A user error occurs when a user deletes or falsifies data (e.g. deletes a table or program that is required for further system operation), either by mistake or due to lack of knowledge.

You can generally correct such a user error when the following requirements are met:

- You exported the object using the SAP utility R3trans, thus backing it up. You can use this copy to restore the condition of the object at the time of the export. (Take into account possible database inconsistencies).
- An object from the ABAP/4 Dictionary or the ABAP/4 Repository is involved. The ABAP/4 Dictionary and the correction system both perform version backups of these objects within the SAP System. If you can carry on working with that version of the object (ideally, the object has not been changed recently), then you can restore it.

In general, you cannot use the ORACLE Export/Import tools to recover a lost SAP object. The reason for this is that the SAP database tables are often shared system-wide. A user cannot import the ATAB (central control table) to recover an individual SAP table, for example, as this would risk overwriting the work of other users.

You cannot recover an object by recovering the database either. The recovery of a lost object would require an incomplete recovery up to the moment the user error occurred (point in time recovery). Any changes which were made to the tablespace from that moment on would be lost.

Recovery after Statement Errors

Recovery after Statement Errors

A statement error occurs when an error in executing a database statement occurs in a running program.

If such an error occurs, the database system cancels the incorrect statement and issues an error message. The program containing the error is terminated. Furthermore, the transaction is completely canceled in the database (rollback), an ABAP/4 dump is issued, and a message is usually written to the system log.

A statement error can also occur if an extensive operation entirely fills up the rollback segment. The reason for such an error is generally incorrect programming.

The database administrator does not have to intervene in order to execute a recovery after a statement error.

Recovery after Process Errors

A process error occurs when a user process is canceled.

The ORACLE instance is not normally affected by the termination. The ORACLE process monitor (PMON) responds by canceling the database changes made by current transactions and releasing the resources which were used by the process. Work with the database system can then be continued normally.

The database administrator does not have to intervene to carry out a recovery after a process error.

Recovery after an Instance Error

Recovery after an Instance Error

An instance error occurs when the ORACLE database instance (System Global Area (SGA) and the corresponding background processes) can no longer run.

An instance error can result from a hardware problem (such as a power failure) or a software error (for example, the crash of the operating system or of an ORACLE background process).

An instance error generally results in an immediate abnormal termination of the entire instance. Even if the database system remains active, the data in the SGA is lost in any case, and the instance can no longer be shut down in the conventional way (with `shutdown normal` or `shutdown immediate`).

Since only an abnormal termination is usually possible (`shutdown abort`) the instance must be recovered. Only transactions completed normally ("COMMIT") can be processed; all others are rolled back. If you are working with a standard configuration, the system monitor (SMON) of the database system automatically carries out the recovery of the instance when you restart the instance (instance recovery). It uses the entries in the appropriate redo log files to do so. In the case of an OPS configuration, recovery of the instance is performed online by another instance.

You do not need to intervene during the recovery, provided no database files were changed. At least one copy of the control file, all data files, and at least one online redo log file in each online redo log group must exist.

Before you restart the database system, you should find out which background process was canceled and why. Check the ALERT and trace files.

Recovery Procedure

If you want to perform a recovery after an instance error, shut down the instance (if it is still active) with the command `shutdown abort`. Analyze and correct the problem. Restart the database with the command `startup`.

When you restart the database, the system monitor performs an automatic instance recovery, as described above.

Recovery after Media Errors

A media error occurs when a read or write error takes place in a database file due to a physical defect on the disk drive where the file is located. The most frequent physical defect is a head crash, which usually renders all the files on a disk unusable.

In most cases, the database must be recovered after a media error. The recovery strategy depends on the type of damage in the database. You must therefore analyze the error and understand how it was caused before you can continue with the recovery.



If you are not sure how to perform a recovery, please contact your SAP consultant or an ORACLE specialist. Do NOT attempt to just "give it a try".

An incorrectly performed recovery can lead to irretrievable loss of data in your database. The damage you risk is much more costly than the time you spend on a carefully planned, properly executed recovery of your database.

In addition to the ALERT and trace files, SAP recommends using the SAPDBA Check function (*Check (and repair) database* → *Check database*) to analyze the database. If defects in the data files of the database are involved, SAPDBA can often perform the database recovery automatically.

See [SAPDBA Database Recovery \[Page 697\]](#).

This section is intended to provide an overview of the recovery process using ORACLE utilities. Use the ORACLE utility SVRMGR. The sections below contain examples of the command line operations with SVRMGR.

If you have followed the SAP recommendations regarding the installation and operation of the database system, you should generally succeed in carrying out a complete recovery after a media error. If, however, in an extreme case your backups of the database and your redo log archives have been damaged too, you may only be able to carry out an incomplete recovery.

The procedures required for the following errors are listed in the following:

- Loss of one or all copies of the control file
- Loss of an online or offline redo log file or even an entire online redo log group
- Loss of a system, temporary, user, or index tablespace
- Error in the archiving process for online redo log files using ARCH

These errors are usually media-related (e.g. if the archiving directory for the offline redo log files is lost or full).

Recovery after Media Errors

[Recovery: One Control File Missing \[Page 737\]](#)

[Recovery: All Control Files Missing \[Page 738\]](#)

[Recovery: Current Online Redo Log Missing \[Page 742\]](#)

[Recovery: One Inactive Online Redo Log Missing \[Page 744\]](#)

[Recovery: User Tablespace Missing \[Page 746\]](#)

[Recovery: SYSTEM Tablespace Missing \[Page 749\]](#)

[Recovery: Index Tablespace Missing \[Page 751\]](#)

[Recovery: Errors During the Archiving of Online Redo Logs \[Page 752\]](#)

[Partial Recovery \[Page 754\]](#)

[Actions after a Partial Recovery \[Page 757\]](#)

[Automating the Recovery \[Page 759\]](#)

[Updating the Control File \[Page 760\]](#)

Recovery: One Control File Missing

A copy of the control file cannot be read or written to.

Database activities continue normally until the next update of the control file - that is, until the next checkpoint or redo log file switch - and the database then crashes.

In this case you do not need to carry out an explicit recovery of the database. The system can carry out a recovery automatically, providing a copy of the control file exists.

Recovery Procedure

There are two main options for recovery after one of the control files is lost:

- Replace the missing control file

Find out why the control file was lost. For example, you cannot replace the file when the disk is not available.

Check the ALERT and trace files of the database system to analyze the error. The error messages will indicate which control file is missing, and may also indicate how it was lost.

Replace the missing control file control file as follows:

- Shut down the database and exit SVRMGR.
- Copy an existing control file to the desired storage location at operating system level (see parameter `control_files` in the `init.ora` profile).
- Start SVRMGR and start up the database.

- Modify the database system

Check the ALERT and trace files to find out which control file was lost.

- Shut down the database and exit SVRMGR.
- Delete the missing file from parameter `control_files` in profile `init.ora` (default: `<ORACLE_HOME>/dbs/init<DBSID>.ora`).
- Start SVRMGR and start up the database. No other actions are required for recovery.

This solution is only acceptable if you have at least two copies of the control file. This guarantees that at least one mirror copy of the file still exists, even when one control file has been deleted. The database should always have at least two control files (original and mirror). In the SAP System, the control files are often mirrored in three or more directories (see [Mirroring the Control File \[Page 36\]](#)).

Recovery: All Control Files Missing

Recovery: All Control Files Missing

You have lost all copies of the control file.

The normal database activities continue until the next update of the control file. When this happens (during the next checkpoint or redo log file switch, at the latest), the database system crashes.

A complete recovery of the database is possible provided one of the following conditions is fulfilled:

- A current backup copy of the control file, that is, a copy with the current structure of the database, exists.
- A current log of the files in the database exists, enabling you to create the control file again.

If all the control files (even the backups) are lost, you must first reconstruct the control file before you can start the recovery process. This procedure is much more complicated and not always guaranteed to be successful. Please therefore strictly observe the following note, which cannot be repeated often enough:



Follow SAP's recommendations and back up your database after every structure change! If you do this, you always have a backup of a control file that reflects the current structure of the database.

Recovery Using the Backed-Up Control File

In the procedure described in the following, we assume that you are able to restore the control file from your last database backup.

To update the database, the appropriate redo log files must exist.

It is essential that the saved control file always reflects the current structure of the database. The paths for the data and log files and the status of the log sequence numbers are not important, but the control file must have the exact information about the number of files and - indirectly - the number of tablespaces in the database.

Proceed as follows for recovery:

1. If the database system is still operating, shut down all instances with the following SVRMGR command:

```
shutdown abort
```

ABORT is generally necessary because the control files are no longer available to include a checkpoint during the shutdown.

2. Use the ALERT and trace files to analyze the error.

Check whether other damage has occurred to the database: Find out whether all data files and redo log files are readable.

Back up the online redo log files of all instances (if they exist in readable form) so that you can repeat the recovery process if an error occurs.

Recovery: All Control Files Missing

3. Place the backup copies of the control file in the directories or on the raw devices specified in the `control_files` parameter in the `init.ora` profile.

If further files were damaged, restore the backup copies of these files. You do not need to restore undamaged files from the backup. If you have to restore data files, you will also have to restore all the offline redo log files of all instances that have been archived since the last backup (for SAP databases, offline redo log files are usually archived by the BRARCHIVE program) in the local archiving directory (default value: `<ORACLE_HOME>/saparch`). For detailed information on recovery after the loss of redo log or data files, please refer to the relevant parts of this documentation and to your ORACLE documentation.
4. Enter the following SVRMGR commands to mount the local instance:

```
connect internal
startup mount
```
5. If you could not load backed up files into their original directories or had to change file name, you must update the control file.

Change path or group names with the following command:

```
alter database rename file '<file name>' to '<file name>';
```

See [Updating the Control File \[Page 760\]](#).
6. If the data files of the database were set to status OFFLINE during the shutdown, change the status of the files in the control file to ONLINE.

To find OFFLINE files, search for "offline" in the ALERT file.

To change the status of a data file in the control file, use the following command:

```
alter database datafile '<file name>' online;
```

See [Updating the Control File \[Page 760\]](#).
7. Start recovery with the following SVRMGR command:

```
recover database until cancel using backup controlfile;
```
8. If you are prompted to do so, enter the full path name for the redo log files required for recovery, including the active online redo log file.
9. When all redo log files are processed, end the recovery process with the command `cancel`.
10. After the message `recovery canceled`, you can reopen the database by using the following SVRMGR command:

```
alter database open resetlogs;
```

or

```
alter database open noresetlogs;
```

The RESETLOGS option initializes the existing online redo log files. Therefore, only use this option after a partial recovery. If a full recovery has been performed, you should not use this option.

The NORESETLOGS option causes the online redo log files to be used in their current form. A complete recovery is required to use this option.

Recovery: All Control Files Missing

The database system resumes operations with the log sequence number following the number of the last current online redo log file.

11. Carry out a complete backup of the database.

The backup is necessary to back up the control file and to guarantee a full recovery of the database if further database problems occur. See also: [Actions after a Partial Recovery \[Page 757\]](#).

DB Recovery Using a New Control File

If you do not have a valid copy of the control file, a full recovery is still possible by reconstructing the control file. To do this, you need a current log of all the database files, for example, the BRBACKUP log. Caution: If a structural change to the database has been carried out after this backup, it may be impossible to perform a recovery (e.g. if no backup of the new or changed data files is available).

Proceed as follows during recovery:

1. If the database is still active, shut down all instances with the following SVRMGR command:

```
shutdown abort
```

ABORT is generally necessary because the control files are no longer available to include a checkpoint during the shutdown.

2. Use the ALERT and trace files to analyze the error.

Make sure no further damage has occurred in the database, and find out whether all data files and online redo log files exist in readable form.

Back up the online redo log files of all instances (if they exist in readable form) so that you can repeat the recovery process if an error occurs.

3. If other files were damaged, restore the backup copies of these files. You do not need to restore undamaged files from the backup. If you have to recover data files, also restore all the offline redo log files of all instances that have been archived since the backup of these data files in the archiving directory.
4. Enter the following SVRMGR commands to demount the database:

```
connect internal
```

```
startup nomount
```

5. Use the following command to create the control file (for detailed syntax information, please refer to your ORACLE documentation):

```
create controlfile
database <name>
logfile '<online redo log groups>'
noresetlogs|resetlogs
maxlogfiles 10
maxlogmembers <your value>
datafile '<names of all data files>'
maxdatafiles 254
archivelog;
```

Path names: The path names of the online redo log files and data files can be found in the last detail log from BRBACKUP.

Recovery: All Control Files Missing

Noresetlogs/Resetlogs: Only select RESETLOGS when an online redo log group was lost in addition to the control file. You should otherwise **always** use NORESETLOGS.

6. Mount the database.

```
alter database mount;
```

7. Start the recovery with the following command (a recovery is required whenever the control file was generated with the RESETLOGS object or when a data file was restored. Recovery is recommended for security reasons in other cases, as well.):

```
recover database [until cancel] [using backup controlfile];
```

You must select the option `using backup controlfile` when you used the RESETLOGS option to create the control file. If you select `until cancel`, you can interactively decide how many files of the existing redo log files you want to read during the recovery. You should enter all the redo log files of all instances, including the current ones.

8. Use the following SVRMGR command to start up the database:

```
alter database open [noresetlogs/resetlogs];
```

- Use `alter database open` if you created the control file with NORESETLOGS and have performed **no** recovery or a **full** recovery (**without** `until cancel`).
 - Use `alter database open noresetlogs` if you created the control file with NORESETLOGS and performed a full recovery **despite** the use of the `until cancel` option.
 - Use `alter database open resetlogs` if you created the control file with RESETLOGS **or** when you performed a partial recovery.
9. After the recovery, be sure to perform a **complete backup** to save the newly created control file and to ensure that a recovery of the database in the event of failure is possible.

Recovery: Current Online Redo Log Missing

Recovery: Current Online Redo Log Missing

A member of the group - or the entire group - of current online redo log files (the redo log files in which the database changes are currently being recorded) is lost.

Use the entries in the ALERT file and in the LGWR trace file to analyze the error situation. Also use the SAPDBA Check function (*Check (and repair) database* → *Check database*). It is the database administrator's responsibility to check all sources of information for possible LGWR errors. Even if an error allows the instance to continue running (e.g. at least one member of the current group can be written to, errors only in the other members), the error should be corrected as soon as possible.

If you have not been mirroring the online redo log files, as supported by ORACLE (and have also not been using hardware based mirroring), the risk of losing online redo log entries is significantly higher. In order to be able to perform a full recovery, only the entries from the current online redo log file are necessary. Use mirroring to guard against complete loss of the online redo log files. If the mirrored online redo log files are available, you can use these later to perform a complete recovery of the database. Otherwise you will only be able to recover the database to the point of the missing redo log entries (i.e. a partial recovery with loss of data).

For this reason, we strongly recommend once again that you make use of the ORACLE options for mirroring the online redo log files.

Recovery Procedure: Mirrored Current Redo Log Files

For this recovery procedure, we assume that the following requirements are met:

- You used the ORACLE option for mirroring the online redo log files (or have hardware-based mirroring), and therefore have at least one copy of each online redo log file (SAP default: one copy of the online redo log files).
- Apart from the one member of the current online redo log group, no other files have been damaged.

If further files have been damaged, restore the missing files and the missing active redo log and follow the recovery procedure for the category of the missing file (control file or files of the system, user, or index tablespaces).

Proceed with the recovery as follows:

1. If the database system is still active, use the SVRMGR command `shutdown abort` to shut it down.
2. Determine the reason why the current online redo log files were lost. Examine the ALERT and trace files.
3. Replace the missing online redo log files with one of their mirrored copies.
4. Start the database with the SVRMGR command `startup`. The system automatically performs an instance recovery.



If all members of the current redo log group have been lost, you can only perform a partial recovery. For more information, please refer to the documentation on your ORACLE database system or the section [Partial Recovery \[Page 754\]](#).

Recovery: One Inactive Online Redo Log Missing

Recovery: One Inactive Online Redo Log Missing

If only one member of an inactive online redo log group has been lost, you can use the recovery procedure described in the previous section. Experienced users can correct this error without shutting down the database. For more information, please refer to the ORACLE documentation.

The recovery procedure is different when the following situation occurs:

The database pauses because a redo log switch to an online redo log file was unsuccessful. None of the members in this inactive online redo log group can be read or written to.

If the problem is temporary (e.g. incorrect access rights), you only need to correct it, and you can then use the group again. If the files have been destroyed, however, the group can not be used again.

No data is lost, providing the missing redo log file was fully archived and the archived copy can be read.

Recovery Procedure

Proceed as follows during recovery:

1. Use the following SVRMGR command to shut down the database:

```
shutdown abort
```

ABORT is needed because the database system cannot carry out a proper shutdown, due to the damaged group.

2. Find out which file is missing, and check the ALERT and trace files for the reason why the redo log files were lost.
3. Mount the database. Enter the following SVRMGR commands:

```
connect internal
```

```
startup mount
```

4. If you were running the database in ARCHIVELOG mode and archiving of the damaged online redo log group was not complete, you will have to temporarily switch to NOARCHIVELOG mode before deleting the defective group, because the system will not otherwise allow you to delete the files:

```
alter database noarchivelog;
```

5. You can also delete the damaged online redo log files as a group with

```
alter database drop group <group number>;
```

or as individual files with

```
alter database drop logfile '<file name>' [,<file name>'];
```

6. To create the new online redo log files (to replace the damaged ones, which you just deleted), enter:

```
alter database add logfile '<complete file name>;'
```

7. If the database was set to NOARCHIVELOG mode during these actions, change it back to ARCHIVELOG mode now:

```
alter database archivelog;
```


Recovery: One Inactive Online Redo Log Missing

8. If you were running the database in ARCHIVELOG mode, and the archiving of the online redo log group was not complete at the moment the problem occurred, it is essential that you now perform an offline backup of the entire database. If you do not, the offline redo log chain will have a gap, and only a partial recovery would be possible if another media error occurred.

Recovery: User Tablespace Missing

Recovery: User Tablespace Missing

One or more data files are missing from a tablespace.

A user tablespace does not contain data from the Data Dictionary, active rollback segments, or temporary segments.

ORACLE issues error messages when a user attempts to access the involved tablespace. Error information is also written to the database ALERT and trace files.



If only one user tablespace is lost, you can perform a tablespace recovery without shutting down the database. Please note, however, that user tablespaces are used intensively in the SAP System, and this procedure is therefore only recommended for experienced database administrators. The loss of a user tablespace often has similar consequences for the SAP system as the loss of the SYSTEM tablespace, because the effects of the loss of this one tablespace are very complex.

A full recovery of the database is possible if you have a backup copy of the corresponding tablespaces and of all redo log files written since the backup.

Recovery Procedure

If you use an SAP database, you should shut down the SAP System before starting the recovery procedure. Tables are used so intensively in the SAP System that it is generally impossible to set the affected tablespace to OFFLINE without terminating the activities of many users.

The information below describes the recovery procedure when the database is closed. For information on recovery options with an open database, please refer to the ORACLE documentation.

Proceed as follows during recovery:

1. Shut down the database system, if it is active, with the following SVRMGR command:

```
shutdown abort
```

You have to shut down the database with ABORT because the missing files cannot be closed.

2. Inspect the ALERT and trace files to determine the cause of the problem.

The problem is often that an entire disk has crashed, and you will have to recover more than one tablespace.

3. Use the log files from the SAP utilities BRBACKUP and BRARCHIVE to find the volume(s) that contain the following files:

- Last backup of the lost tablespace(s)
- Offline redo log files of all instances archived since the last backup

It is important to identify the location of the lost files. This information appears at the start of the detail log from the backup you are using.

4. Restore only the damaged or lost files. You can minimize the time required for recovery by only restoring the missing or damaged files.

Recovery: User Tablespace Missing

You also have to restore the archived redo logs of all instances that are required for the recovery.

To do this, use the SAP utility BRRESTORE (see [Command Options for BRRESTORE \[Page 413\]](#)).

5. Mount the database. Enter the following SVRMGR commands:

```
connect internal
startup mount
```

6. If you could not restore backed up files into their original directories or had to change file name, you have to update the control file.

Use the following command to change a path:

```
alter database rename file '<file name>' to '<file name>';
```

See [Updating the Control File \[Page 760\]](#).

7. If the data files of the database were set to status OFFLINE when the error occurred, change the status of the files in the control file to ONLINE.

To find the relevant files, search for “offline” in the ALERT file.

To change the status of a data file in the control file, use the following SVRMGR command:

```
alter database datafile '<file name>' online;
```

See [Updating the Control File \[Page 760\]](#).

8. Use the following SVRMGR command to start the recovery:

```
recover database;
```

When prompted to do so, enter the paths of the offline redo log files required for updating the backup.

The system processes online redo logs automatically.

The alternatives: `recover tablespace` and `recover datafile` were not discussed here, since SAP recommends shutting down the database when an error occurs. The `recover database` command only performs the actions necessary to recover the database, and therefore does not take much longer than the `recover tablespace` and `recover datafile` commands.

For information on `recover tablespace` and `recover datafile`, please refer to the ORACLE documentation.

9. When the message `recovery complete` is displayed, use the following SVRMGR command to start up the database system:

```
alter database open;
```

For more detailed information, please refer to the ORACLE documentation.



Please note: in most cases, SAPDBA can be used to correct media errors affecting the data files of a user tablespace. Therefore, we recommend using the SAPDBA recovery functions (see [SAPDBA Database Recovery \[Page 697\]](#)).

Recovery: User Tablespace Missing

Recovery: SYSTEM Tablespace Missing

One or more files of the SYSTEM tablespace has been damaged or lost due to a media error.

Backups of the affected files exist. All offline redo log files that have been written since the last backup are available uninterrupted. The control files and all online redo log files are undamaged.

Recovery Procedure

Because the SYSTEM tablespace is affected, the recovery must take place with the database closed. If a backup of the missing files and all redo log entries are available, a complete recovery is possible.

1. If the database system is still active, use the following SVRMGR command to shut it down:

```
shutdown abort
```

ABORT is required because the loss of individual files from the tablespace means that the changes from SGA can no longer be recorded, and the database cannot be closed properly as a result.

2. Examine the ALERT and trace files to determine the cause of the problem.
3. Use the log files created by the SAP utilities BRBACKUP and BRARCHIVE to find the volume(s) containing the following files:
 - Last backup of the SYSTEM tablespace
 - Offline redo log files of all instances archived since the last backup
4. Restore the backups of the damaged/lost files and the archived offline redo log files of all instances.

Use the SAP utility BRRESTORE (see [Command Options for BRRESTORE \[Page 413\]](#)).

5. Mount the database. Enter the following SVRMGR commands:

```
connect internal
```

```
startup mount
```

6. You may have to change the names and paths of the files in the control file. See [Updating the Control File \[Page 760\]](#).
7. The database administrator can automate the recovery using the `autorecovery` option. See [Automating the Recovery \[Page 759\]](#). Also read the corresponding ORACLE documentation.
8. Use the following SVRMGR command to start the recovery:

```
recover database;
```
9. Depending on the recovery mode (`autorecovery on/off`), the required offline redo log files are either processed automatically or you will have to enter their paths and names. The system reads the online redo log files automatically.
10. When the message `recovery complete` is displayed, use the following SVRMGR command to open the database again:

```
alter database open;
```

Recovery: SYSTEM Tablespace Missing

Please note: in most cases, SAPDBA can be used to correct media errors affecting the data files of a user tablespace. Therefore, we recommend using the SAPDBA recovery functions (see [SAPDBA Database Recovery \[Page 697\]](#)).

Recovery: Index Tablespace Missing

Generally, ORACLE treats an index tablespace just like a user tablespace. You can therefore apply the recovery procedure for a user tablespace as described above. An additional recovery option also exists for SAP databases, however; it is described below.

Recovery Procedure

1. Use the following SVRMGR command to shut down the database:

```
shutdown abort
```
2. Find out which data file is affected by the media error, using the information in the ALERT and trace files and/or the Check function from the SAPDBA recovery menu.
3. Mount the database. Enter the following SVRMGR commands:

```
connect internal  
startup mount
```
4. Set the data files to OFFLINE with:

```
alter database datafile '<complete file name>' offline;
```
5. Open the database with:

```
alter database open;
```
6. First make sure that the index tablespaces do not contain any tables (you can check this using the ORACLE tables DBA_SEGMENTS and DBA_TABLES).
7. Use the corresponding SAPDBA function to export the affected index tablespace (Export of table and index definitions with options *Compress extents: no, Use SAP DDIC for NEXT: no, Reduce object size: no, Change storage manually: no*).
8. Determine the default storage parameters that apply to this tablespace by examining ORACLE table DBA_TABLESPACES. Write down these values.
9. Enter the following SVRMGR commands to delete the index tablespace and create it again:

```
alter tablespace <tablespace name> offline;  
drop tablespace <tablespace name> including contents;
```

Delete the files belonging to the tablespace at operating system level, if any still exist.

```
create tablespace <tablespace name>  
datafile '<complete file name>' size <n>M  
default storage (<storage parameters>);
```

For more information on these ORACLE commands, please refer to the ORACLE documentation.
10. Use the SAPDBA import functionality to import the indexes (script `ind<TSP>.sql`). The recovery of the index tablespace is complete.

Recovery: Errors During the Archiving of Online Redo Logs

Recovery: Errors During the Archiving of Online Redo Logs

For SAP systems with productive ORACLE databases, the database mode is set to ARCHIVELOG. This mode causes the database system to save an online redo log file (i.e., the archive process ARCH is initiated and produces the corresponding offline redo log files in the archiving directory) before the database changes recorded in the redo log files are overwritten during a log file switch.

If the background process ARCH for archiving redo logs fails, the system shuts down until the error is corrected.

There are two different causes for this problem:

- The background process ARCH was not started
- The archiving directory is full (“archive stuck”) or is not available

Recovery Procedure

Proceed as follows during recovery:

1. Make sure the ARCH archiving process was started.

Enter the following SVRMGR commands:

```
archive log list
```

If the ARCH process was not started or is not currently active (the line `Automatic archival DISABLED` is displayed), start it by entering the following SVRMGR command:

```
connect internal
```

```
archive log start
```

Also check the parameters in the `init.ora` file that control the archiving process. Make sure the archiving process is started automatically during the next database startup. See [Archiving Parameters \[Page 34\]](#).

2. If the disk or archiving directory for the offline redo log files is full or is not available, choose one of the two following alternatives:

- Enter a new target directory for the online redo log archiving.

You will probably have to cancel the background process ARCH and restart it, specifying a new target directory. To do this, use the following SVRMGR commands:

```
archive log stop
```

```
archive log start '<new path>'
```

`<new path>` is the name of the “directory” that will be used for archiving. Please note the following special naming conventions for this “directory”: ORACLE interprets the last part of the specified path as a file prefix, it therefore must not physically exist in the directory. Find out about the default settings of the background process ARCH in the `init.ora` file (see [Archiving Parameters \[Page 34\]](#)). The database must remain active when you stop and restart the ARCH process.

Recovery: Errors During the Archiving of Online Redo Logs

Note that the path is automatically reset to the predefined value in profile init.ora the next time you start up the database.

Make sure that BRARCHIVE is able to archive the offline redo log files.

- Use the SAP utility BRARCHIVE to archive and delete the offline redo log files - this provides space in the archiving directory.

Partial Recovery

Partial Recovery

A partial recovery means that some data cannot be recovered after a database error. If one of the following problem constellations occurs, you will not be able to restore your database completely:

- All copies of the control file have been lost and you do not have any of the following files either:
 - Current backup copy of the control file
 - Log of the files in the database

In such cases, contact your SAP consultant or an ORACLE specialist. It may be possible to perform a complete recovery anyway. This depends on the exact situation, however, and cannot be explained in detail here.

- All members of the current online redo log group have been lost.
- One or more database files has been damaged and requires recovery, but one of the redo log files (offline redo log files or online redo log group) required for recovery is missing. You do not have a backup copy of the required redo log entries.

When you follow the SAP recommendations, this problem should not occur. The loss of all the members in an online redo log group is highly improbable, since the mirrored copies should be stored on different disks. The offline redo logs should also be regularly archived to tape using the BRARCHIVE utility.

- A tablespace has been lost and you have no backup copy of the tablespace. You should always have at least two generations of backups for every data file in the database, as well as the corresponding redo log files.

A partial recovery will cause data to be irretrievably lost, because the database can only be recovered in an older version. You can significantly reduce this risk by using a continual backup procedure for your database. We therefore ask you to adhere to the configuration and backup procedures recommended by SAP.



If you do not use the current control file for a recovery, but instead an older copy of the same, it is essential that you indicate this by using the supplement clause `using backup controlfile`.

The following section only describes the partial recovery after the loss of an offline redo log file.

For more information on partial recovery, please refer to the ORACLE documentation.

Recovery: Loss of an Offline Redo Log File

A media error has occurred in the data file area of the database, and one of the offline redo log files is no longer readable. For this reason, the recovery terminates with the last available redo log file in the sequence.

Proceed as follows during recovery:

1. If the database system is still active, use the following SVRMGR command to shut it down:

`shutdown abort`

ABORT is required in most cases, because the loss of individual data files means that changes in the SGA can no longer be copied.

2. Use the ALERT and trace files to analyze the error.
3. Restore all the available backups of all data files and the offline redo log files of all instances.

Use the SAP utility BRRESTORE (see [Command Options for BRRESTORE \[Page 413\]](#)).

4. After a partial recovery, the structure of the database may no longer be identical to that fixed in the current control file. Therefore, use a copy of the control file that reflects the structure of the database at the end of the recovery.
5. Mount the database. Enter the following SVRMGR commands:

```
connect internal
```

```
startup mount
```

6. If you could not load the backup files into their original directories or had to change file names, update the control file.

Use the following command to change the path names:

```
alter database rename file '<file name>' to '<file name>';
```

See [Updating the Control File \[Page 760\]](#).

7. You may be able to automate the recovery.

See [Automating the Recovery \[Page 759\]](#).

8. Use the following SVRMGR command to start the recovery:

```
recover database until cancel;
```

The use of the option `until cancel` means that the online redo logs will be reset when opened or not, depending on whether parameter `resetlogs` or `noresetlogs` is used.

If you do not use the current control file, enter the command

```
recover database until cancel using backup controlfile;
```

9. Depending on the recovery mode, the required offline redo log files are processed automatically (`autorecovery on`) until the file for the next log sequence number cannot be found, or (with `autorecovery off`) the recovery is stopped with `cancel`.
10. Once the message `recovery complete` or `recovery canceled` is displayed, you can use the following SVRMGR command to open the database again:

```
alter database open resetlogs;
```

RESETLOGS: The existing online redo log files are initialized, and the current log sequence number is reset to 1.

NORESETLOGS: The online redo log files are not initialized. This option should only be used when you did not use the option `using backup controlfile` and (unusually) all the online redo log files, including the current ones, were processed during the recovery.

Partial Recovery

The options `resetlogs` and `noresetlogs` are only possible after a `recover database until...` or after a recovery with the option using `backup controlfile`. See also [Actions after a Partial Recovery \[Page 757\]](#).

Point in Time Recovery

You can also select a point in time recovery, which you can carry out either manually or automatically. In contrast to the partial recovery with `until cancel`, this recovery is terminated at a specific time.

Use the following SVRMGR command:

```
recover database until time 'dd-mm-yyyy:hh:mm:ss';
```

Depending on the recovery mode (manual or automatic), the required redo log files are processed automatically, or you have to enter their paths and names. When the specified point in time is reached, the recovery is terminated. See also [Actions after a Partial Recovery \[Page 757\]](#).

Actions after a Partial Recovery

This section describes the measures that you should perform after a partial recovery with the case tapes below:

- **Case 1:** Restore of a complete offline backup and subsequently opening the database, without performing a complete recovery of the database
- **Case 2:** Restore of a complete online or offline backup and subsequent point in time recovery of the database (with ALTER DATABASE OPEN RESETLOGS)

Possible Problem Situations:

Situation A: The information about the last backups and volumes used in database tables SDBAH and SDBAD has been lost, because neither item is current in the database. As a result, during the next backup, BRBACKUP prompts you to mount volumes (based on the automatic volume management) that are logically free, but are physically locked.

Situation B: The current log sequence number was reset during a partial recovery.

- To a smaller value in case 1
- To the value 1 in case 2

BRARCHIVE does not find the newly written offline redo log files after the restore, because offline redo logs with these log sequence numbers have already been saved. The summary BRARCHIVE log arch<DBSID>.log still contains entries for successful archiving runs of these offline redo log files and, as a result, BRARCHIVE does not detect the new offline redo log files as files requiring archiving.

Solutions to the Problem Situations

Situation A: Using the detail BRBACKUP log, you can find out which volume was the last one used. Based on the information on the volume pool in initialization profile init<DBSID>.sap (parameter volume_backup), you can determine which volume should be used for the next backup. Explicitly name this volume when you start the next backup:

```
brbackup -v <volume name1>[,<volume name2>,...]
```

Situation B: Make sure the old offline redo log files in the archiving directory are renamed.

- **Case 1:** The current log sequence number can be seen in the detail BRBACKUP log of the backup you used to restore the data. Find the line `Current log sequence` (message BR116I). Then change the log sequence number in the last line of the summary BRARCHIVE log arch<DBSID>.log, which starts with #ARCHIVE, to the value: <(determined current log sequence number) -1>.

Example: entry before:

```
#ARCHIVE. 86 /oracle/C11/saparch/C11arch_86 1995-04-18 15.55.55
```

Current log sequence number determined: 30

Entry must be changed to:

```
#ARCHIVE. 29 /oracle/C11/saparch/C11arch_86 1995-04-18 15.55.55
```

Actions after a Partial Recovery

After the archiving, you should reset the changes in this line (note that new lines have been added).

- **Case 2:** Change the log sequence number in the last line of the summary BRARCHIVE log arch<DBSID>.log, which starts with #ARCHIVE, to zero (0).

Example: entry before:

```
#ARCHIVE. 86 /oracle/C11/saparch/C11arch_86 1995-04-18 15.55.55
```

Entry must be changed to:

```
#ARCHIVE. 0 /oracle/C11/saparch/C11arch_86 1995-04-18 15.55.55
```

After the archiving, you should reset the changes in this line (note that new lines have been added).

BRARCHIVE automatically recognizes the resetting of log sequence numbers if the database is opened when BRARCHIVE is started. In this case, the actions described in situation B are not necessary.



In general (as after all recovery operations), offline redo log files that were restored from tape to disk should be deleted from the disk after the recovery.

Automating the Recovery

You can control whether you want to perform a manual (`autorecovery off`) or an automatic recovery (`autorecovery on`). To do this, enter the appropriate SVRMGR command before you enter the Recover command.

- **set autorecovery on**

The required offline redo log files are automatically processed without requiring any user entry. The names and paths of the offline redo log files are derived from the `init.ora` parameters `log_archive_dest` and `log_archive_format`, which means that that required offline redo log files must first be imported under the appropriate names (see [Archiving Parameters \[Page 34\]](#)).

If the files cannot be imported under `log_archive_dest`, you can override the source specified in `archive_log_dest` by entering the command `set logsource = <log source>`. This means the files will now be expected in the directory identified under `logsource`.

- **set autorecovery off**

The processing of the individual redo log files must be initiated by the user (default value).

In the process, ORACLE automatically suggests a file derived from `log_archive_dest` (or `logsource`) and `log_archive_format`. Press RETURN to accept this value.

You can also explicitly enter the name of the redo log files.

Enter

- `cancel` to interrupt or cancel the recovery.
- `auto` to continue the recovery in automatic mode from this point.
- `from <log source>` to change the log source.

Updating the Control File

Updating the Control File

In the following cases, you will have to update the control file before you restore data:

- The saved data files are to be recovered on another hard disk, in a new directory or under new names.
- The status (ONLINE or OFFLINE) of one or several data files must be changed for the recovery to be continued.

The control file records the name, the path, or the group and the status of each data file in the database. You can update these specifications, which control the recovery process, with ORACLE commands.

If a disk error has occurred, for example, it may be necessary to restore the tablespaces in question on another disk. Before you restore the tablespaces, you have to update the path of the affected files in the control file.

Changing the Path Specifications

SAP recommends using the first of the methods listed below for renaming files.

- Mount the database. Enter the following SVRMGR commands:

```
connect internal
startup mount
```

To update the path of data files in the control file, use the SVRMGR command

```
alter database rename file '<file name>' to '<file name>';
```

The target file must exist; the name of the source file must agree with that in the control file.

You can also specify a list of file names, to rename all the files at once. Please note, however, that any problems that may occur during renaming will be easier to diagnose if you rename the files individually.

- You can also change the paths of the data files of a tablespace when the database is running. Set the corresponding tablespace to OFFLINE before renaming. Use the following SVRMGR commands:

```
alter tablespace <tablespace name> offline;
alter tablespace <tablespace name> rename datafile
'<file name1>' [, '<file name2>',...] to '<file name1>' [, '<file
name2>',...];
```

You have to enter this command separately for each tablespace in which you have to change the file information.

The target file must exist; the name of the respective source files must agree with those in the control file.

Setting Files to ONLINE

If the data files of a tablespace are OFFLINE, when the database crashes or was stopped with `shutdown abort` and a recovery is necessary, you may have to reset the files that belong to this tablespace to ONLINE again.

To do this, use the following SVRMGR commands:

```
connect internal
startup mount
alter database datafile '<complete file name>' online;
```

Disaster Recovery

Disaster Recovery

Prerequisites

If you lose your entire system (possibly including hardware), and have not made any special security precautions for such an occurrence, for example an ORACLE standby configuration, then you have to recover the system as much as possible, step by step. The following describes how to install the system again, and how to keep data loss at a minimum by using the SAP program BRRESTORE.

Note:

`$ORACLE_HOME` and `$SAPDATA_HOME` stand for the environment variables set under UNIX, `%ORACLE_HOME%` and `%SAPDATA_HOME%` are the corresponding variables under Windows NT. Under UNIX `svrmgrl` stands for the ORACLE Program Server Manager, under Windows NT the name is `SVRMGR30`.

Under UNIX the profile files are in the directory `$ORACLE_HOME/dbs`, under Windows NT they are in the directory `%ORACLE_HOME%\database`.

Procedure

1. Recovering the hardware environment

Install the same operating system that you had before the loss.
Create the distribution of the hard disks according to the original size and structure. Take this into account when purchasing the replacement disks.
Format the disks as described in the SAP Installation Guide.

2. Installation of the R/3 System

The installation of the SAP software includes the software installation of the database. When you do this, you must not create the initial SAP database.

3. Modifying the installation

When you do this, take special consideration of the mounted file systems at the time of the loss. If necessary, create new SAPDATA directories (mount points). These usually identify a hard disk or a logical volume.

Place the **BRBACKUP backup tape** of the last successful and complete backup in the tape unit.

4. Adapting the `init<SID>.sap` file

At least one valid `init<SID>.sap` file must exist before you can start BRRESTORE.

Adapt this file, for example with respect to the devices and drivers used. (See: Configuration of the BRBACKUP profile file).

In addition to this, note the following points:

- Check all parameters specific to the operating system and adapt them.

- To retrieve the latest backup of the BRBACKUP profile file `$ORACLE_HOME/dbs/init<SID>.sap`, enter the following commands (after adapting the parameters specific to the operating system):

```
BS> cd $ORACLE_HOME/dbs
BS> brrestore -n init_sap
```

Alternatively:

- The file `init<SID>.sap` can also be read as a third file from a BRBACKUP tape, using an operating system command (`cpio`).

5. Restoring certain files

Restore the files of the last BRBACKUP database backup, listed below.

- ORACLE profile file: `$ORACLE_HOME/dbs/init<SID>.ora`
SAPDBA profile file: `$ORACLE_HOME/dbs/init<SID>.dba`

Method:

```
BS> cd $ORACLE_HOME/dbs
BS> brrestore -n init_ora
```

- SAPDBA central log: `$SAPDATA_HOME/sapreorg/reorg<SID>.log`
SAPDBA structure log: `$SAPDATA_HOME/sapreorg/struct<SID>.log`

Method:

```
BS> cd $SAPDATA_HOME/sapreorg
BS> brrestore -n reorg_log
```

- BRBACKUP summary log: `$SAPDATA_HOME/sapbackup/back<SID>.log`

Method:

```
BS> cd $SAPDATA_HOME/sapbackup
BS> brrestore -n sum_log
```



The last entry in the summary log matches the last backup. This is the relevant entry for the detail log (see next step). The last entry is normally incomplete, for example: return code not current. However this does not have any consequences for any of the subsequent operations.

- BRARCHIVE detail log `$SAPDATA_HOME/sapbackup/<coded timestamp>.<fid>`

Method:

```
BS> brrestore -n det_log
```

Take the `<coded_timestamp>` and the `<fid>` from the name of this restored file in the restore directory. This is required for the following operations.

Remove the BRBACKUP tape from the tape unit and replace it with the **BRARCHIVE tape** of the last backup of the offline redo log files.

Restore the following file from the BRARCHIVE tape:

- BRARCHIVE summary log: `$SAPDATA_HOME/saparch/arch<SID>.log`

Method:

```
BS> cd $SAPDATA_HOME/saparch
BS> brrestore -n sum_log
```

Disaster Recovery

Remove the **BRARCHIVE** tape from the tape unit and replace it with the **BRBACKUP** tape.

Using an external backup program

If you use the **BACKINT** interface the procedure differs in some respects:

The profiles `init<SID>.utl` (optional, depending on backup program) and `init<SID>.sap` must exist before you use **BACKINT**. In the profile `init<SID>.sap` you only have to define the name of the profile of the external backup program in the parameter `util_par_file`. You may have to reconstruct this first. You can do this either by editing, or by using the **BACKINT** command line interface.

Then restore the following files individually:

- **BRBACKUP** current summary log: `$SAPDATA_HOME/sapbackup/back<SID>.log`

Method:

```
BS> cd $SAPDATA_HOME/sapbackup
BS> brrestore -d util_file -b2 #NULL -m
$SAPDATA_HOME/sapbackup/back<SID>.log
```

The last entry in the summary log matches the last backup. This is the relevant entry for the detail log (see next step). The last entry is normally incomplete, for example: return code not current. However this does not have any consequences for any of the subsequent operations. Take the `<coded_timestamp>` and the `<fid>` from this entry. This is required for the following operations.

- **BRBACKUP** current detail log:
`$SAPDATA_HOME/sapbackup/<coded_timestamp>.<fid>`

Method.

```
BS> brrestore -d util_file -b2 #NULL -m
$SAPDATA_HOME/sapbackup/<coded_timestamp>.<fid>
```

Take the ID of the external backup tool (`<external backup tool ID>`) from this **BRBACKUP** detail log.

To do this search for the character string `#INLOG` in the detail log.

You can find the 'backup tool ID' in the line beginning with '#SAVED', for example:

```
#INLOG.../oracle/C11/dbs/initC11.sap
#SAVED...<external backup tool ID>
```

- All other files that carry the ID of the external tool determined in the last step.

Method:

```
BS> brrestore -d util_file -b2 <external backup tool ID> -m
<pathname>
```

```
$ORACLE_HOME/dbs/init<SID>.ora
$ORACLE_HOME/dbs/init<SID>.sap
$ORACLE_HOME/dbs/init<SID>.dba
$ORACLE_HOME/dbs/init<SID>.utl
$SAPDATA_HOME/saparch/arch<SID>.log
$SAPDATA_HOME/saparch/<coded timestamp of BRARCHIVE>.<fid>
$SAPDATA_HOME/sapreorg/struct<SID>.log
$SAPDATA_HOME/sapreorg/reorg<SID>.log
```



The procedure described above is relatively complex. We recommend that you use the restore function of the external tool as an alternative.

6.) Resetting the database to the last online or offline backup.

Note: An online backup must always be recovered to a point in time before the backup. If you require a more current status than that of the imported offline backup you can also make a recovery.

Restore the backup completely:

```
BS> brrestore -b <coded_timestamp>.<fid> -m full
```

The <coded_timestamp> and <fid> were already determined from the summary BRBACKUP log.



If you have to make another recovery you must **not open** the database at this point.

7.) For a **recovery** you need to perform the following steps:

Remove the BRBACKUP tape from the tape unit and replace it with the **BRARCHIVE tape**.

a) Retrieval of the required offline redo log files

```
BS> brrestore -a <log_seg_No_A>-<log_seg_No_B>
```

<Log_Seg_No_A> stands as 'Current log sequence' in the header of the detail BRBACKUP log.

<Log_Seg_No_B> stands as 'target log sequence no.' and matches the last backed up offline redo log file (last entry in the summary BRARCHIVE log arch<SID>.log).

b) Database recovery with the ORACLE Server Manager

```
svrmgrl> set autorecovery off
svrmgrl> recover database using backup controlfile until cancel
svrmgrl> alter database open resetlogs
```

See also: [Standby Database Scenario: \[Page 185\]](#)

Oracle Databases on Raw Devices

Oracle Databases on Raw Devices

Application programs such as database management systems frequently avoid the file management of the UNIX system and the data is written directly to a partition. In this case, the partition is used as a raw device. No file system exists in such a partition. Using this type of storage might bring certain advantages regarding the speed when accessing the data, but requires its own file management.

An Oracle database can be operated from raw devices. The SAP utilities for backing up the database support this configuration.

Advantages

Working with raw devices brings considerable performance gains for the following reasons:

- Since the usual buffer cache for a file system is not needed, less main memory is required.
- You do not have to administer a file system.
- Since no management information has to be stored on the disk, less disk space is required.
- Since the accesses do not have to be synchronized and the management information does not have to be recorded, the load on the CPU is reduced.

If you operate the Oracle database system on raw devices, you can use the parallel server option.

[Oracle Databases and Parallel Server Option \(OPS\) \[Page 769\]](#)

Disadvantages

Working with raw devices makes administration more difficult for the following reasons:

- There is no description of the files residing on the raw devices in the system.
- The configuration of the storage space is inflexible because only one database file is permitted for each raw device (and therefore for each partition). The sizes of the individual partitions must be adjusted to the sizes of the database files. This makes later relocation of the database files to other partitions more difficult.
- It is possible to save raw devices with the `dd` command, but its complicated syntax must be accepted.



SAPDBA can make only restricted checks in raw devices, since it can check only those devices that belong to the database user `ora<sapsid>`. Raw devices are recognized only if they are allocated to this database SAPSID (SAPDBA does not check whether there are also objects from other applications in these raw devices).

SAPDBA Support for Raw Device Configuration

The following information is important for you if you intend to use SAPDBA for ORACLE database administration on raw devices:

- The data for a tablespace can be stored either in raw devices or in files. Be sure to observe the SAP naming convention for tablespaces on raw devices:

`<ORACLE_HOME>/sapraw/<TSP>_<Number> → <Raw-Device-Dir>/<Device>`

`<Number>`: Sequence number assigned to the raw devices (or files) belonging to the tablespace

`<Raw-Device-Dir>`: Specifies the central directory containing the devices. This directory may also be specified in the [rawdev_dir \[Page 283\]](#) parameter.

`<Device>`: `<DBSID><TSP>_<Number>`

See also [Environment Variables \[Page 45\]](#).

Each tablespace “file” (held on a raw device) visible to ORACLE is a symbolic link to a raw device. SAPDBA checks that the naming convention has been observed.



Tablespace PSAPDOCUD

`/oracle/C11/sapraw/docud_1 → /dev/rdsk/C11docud_1`

You should also compare this with the [Naming Convention \[Page 691\]](#) for tablespaces in the file system.

- The database link structure is one of things recorded in the structure log `struct<DBSID>.log`.

If the database check is done using SAPDBA, SAPDBA uses this structure log to check whether the link structure is still complete and immediately repairs it if not. If a tablespace is extended, a new tablespace is created or a tablespace is reorganized with data files then SAPDBA updates the structure log. See [Structure Log \[Page 299\]](#).

- Each raw device can contain only one tablespace “file” (requirement of ORACLE). SAPDBA can set the size of this partition. Therefore, in the case of tablespace extension (adding a “file” from a raw device), for example, the size of the raw device is determined automatically and is used for the size of the file. See [Extending a Tablespace \[Page 686\]](#).
- It may be a good idea to specify the [rawdev_dir \[Page 283\]](#) and [blkdev_dir \[Page 265\]](#) parameters in the `init<DBSID>.dba` profile.

Raw Devices and BRBACKUP/ BRRESTORE

Raw Devices and BRBACKUP/ BRRESTORE

The BRARCHIVE utility remains unchanged in a raw device configuration because offline redo log files must always reside in a file system.

There are some changes in the BRBACKUP and BRRESTORE programs in a raw device configuration, but these changes have no effect on the function scope of the programs. The known functionality of BRBACKUP and BRRESTORE for backing up and restoring file systems remains unchanged.

Use the `dd` command to access raw devices.

- With `dd`, you can write directly from/to the raw device to/from tape (backup/restore to/from tape)
- With `dd`, you can copy directly from/to the raw device to/from disk with file system (backup/restore to/from disk)
- In case of a backup with software compression, output of the command `dd` is sent directly to the `compress`. In case of a restore with decompression, output of `uncompress` is sent to `dd`.
- In case of a backup on a remote computer, output of `dd` is sent for example directly to `remsh` (`/rsh` etc.). In case of a restore from a remote computer, it is sent from `remsh` (`/rsh` etc.) to the `dd` command.
- There are new `init<DBSID>.sap` parameters for defining the options for the `dd` command: see [dd_flags \[Page 328\]](#) and [dd_in_flags \[Page 329\]](#).



Since the `dd` command **does not support a dd continuation tape** (in this case an I/O error is reported), a method which is similar to the `cpio` continuation method is not supported. This means that each individual database file residing on the raw devices must completely fit onto one tape. This restriction does **not** refer to the BRBACKUP continuation tape management, which means that the BRBACKUP utility can request continuation tapes if they should be necessary for backup of the next database files.

ORACLE Databases and Parallel Server Option

If an ORACLE database is operated from raw devices, it is possible to use the Parallel Server option of the ORACLE-RDBMS. The SAP utilities for backing up the database support this configuration. In the following sections, the special features of the SAP utilities for backing up when using the Parallel Server option are explained.

[ORACLE Databases and Parallel Server Option: Overview \[Ext.\]](#)

[Parallel Server Configuration for Using SAPDBA \[Ext.\]](#)

[Parallel Server Configuration when Using BRBACKUP, BRARCHIVE and BRRESTORE \[Page 770\]](#)

OPS with BRBACKUP, BRARCHIVE and BRRESTORE

OPS with BRBACKUP, BRARCHIVE and BRRESTORE

BRBACKUP and BRRESTORE are normally started in the DDB instance only, as well as BRARCHIVE in the case of the central scenario (see [Central Scenario \[Page 778\]](#)). The following requirements must be fulfilled:

- Read the general remarks in [Oracle Databases and Parallel Server Option \(OPS\) \[Page 769\]](#).
- The necessary executables (br*) must exist on the DDB instance host in directory `/usr/sap/<SAPSID>/SYS/exe/run`.
- The init.ora profiles for the individual instances (`init<DBSID>.ora` for all instances) must exist on the DDB instance host in directory `<ORACLE_HOME>/dbs`.

Make sure that the profiles of all instances are actually stored in the directory of the DDM instance. All profile files that may have been included using the parameter `IFILE` must also exist there. BRBACKUP/BRARCHIVE save all profiles as a cpio archive file.
- The backup devices (tape units or hard disks) are normally connected locally to the DDB instance host.
 - If you have read the remarks in [Oracle Databases and Parallel Server Option \(OPS\) \[Page 769\]](#), user `system` now has the authorization to start and stop the database instances remotely. You can therefore call BRBACKUP and BRARCHIVE in the usual manner, for example `brbackup/brarchive -u system/<password>...`

BRBACKUP, BRARCHIVE and BRRESTORE can manage up to 255 parallel instances.

Information in the Profile `init<DBSID>.sap`

- `dd_flags`, `dd_in_flags`

A requirement for using the Parallel Server option is that the database is installed on raw devices. The backup then runs via the `dd` command. Therefore make sure that the parameters [dd_flags \[Page 328\]](#) and [dd_in_flags \[Page 329\]](#), which specify the required `dd` command, are set. See [Supporting the Raw Device Configuration with BRBACKUP and BRRESTORE \[Page 768\]](#).

- `parallel_instances`

This parameter defines the instances running in parallel to the DDB instance. See [parallel_instances \[Page 336\]](#).



The local (DDB) instance must not appear in the list of instance descriptions.

- `archive_copy_dir`

A BRARCHIVE archivation on the hard disk is required when archiving according to the mixed scenario (see [Mixed Scenario \[Page 780\]](#)). See [archive_copy_dir \[Page 311\]](#).

Special Features of BRBACKUP

Special Features of BRBACKUP

The parameter [parallel_instances \[Page 336\]](#) must be set in the `init<DBSID>.sap` profile used by BRBACKUP in order that BRBACKUP be able to access all the ORACLE instances.

The backup of a database with the Parallel Server configuration using the SAP utility BRBACKUP runs as follows:

1. All ORACLE instances which are stopped are restarted in parallel by BRBACKUP with the appropriate SQLDBA commands.
2. BRBACKUP logs on to the local (DDB) instance and carries out the common actions in the database (determination of the database structure, the volumes used...).
3. If an offline backup is to be carried out, all instances are stopped. In all other cases, they continue running during the backup.
4. The required backup is started.
5. If an offline backup was carried out, all database instances are restarted in parallel at the end of the backup.
6. BRBACKUP updates the tables SDBAH and SDBAD.
7. The status of all instances before the start of the backup with BRBACKUP is recovered. That is, the instances which were stopped before the backup are now stopped again.

During the entire backup, BRCONNECT monitors **all** database instances and checks them constantly to see that the required status of the instances stays unchanged. If the status of an instance changes during the backup, BRCONNECT interrupts the entire backup.

In the BRBACKUP log, both before and after the actual backup, a log is made of the parameters relevant to the online redo log files that are being archived for **all** database instances in question (output of the ARCHIVE LOG LIST command).

If you run an offline backup of the entire database, a file of **all** online redo log groups for the ORACLE instances is saved in each case.

The functionality of BRBACKUP is available (in particular, the support of software and hardware compression) with one exception concerning the `dd` command: when using the `dd` command, there is no continuation mechanism. See [Supporting the Raw Device Configuration with BRBACKUP and BRRESTORE \[Page 768\]](#).

Special Features of BRARCHIVE

There are three ways to archive the offline redo log files using BRARCHIVE in the case of a Parallel Server configuration of the database:

1. [Distributed Scenario \[Page 775\]](#)
2. [Central Scenario \[Page 778\]](#)
3. [Mixed Scenario \[Page 780\]](#)



Normally the R/3 System is installed so that the mixed scenario is used for archiving for a Parallel Server configuration. SAP recommends this scenario for archiving the offline redo log files in production systems.

The distributed and especially the central scenario should only be used with some restrictions and then only for test systems.



Offline redo log files of different instances can be stored in a local archiving directory of an instance. This can occur if the online redo log files of instance X were not saved in the archiving directory for a long time (either the instance is closed or it is open but not active. Therefore a redo log file change is not forced and thus ORACLE archivation is not activated). One of the “active” instances notices this status and ensures that the current online redo log file of instance X is saved in its own archiving directory.

Parameter [parallel_instances \[Page 336\]](#) must therefore be set in the `init<DBSID>.sap` profiles used by BRARCHIVE in all three scenarios, so that BRARCHIVE can find and archive the offline redo log files of the other ORACLE instances.

Special Features of BRRESTORE

Special Features of BRRESTORE

BRRESTORE uses the `dd` command to restore files that were saved on a volume (tape, hard disk,...) with this command. You should therefore make sure that the `init<DBSID>.sap` parameter [`dd_in_flags` \[Page 329\]](#) is set accordingly. See [Supporting the Raw Device Configuration with BRBACKUP and BRRESTORE \[Page 768\]](#).

BRRESTORE for Restoring Database Files

There are no special features of BRRESTORE concerning the functionality of the utility programs.

BRRESTORE for Restoring the Archived Offline Redo Log Files

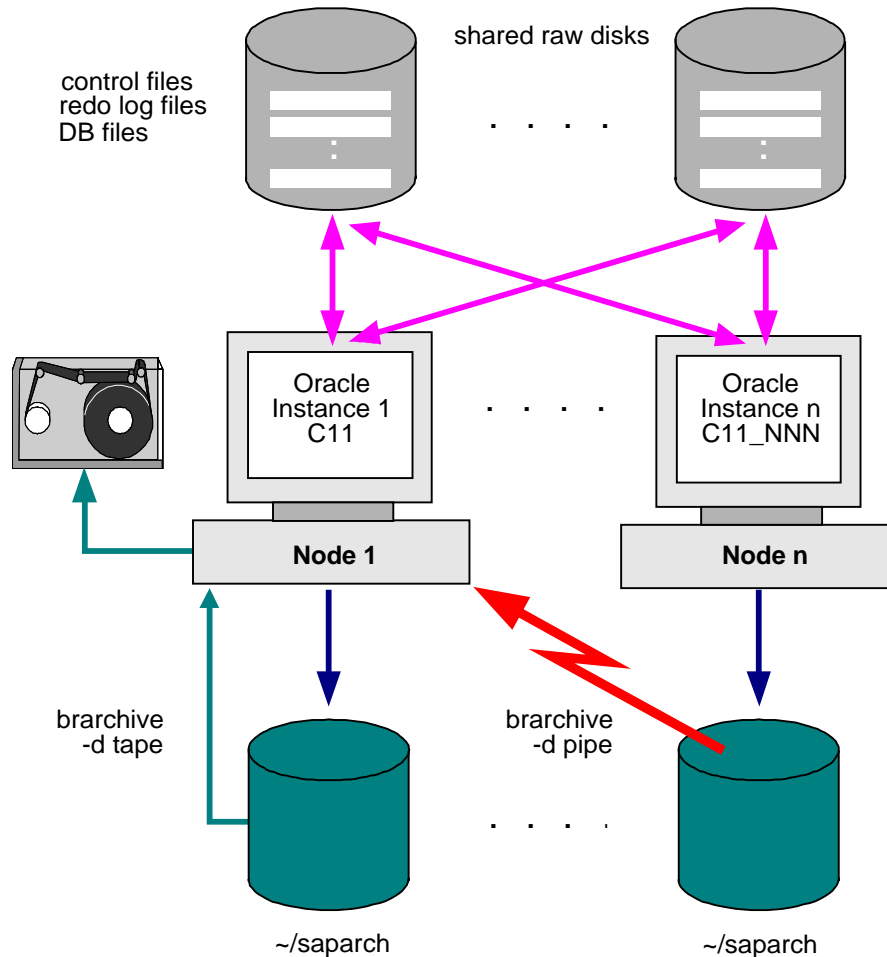
- **Distributed Scenario:** BRRESTORE must be started on every ORACLE instance in order to restore the corresponding archived redo log files into the local archiving directory. If required, the restored redo log files can be copied onto the DDB instance using operating system resources or be made available via several NFS-mounts. (This is necessary, for example, if a SQLDBA recovery session is to be started, which is normally done on the DDB instance). See also “*Further Remarks*” in [Distributed Scenario \[Page 775\]](#).
- **Central and Mixed Scenario:** The required archived redo log files of all ORACLE instances can be restored with one BRRESTORE call. To do this, the syntax of BRRESTORE was extended with the options `-a` | `-a1` | `-a2` to specify the instance ID `<DBSID>`. See [-a|-archive|a1|archive1 \[Page 415\]](#) or [-a2|-archive2 \[Page 416\]](#). See also “*Further Remarks*” in [Central Scenario \[Page 778\]](#) and [Mixed Scenario \[Page 780\]](#).



```
brrestore -a 110-120,C11_002,70-80,C11_003,85-95
```

The archived offline redo log files with log sequence numbers 110-120 of the local ORACLE instance, those with log sequence numbers 70-80 of ORACLE instance C11_002 and those with log sequence numbers 85-95 of instance C11_003 are restored.

Distributed Scenario



Each ORACLE instance has its own local archive directory. This directory should not reside on a mounted disk.

BRARCHIVE is started locally on the hosts of all ORACLE instances.

In this case, BRARCHIVE works similarly to a configuration without the Parallel Server option. In each case, BRARCHIVE saves the offline redo log files of the local instance. BRARCHIVE can determine these by means of the thread number (`init<DBSID>.ora` parameter `thread`), which is allocated to every local instance.

If offline redo log files of other instances occur in the local archiving directory, BRARCHIVE also archives these.

Every ORACLE instance has its own volume management controlled by BRARCHIVE. The offline redo log files which were found in the local archiving directory of the corresponding instance are saved on every tape (in exceptions these can be offline redo log files of different

Distributed Scenario

instances). The assignment of the offline redo log files to the individual instances can be seen in the file names. The name conventions are defined by the `init<DBSID>.ora` parameters `log_archive_dest` and `log_archive_format`.



```
log_archive_dest = /oracle/C11/saparch/C11arch
```

```
log_archive_format = %t_%s.dbf
```

%t - Thread number (for identifying the instance)

%s - Log sequence number



The ORACLE placeholder @ (for ORACLE_SID) may not appear in the name of an offline redo log file; this means that it may not appear in either `log_archive_dest` nor in `log_archive_format`.

The offline redo log files can be archived either to local tape units (connected to the host of the instance, `backup_dev_type = tape`) or to tape units connected to the host of the DDB (`backup_dev_type = pipe`).

Advantages

- The use of BRARCHIVE in this case is similar to its use during a non-OPS configuration.

Disadvantages

- Free disk space must be made available on all instance hosts, so that a local ORACLE backup can be made of the offline redo log files in the archiving directory.
- BRARCHIVE must be started once per ORACLE instance.
- For every instance, a separate BRARCHIVE volume pool must be managed.
- SAP does not recommend this scenario. See [Special Features of BRARCHIVE \[Page 773\]](#).

The use of BRARCHIVE in the distributed scenario is useful if a large number of offline redo log files are created on every ORACLE instance, which would nearly fill a volume (tape).

Further Remarks

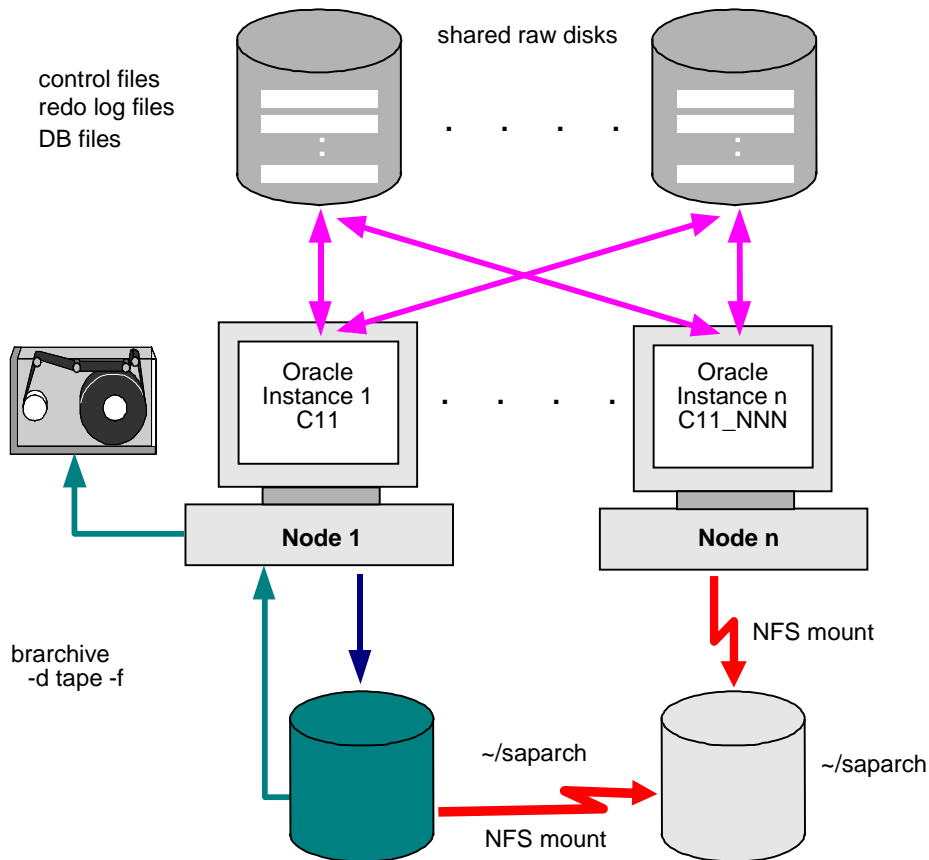
The offline redo log files are required if the database has to be recovered. Recovery measures should only be performed by an experienced administrator, who is familiar with the OPS-specific behavior of the database system in this case.

Note the following if the offline redo log files are to be restored:

The required offline redo log files must be restored separately for each instance (using SAPDBA or a BRRESTORE call). The target directory for the offline redo log files of all instances is the archiving directory of the instance from which the database system recovery is to be started. Then start the recovery (information about this can be found in the ORACLE documentation).

Central Scenario

Central Scenario



There is a global archiving directory which normally is created on the host of the DDB instance. This directory can be accessed from the other hosts with NFS-Mount. All the archiving processes save the online redo log files in this global working directory in this manner (if necessary using the network).



Definition in `init<DBSID>.ora`:

```
log_archive_dest = /oracle/C11/saparch/C11arch
```

```
log_archive_format = %t_%s.dbf
```

%t - Thread number (for identifying the instance)

%s - Log sequence number



The ORACLE placeholder @ (for ORACLE_SID) may not appear in the name of an offline redo log file; this means that it may not appear in either `log_archive_dest` nor in `log_archive_format`.

BRARCHIVE is only started on the host of an ORACLE instance (generally the DDB instance since normally the archiving directory is available locally there and thus archiving is not necessary with the network) and archives the complete offline redo log files of all the ORACLE instances.

BRARCHIVE creates a separate summary log, `arch<DBSID>.log`, for every ORACLE instance. It is stored in the global archiving directory.

Advantages

- There is only one volume pool managed by BRARCHIVE.
- The entire BRARCHIVE functionality is available, even continuous archiving with the option `-f`.
- Only the disk space in the global archiving directory must be provided.

Disadvantages

- If a network error occurs, it is possible that the online redo log files cannot be saved in the archiving directory by the ORACLE archiving process. This can cause an “archive stuck” and thus result in failure of a database instance.
- If a network error occurs, access to the NFS-mounted disks might be disturbed. This can cause corrupt offline redo log files to be created by the ORACLE archiving processes.
- SAP does not recommend this scenario. See [Special Features of BRARCHIVE \[Page 773\]](#).

Due to possible network problems, use of the central scenario is only acceptable for test systems with a very stable and reliable network.

Further Remarks

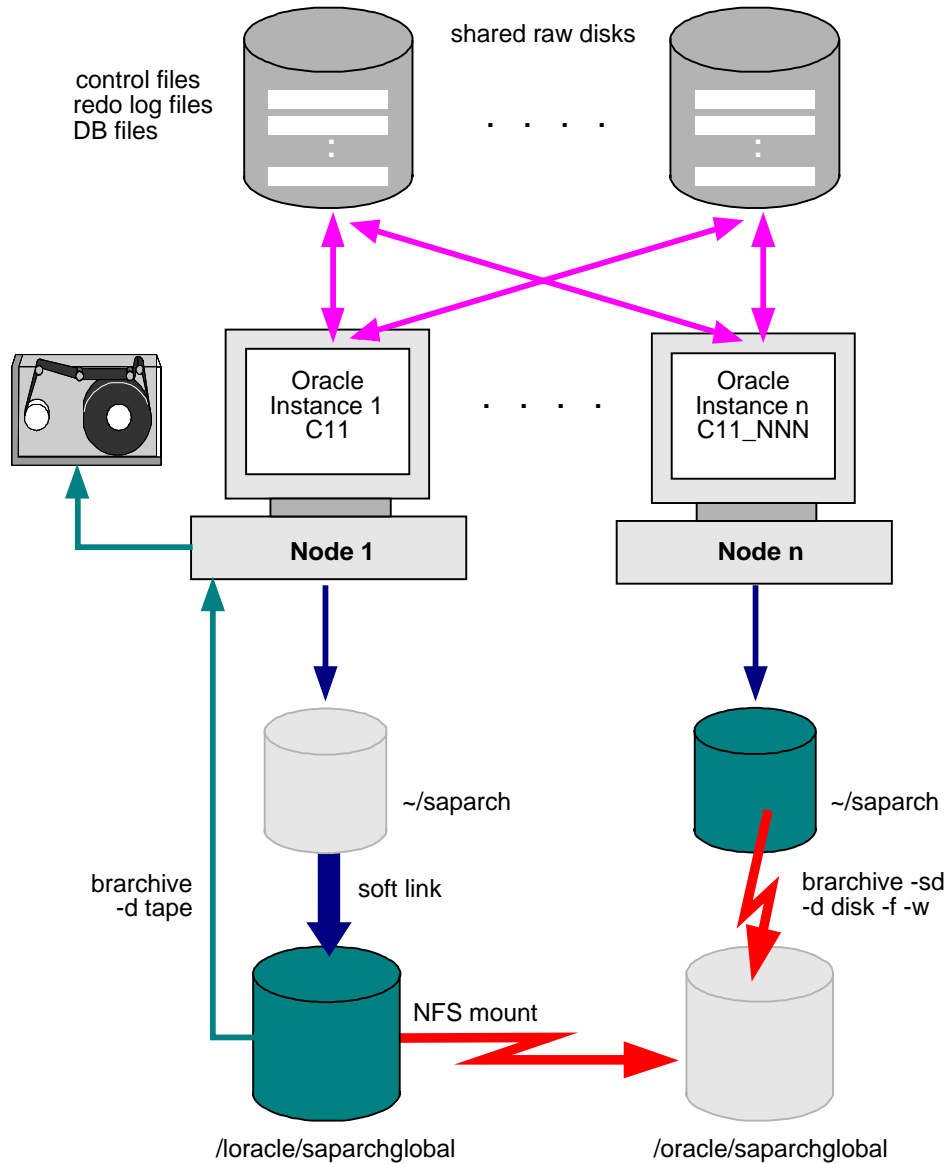
The offline redo log files are required if the database has to be recovered. Recovery measures should only be performed by a very experienced administrator, who is familiar with the OPS-specific behavior of the database system in this case.

Note the following if the offline redo log files have to be restored:

The required offline redo log files of all the instances can be restored in the global archiving directory (with SAPDBA or a BRRESTORE call). You can now start the recovery on the instance with the global archiving directory (generally the DDB instance, see the above graphic). Information about this can be found in the ORACLE documentation.

Mixed Scenario

Mixed Scenario



BRARCHIVE is started on all ORACLE instance hosts. However, its activities differ depending on whether or not it is the DDB instance host.

- On non-DDB instance hosts, there are relatively small archive directories, which can temporarily contain the respective offline redo log files.
- On the DDB instance host, there is, locally, a relatively large (central) archiving directory that can contain the offline redo log files of all ORACLE instances.

Every non-DDB instance can access the central archiving directory via NFS-mount.



Name of the R/3 System: C11

Large local archiving directory on the DDB instance: `/loracle/saparchglobal`

Directory `/oracle/C11/saparch` refers to this large archiving directory using a link.

Small, local archiving directories on the non-DDB instances:
`/oracle/C11/saparch`

The large archiving directory is added to every non-DDB instance using an NFS-mount under: `/oracle/saparchglobal`

Definitions in `init<DBSID>.sap`

Parameter `archive_copy_dir` on all ORACLE instances
`archive_copy_dir = /oracle/saparchglobal`

Definitions in `init<DBSID>.ora`:

`log_archive_dest = /oracle/C11/saparch/C11arch`

`log_archive_format = %t_%s.dbf`

`%t` - Thread number (for identifying the instance)

`%s` - Log sequence number



The ORACLE placeholder `@` (for `ORACLE_SID`) may not appear in the name of an offline redo log file; that is, it may not appear in either `log_archive_dest` nor in `log_archive_format`.

Archiving of the offline redo log files of the non-DDB instances executes as follows:

Archiving on the Disk

On every non-DDB instance, `BRARCHIVE` is started regularly or with the `-f` option. `BRARCHIVE` copies the offline redo log files for each instance from the local archiving directory into the NFS mounted central archiving directory (`BRARCHIVE` call to archive the offline redo log files on disk, option `-d disk`). To check the success of the operation, you should use the `-w` option to verify the backup (see above graphic).



This `BRARCHIVE` run should be terminated with `brarchive -f stop` and then restarted every day (for performance reasons).

Archiving on Tape

The `BRARCHIVE` running on the DDB instance host archives the offline redo log files found in the central archiving directory (`BRARCHIVE` call to archive the offline redo log files on tape, option `d tape`). The offline redo log files of all ORACLE instances are continuously archived to the volumes (tapes) via this backup device connected to the DDB host.

Mixed Scenario

Advantages

- Local backups of the offline redo log files can be stored in relatively small archiving directories on the hosts of all non-DDB instances.
- There is only one volume pool managed by BRARCHIVE.
- Normally SAP installs this scenario for OPS databases.

Disadvantages

- On all non-DDB instances, BRARCHIVE must either be started regularly or started with the option `-f` so that the relatively small archiving directories do not overflow.

SAP normally installs this scenario. It should generally be used in production systems, especially if offline redo log files occur relatively frequently on all instances (and you are therefore working with option `-f` to avoid an "Archive Stuck").

Further Remarks

The offline redo log files are required if the database has to be recovered. Recovery measures should only be performed by an experienced administrator, who is familiar with the OPS-specific behavior of the database system in this case.

Note the following if the offline redo log files have to be restored:

The required offline redo log files of all the instances which were archived on tape can be restored in the central working directory (generally residing on the DDB instance) (with SAPDBA or a BRRESTORE call). It is essential that you now check whether there are offline redo log files in the archiving directories of the other instances which were not yet copied to the directory of the central instance by BRARCHIVE (this means that archiving on disk did not yet execute). If necessary, BRARCHIVE has to be started on the individual instances, in order to copy missing offline redo log files into the central archiving directory. You can then start the recovery on the central instance (information about this can be found in the ORACLE documentation).